



(12) 发明专利申请

(10) 申请公布号 CN 115048372 A

(43) 申请公布日 2022. 09. 13

(21) 申请号 202210378909.5

(22) 申请日 2022.04.12

(71) 申请人 北京贝壳时代网络科技有限公司
地址 101520 北京市密云区兴盛南路8号院
2号楼106室-802(商务区集中办公区)

(72) 发明人 张志强 吕冬冬

(74) 专利代理机构 北京润平知识产权代理有限公司 11283
专利代理师 肖冰滨

(51) Int. Cl.

G06F 16/22 (2019.01)

G06F 16/2457 (2019.01)

G06F 9/54 (2006.01)

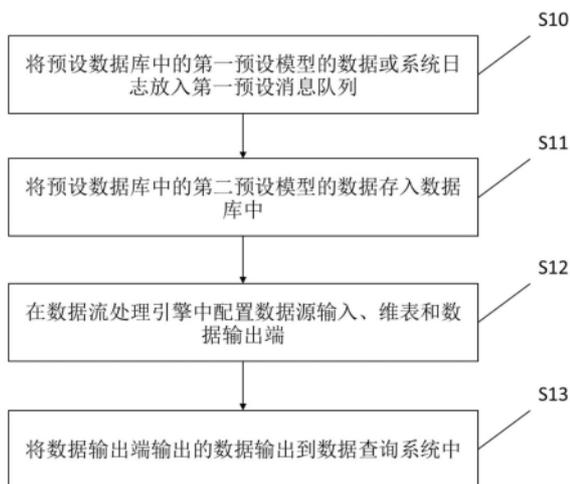
权利要求书2页 说明书8页 附图2页

(54) 发明名称

多流数据的关联方法和关联装置

(57) 摘要

本发明实施例提供一种多流数据的关联方法和关联装置,属于实时数据指标领域。该关联方法包括:将预设数据库中的第一预设模型的数据或系统日志放入第一预设消息队列,以得到主流数据;将所述预设数据库中的第二预设模型的数据存入数据库中,以得到实时维表数据;在数据流处理引擎中配置数据源输入、维表和数据输出端,其中,所述数据源输入对应所述主流数据,所述维表对应所述实时维表数据;以及将所述数据输出端输出的数据输出到数据查询系统中,以供数据查询。藉此,实现了提供实时查询的服务。



1. 一种多流数据的关联方法,其特征在于,该关联方法包括:

将预设数据库中的第一预设模型的数据或系统日志放入第一预设消息队列,以得到主流数据;

将所述预设数据库中的第二预设模型的数据存入数据库中,以得到实时维表数据;

在数据流处理引擎中配置数据源输入、维表和数据输出端,其中,所述数据源输入对应所述主流数据,所述维表对应所述实时维表数据;以及

将所述数据输出端输出的数据输出到数据查询系统中,以供数据查询。

2. 根据权利要求1所述的关联方法,其特征在于,所述在数据流处理引擎中配置数据源输入、维表和数据输出端包括:

将所述主流数据与所述实时维表数据进行关联。

3. 根据权利要求2所述的关联方法,其特征在于,响应于将所述主流数据和所述实时维表数据进行关联时,所述主流数据与所述实时维表数据不能成功关联,该关联方法还包括:

控制所述主流数据与所述实时维表数据进行延迟重试关联。

4. 根据权利要求3所述的关联方法,其特征在于,控制所述主流数据与所述实时维表数据进行延迟重试关联后,响应于所述主流数据与所述实时维表数据仍不能成功关联,该关联方法还包括:

将所述主流数据输出到第二预设消息队列中,以使得所述主流数据能够重新执行在数据流处理引擎中配置数据源输入、维表和数据输出端的操作。

5. 根据权利要求1所述的关联方法,其特征在于,该关联方法还包括:

用离线数据覆盖对应的在线数据,以修复所述在线数据的误差,其中,所述在线数据为所述数据输出端输出的数据,所述离线数据为基于所述第一预设模型的数据或系统日志以及所述第二预设模型的数据进行离线计算得到的。

6. 根据权利要求5所述的方法,其特征在于,所述用离线数据覆盖对应的在线数据为采用离线T-N数据覆盖所述在线数据中与所述离线T-N数据具有相同时间范围的所述在线数据,其中,所述离线T-N数据指的是采用截止到当前时刻T的前N天获取到的所述离线数据。

7. 根据权利要求1-6中任一项所述的关联方法,其特征在于,该关联方法还包括:

对比离线数据与对应的在线数据是否相同,其中,所述在线数据为所述数据输出端输出的数据,所述离线数据为基于所述第一预设模型的数据或系统日志以及所述第二预设模型的数据进行离线计算得到的;以及

发送对比结果。

8. 一种多流数据的关联装置,其特征在于,该关联装置包括:

主流数据获取模块,用于将预设数据库中的第一预设模型的数据或系统日志放入第一预设消息队列,以得到主流数据;

实时维表数据获取模块,用于将所述预设数据库中的第二预设模型的数据存入数据库中,以得到实时维表数据;

配置模块,用于在数据流处理引擎中配置数据源输入、维表和数据输出端,其中,所述数据源输入对应所述主流数据,所述维表对应所述实时维表数据;以及

输出模块,用于将所述数据输出端输出的数据输出到数据查询系统中,以供数据查询。

9. 一种机器可读存储介质,其特征在于,该机器可读存储介质上存储有指令,该指令用

于使得机器执行权利要求1-7中任一项所述的方法。

10. 一种计算机程序产品,包括计算机程序/指令,其特征在于,该计算机程序/指令被处理器执行时实现权利要求1-7中任一项所述方法的步骤。

多流数据的关联方法和关联装置

技术领域

[0001] 本发明涉及实时数据指标领域，具体地涉及一种多流数据的关联方法和关联装置。

背景技术

[0002] 各企业在构建自己的实时数仓以及实时指标的过程中，需要进行多维度指标的整合，指标对应的数据往往存在于关系型业务库的多个表里或者消息队列中。全量实时场景往往只能作为统计，而不能提供实时的服务。流跟流进行数据关联，因网络抖动或者binlog同步延迟导致多流数据未关联上进而出现指标数据丢失，影响最终的实时数据指标准确性。目前现有的实时计算多以增量计算或者简单的关联为主，因计算的复杂度跟业务逻辑强相关，涉及到截止到当前这种全量实时场景类的指标准确性不是很高。

发明内容

[0003] 本发明实施例的目的是提供一种多流数据的关联方法和关联装置，其可解决或至少部分解决上述问题。

[0004] 为了实现上述目的，本发明实施例的一个方面提供一种多流数据的关联方法，该关联方法包括：将预设数据库中的第一预设模型的数据或系统日志放入第一预设消息队列，以得到主流数据；将所述预设数据库中的第二预设模型的数据存入数据库中，以得到实时维表数据；在数据流处理引擎中配置数据源输入、维表和数据输出端，其中，所述数据源输入对应所述主流数据，所述维表对应所述实时维表数据；以及将所述数据输出端输出的数据输出到数据查询系统中，以供数据查询。

[0005] 可选地，所述在数据流处理引擎中配置数据源输入、维表和数据输出端包括：将所述主流数据与所述实时维表数据进行关联。

[0006] 可选地，响应于将所述主流数据和所述实时维表数据进行关联时，所述主流数据与所述实时维表数据不能成功关联，该关联方法还包括：控制所述主流数据与所述实时维表数据进行延迟重试关联。

[0007] 可选地，控制所述主流数据与所述实时维表数据进行延迟重试关联后，响应于所述主流数据与所述实时维表数据仍不能成功关联，该关联方法还包括：将所述主流数据输出到第二预设消息队列中，以使得所述主流数据能够重新执行在数据流处理引擎中配置数据源输入、维表和数据输出端的操作。

[0008] 可选地，该关联方法还包括：用离线数据覆盖对应的在线数据，以修复所述在线数据的误差，其中，所述在线数据为所述数据输出端输出的数据，所述离线数据为基于所述第一预设模型的数据或系统日志以及所述第二预设模型的数据进行离线计算得到的。

[0009] 可选地，所述用离线数据覆盖对应的在线数据为采用离线T-N数据覆盖所述在线数据中与所述离线T-N数据具有相同时间范围的所述在线数据，其中，所述离线T-N数据指的是采用截止到当前时刻T的前N天获取到的所述离线数据。

[0010] 可选地,该关联方法还包括:对比离线数据与对应的在线数据是否相同,其中,所述在线数据为所述数据输出端输出的数据,所述离线数据为基于所述第一预设模型的数据或系统日志以及所述第二预设模型的数据进行离线计算得到的;以及发送对比结果。

[0011] 相应地,本发明实施例的另一方面还提供一种多流数据的关联装置,该关联装置包括:主流数据获取模块,用于将预设数据库中的第一预设模型的数据或系统日志放入第一预设消息队列,以得到主流数据;实时维表数据获取模块,用于将所述预设数据库中的第二预设模型的数据存入数据库中,以得到实时维表数据;配置模块,用于在数据流处理引擎中配置数据源输入、维表和数据输出端,其中,所述数据源输入对应所述主流数据,所述维表对应所述实时维表数据;以及输出模块,用于将所述数据输出端输出的数据输出到数据查询系统中,以供数据查询。

[0012] 可选地,所述配置模块在数据流处理引擎中配置数据源输入、维表和数据输出端包括:将所述主流数据与所述实时维表数据进行关联。

[0013] 可选地,所述配置模块还用于:响应于将所述主流数据和所述实时维表数据进行关联时,所述主流数据与所述实时维表数据不能成功关联,控制所述主流数据与所述实时维表数据进行延迟重试关联。

[0014] 可选地,该关联装置还包括:主流数据输出模块,用于控制所述主流数据与所述实时维表数据进行延迟重试关联后,响应于所述主流数据与所述实时维表数据仍不能成功关联,将所述主流数据输出到第二预设消息队列中,以使得所述主流数据能够重新执行在数据流处理引擎中配置数据源输入、维表和数据输出端的操作。

[0015] 可选地,该关联装置还包括:修复模块,用于用离线数据覆盖对应的在线数据,以修复所述在线数据的误差,其中,所述在线数据为所述数据输出端输出的数据,所述离线数据为基于所述第一预设模型的数据或系统日志以及所述第二预设模型的数据进行离线计算得到的。

[0016] 可选地,所述用离线数据覆盖对应的在线数据为采用离线T-N数据覆盖所述在线数据中与所述离线T-N数据具有相同时间范围的所述在线数据,其中,所述离线T-N数据指的是采用截止到当前时刻T的前N天获取到的所述离线数据。

[0017] 可选地,该关联装置还包括:对比模块,用于对比离线数据与对应的在线数据是否相同,其中,所述在线数据为所述数据输出端输出的数据,所述离线数据为基于所述第一预设模型的数据或系统日志以及所述第二预设模型的数据进行离线计算得到的;以及发送模块,用于发送对比结果。

[0018] 此外,本发明实施例的另一方面还提供一种机器可读存储介质,该机器可读存储介质上存储有指令,该指令用于使得机器执行上述的方法。

[0019] 另外,本发明实施例的另一方面还提供一种计算机程序产品,包括计算机程序/指令,该计算机程序/指令被处理器执行时实现上述方法的步骤。

[0020] 通过上述技术方案,通过基于主流数据和实时维表数据在数据流处理引擎中配置数据源输入、维表和数据输出端以及将数据输出端输出的数据输出到数据查询系统中供数据查询,实现了提供实时查询的服务。

[0021] 本发明实施例的其它特征和优点将在随后的具体实施方式部分予以详细说明。

附图说明

[0022] 附图是用来提供对本发明实施例的进一步理解,并且构成说明书的一部分,与下面的具体实施方式一起用于解释本发明实施例,但并不构成对本发明实施例的限制。在附图中:

[0023] 图1是本发明一实施例提供的多流数据的关联方法的流程图;

[0024] 图2是本发明另一实施例提供的实时计算数据的架构图;以及

[0025] 图3是本发明另一实施例提供的多流数据的关联装置的结构框图。

[0026] 附图标记说明

[0027] 1 主流数据获取模块 2 实时维表数据获取模块

[0028] 3 配置模块 4 输出模块

具体实施方式

[0029] 以下结合附图对本发明实施例的具体实施方式进行详细说明。应当理解的是,此处所描述的具体实施方式仅用于说明和解释本发明实施例,并不用于限制本发明实施例。

[0030] 本发明实施例的一个方面提供一种多流数据的关联方法。

[0031] 图1是本发明一实施例提供的多流数据的关联方法的流程图。如图1所示,该关联方法包括以下内容。

[0032] 在步骤S10中,将预设数据库中的第一预设模型的数据或系统日志放入第一预设消息队列,以得到主流数据。其中,预设数据库可以是mysql;第一预设模型可以是事实表。可选地,可以将mysql的binlog日志或者系统日志放入第一预设消息队列以得到主流数据。此外,第一预设消息队列可以是kafka消息队列。具体地,读取mysql的binlog日志存入kafka消息队列,或者埋点日志数据、服务器日志数据等系统日志直接存入kafka消息队列,作为数据流处理引擎计算的主流数据,也叫做事实表数据。其中,数据流处理引擎可以是flink。

[0033] 在步骤S11中,将预设数据库中的第二预设模型的数据存入数据库中,以得到实时维表数据。其中,数据库可以是hbase;第二预设模型可以是维表。

[0034] 在步骤S12中,在数据流处理引擎中配置数据源输入、维表和数据输出端,其中,数据源输入对应主流数据,维表对应实时维表数据。具体地,将数据源输入对应主流数据,将维表对应实时维表数据,将主流数据与实时维表数据进行关联;在关联以后根据业务需求基于关联进行计算得到在线数据,得到的在线数据就是数据输出端输出的数据。其中,数据流处理引擎可以是flink;数据源输入可以是source输入;数据输出端可以是sink输出。

[0035] 在步骤S13中,将数据输出端输出的数据输出到数据查询系统中,以供数据查询。其中,数据查询系统可以是OLAP引擎。例如,将sink输出输出的数据输出到druid,doris,clickhouse(三选一)这类OLAP引擎中,作为api服务对外提供数据查询。

[0036] 通过上述技术方案,通过基于主流数据和实时维表数据在数据流处理引擎中配置数据源输入、维表和数据输出端以及将数据输出端输出的数据输出到数据查询系统中供数据查询,实现了提供实时查询的服务。

[0037] 此外,在本发明实施例中,还提供了针对多流数据关联的数据准确性的优化策略,提升在线数据的准确度。

[0038] 可选地,在本发明实施例中,在数据流处理引擎中配置数据源输入、维表和数据输出端包括:将主流数据与实时维表数据进行关联。例如,基于interval join利用watermark加窗口机制将主流数据与实时维表数据进行关联。基于interval join利用watermark加窗口机制进行关联可以保证更多的主流数据与实时维表数据能够关联上。此外,基于interval join利用watermark+窗口机制实现跨窗口的关联,适用于双流join,可以解决消息乱序以及消息延迟导致主流数据未关联上进而出现主流数据丢失从而影响最终的在线数据准确性的问题,提高了最终的在线数据的准确性。另外,在采用interval join这种关联方式时,需要配置watermark延迟的时间参数以及窗口的大小,通常窗口的大小为1分钟。

[0039] 可选地,在本发明实施例中,响应于将主流数据和实时维表数据进行关联时,主流数据与实时维表数据不能成功关联,该关联方法还包括:控制主流数据与实时维表数据进行延迟重试关联。其中,延迟重试的时间可以根据情况设置。另外,延迟重试关联的次数也可以根据情况进行设置。通过对主流数据与实时维表数据关联选择延迟重试,来保障主流数据与实时维表数据能够关联上,解决因数据未关联上导致主流数据丢失进而影响最终的在线数据准确性的问题,提高了最终的在线数据的准确性。

[0040] 可选地,在本发明实施例中,控制主流数据与实时维表数据进行延迟重试关联后,响应于主流数据与实时维表数据仍不能成功关联,该关联方法还包括:将主流数据输出到第二预设消息队列中,以使得主流数据能够重新执行在数据流处理引擎中配置数据源输入、维表和数据输出端的操作。将未关联上的主流数据输出到专门接收未关联上实时维表数据的主流数据的消息队列中,例如未关联上实时维表数据的主流数据放入到一消息队列的预设位置处,用于通过回放消息队列使得未关联上实时维表数据的主流数据能够重新执行在数据流处理引擎中配置数据源输入、维表和数据输出端的操作,重新参与计算,解决了主流数据丢失影响最终的在线数据准确性的问题,提高了最终的在线数据的准确性。可选地,第二预设消息队列可以是kafka消息队列。

[0041] 可选地,在本发明实施例中,该关联方法还包括:用离线数据覆盖对应的在线数据,以修复在线数据的误差,其中,在线数据为数据输出端输出的数据,离线数据为基于第一预设模型的数据或系统日志以及第二预设模型的数据进行离线计算得到的。可选地,离线数据可以是利用hive表进行计算得到的。在本发明实施例中,数据输出端输出的数据即为在线数据,离线数据与在线数据是采用相同的源数据和不同的计算方法得到的,在线数据是基于数据流处理引擎实时计算得到的,离线数据是将源数据存储于离线数据仓库例如HiveSQL中之后,通过离线处理计算得到的,其中,源数据即为预设数据库中的第一预设模型的数据或系统日志以及预设数据库中的第二预设模型的数据。

[0042] 此外,在采用离线数据覆盖在线数据时,得到对应的离线数据和在线数据的源数据采用相同时间节点上的源数据,与离线数据对应的在线数据指的是采用相同时间节点上的源数据计算得到的在线数据。可选地,用离线数据覆盖对应的在线数据为采用离线T-N数据覆盖在线数据中与离线T-N数据具有相同时间范围的在线数据,其中,离线T-N数据指的是采用截止到当前时刻T的前N天获取到的离线数据。被离线T-N数据覆盖的对应的在线数据也是采用截止到当前时刻T的前N天获取到的在线数据。其中,可以根据实际情况设置N的大小。例如,可以是采用离线T-1数据覆盖对应的在线数据,离线T-1数据是采用截止到当前时刻的前一天获取到的所有的源数据计算得到的离线数据,被覆盖的在线数据也是采用截

止到当前时刻的前一天获取到的所有的源数据计算得到的在线数据。其中,源数据即为预设数据库中的第一预设模型的数据或系统日志以及预设数据库中的第二预设模型的数据。离线数据准确性高,通过使用离线数据覆盖对应的在线数据,可以解决主流数据丢失影响最终的在线数据准确性的问题,提高了最终的在线数据的准确性。另外,相比于基于interval join利用watermark+窗口机制实现跨窗口的关联、延迟重试关联和/或将未关联上的主流数据输出到专门接收未关联上实时维表数据的主流数据的消息队列中,使用离线数据覆盖对应的在线数据,可以更好的解决因主流数据丢失影响最终的在线数据准确性的问题,提高最终的在线数据的准确性。此外,通过将离线数据覆盖在线数据,提高了截止到当前这种全量实时场景类的在线数据的准确性。另外,使用离线数据覆盖在线数据可以是每隔一段时间进行一次,例如,每周进行一次,在线数据每周基于离线T-1数据进行一次合并(merge),用来修复在线数据的误差。

[0043] 具体地,在本发明实施例中,用离线数据覆盖对应的在线数据,以修复在线数据的误差可以包括以下内容。其中,在下面的示例中,使用离线T-1数据代替对应的在线数据,需要说明的是,采用其他时间节点的源数据计算出来的离线数据代替对应的采用相同时间节点的源数据计算出来的在线数据的时候也可以参照下述内容实现。a). 停止实时计算任务,也就是停止在flink中配置source输入、维表和sink输出。具体地,调用flink的系统接口,输入任务ID以及用户ID,将实时计算任务的状态置为停止。其中,任务ID是在flink中配置source输入、维表和sink输出的实时计算任务的识别标识,用户ID是在flink中提交了实时计算任务的用户的身份识别标识。b). 用离线T-1数据覆盖对应的在线数据。具体地,用截止到当前时刻的前一天获取到的所有的源数据基于hive表计算得到离线T-1数据,用截止到当前时刻的前一天获取到的所有的源数据基于flink计算得到该离线T-1数据对应的在线数据,用离线T-1数据覆盖该在线数据。c). 启动实时计算任务,并重置kafka的偏移量参数(offset),进行kafka消息回放,其中,消息回放就是kafka消息队列中的数据重新参与计算。具体地,通过flink系统接口,定义好任务ID和用户ID以及要重置的kafka消息队列的topic(即,在执行用离线T-1数据覆盖对应的在线数据之前,kafka消息队列中正在参与实时计算任务的数据的位置),随后即重新开始实时计算任务。例如,当前时刻为2022年3月31日,当前时刻的前一天为2022年3月30日,开始进行数据计算的起始时间为2022年3月1日;离线T-1数据为从2022年3月1日至2022年3月30日之间获取到的离线数据,用从2022年3月1日至2022年3月30日之间获取到的离线数据覆盖从2022年3月1日至2022年3月30日之间获取到的在线数据;进行覆盖后,设置offset对应2022年3月31日0:00,kafka消息队列中从设置的该时间节点之后的数据重新参与实时计算任务。此外,在本发明实施例中,开始进行数据计算的起始时间根据具体情况而定。

[0044] 可选地,在本发明实施例中,该关联方法还包括:对比离线数据与对应的在线数据是否相同,其中,在线数据为数据输出端输出的数据,离线数据为基于第一预设模型的数据或系统日志以及第二预设模型的数据计算得到的;以及发送对比结果。其中,关于离线数据及对应的在线数据的解释可以参见上述实施例中所述的内容。具体地,基于OLAP引擎(druid,clickhouse,doris)通过python脚本调用对比sql将离线数据与在线数据进行对比。另外,得到对比结果可以通过企业微信机器人的形式每天早晨8点定时发送给用户。具体地,在本发明实施例中,可以是针对源数据选取昨天的日期,也就是源数据采用截止到当

前时刻的前一天获取到的源数据。利用源数据计算出离线T-1数据。用户可以设置需要用于对比的在线数据对应的任务ID和用户ID,获取设置的用户ID提交的flink实时计算任务中与设置的任务ID对应的实时计算任务基于截止到当前时刻的前一天获取到的源数据计算得到的在线数据。将离线T-1数据与得到的在线数据进行对比,将得到的对比结果通过企业微信机器人的形式每天早晨8点定时发送。如此,可以实现对在线数据指标准确性监控,用于了解在线数据的准确性。

[0045] 图2是本发明另一实施例提供的实时计算数据的架构图。其中,多流数据关联主要包括以下内容。1)以EPX读取mysql的A模型的binlog日志存入kafka消息队列,或者埋点日志数据、服务器日志数据等系统日志直接存入kafka队列,作为flink计算的主流数据,也叫做事实表数据。2)需要数据如mysql的B模型的数据基于binlog存入hbase中,作为实时维表数据,用来补充维度数据,或者将mysql中的事实表进行维度化。3)在flink中配置好source输入、维表、sink输出三部分。source端即为步骤1)中的主流数据,维表即为步骤2)中的实时维表数据。将sink输出输出的数据输出到druid,doris,clickhouse(三选一)这类OLAP引擎中,作为api服务对外提供数据查询。4)将在进行在flink中进行计算的源数据通过数据快照进行复制,基于hive表计算得到离线数据。其中,源数据包括mysql的A模型的binlog日志或者埋点日志数据、服务器日志数据等系统日志以及mysql的B模型的数据。如此,本发明实施例中提供了实时计算的数据架构,提供端到端的数据计算流程。

[0046] 综上所述,本发明实施例的技术方案提供了一套实时计算的数据架构,并基于多表(多流)关联的数据准确性做进一步的优化提升,提供截止到当前全量场景的实时指标,以及查询服务。本发明实施例提供的技术方案应用在提供稳定、准确的在线数据指标领域,为业务的实时策略以及在线数据展示提供技术支持。

[0047] 相应地,本发明实施例的另一方面还提供多流数据的关联装置。

[0048] 图3是本发明另一实施例提供的多流数据的关联装置。如图3所示,该关联装置包括主流数据获取模块1、实时维表数据获取模块2、配置模块3和输出模块4。其中,主流数据获取模块1用于将预设数据库中的第一预设模型的数据或系统日志放入第一预设消息队列,以得到主流数据;实时维表数据获取模块4用于将预设数据库中的第二预设模型的数据存入数据库中,以得到实时维表数据;配置模块2用于在数据流处理引擎中配置数据源输入、维表和数据输出端,其中,数据源输入对应主流数据,维表对应实时维表数据;输出模块3用于将数据输出端输出的数据输出到数据查询系统中,以供数据查询。

[0049] 可选地,在本发明实施例中,配置模块在数据流处理引擎中配置数据源输入、维表和数据输出端包括:将主流数据与实时维表数据进行关联。

[0050] 可选地,在本发明实施例中,配置模块还用于:响应于将主流数据和实时维表数据进行关联时,主流数据与实时维表数据不能成功关联,控制主流数据与实时维表数据进行延迟重试关联。

[0051] 可选地,在本发明实施例中,该关联装置还包括:主流数据输出模块,用于控制主流数据与实时维表数据进行延迟重试关联后,响应于主流数据与实时维表数据仍不能成功关联,将主流数据输出到第二预设消息队列中,以使得主流数据能够重新执行在数据流处理引擎中配置数据源输入、维表和数据输出端的操作。

[0052] 可选地,在本发明实施例中,该关联装置还包括:修复模块,用于用离线数据覆盖

对应的在线数据,以修复在线数据的误差,其中,在线数据为数据输出端输出的数据,离线数据为基于第一预设模型的数据或系统日志以及第二预设模型的数据进行离线计算得到的。

[0053] 可选地,在本发明实施例中,用离线数据覆盖对应的在线数据为采用离线T-N数据覆盖在线数据中与离线T-N数据具有相同时间范围的在线数据,其中,离线T-N数据指的是采用截止到当前时刻T的前N天获取到的离线数据。

[0054] 可选地,在本发明实施例中,该关联装置还包括:对比模块,用于对比离线数据与对应的在线数据是否相同,其中,在线数据为数据输出端输出的数据,离线数据为基于第一预设模型的数据或系统日志以及第二预设模型的数据进行离线计算得到的;以及发送模块,用于发送对比结果。

[0055] 本发明实施例提供的多流数据的关联装置的具体工作原理及益处与本发明实施例提供的多流数据的关联方法的具体工作原理及益处相似,这里将不再赘述。

[0056] 所述多流数据的关联装置包括处理器和存储器,上述主流数据获取模块、实时维表数据获取模块、配置模块和输出模块等均作为程序单元存储在存储器中,由处理器执行存储在存储器中的上述程序单元来实现相应的功能。

[0057] 处理器中包含内核,由内核去存储器中调取相应的程序单元。内核可以设置一个或以上,通过调整内核参数来提供实时查询的服务。

[0058] 存储器可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM),存储器包括至少一个存储芯片。

[0059] 本发明实施例的另一方面还提供了一种机器可读存储介质,该机器可读存储介质上存储有指令,该指令用于使得机器执行上述实施例中所述的方法。

[0060] 本发明实施例的另一方面还提供了一种处理器,所述处理器用于运行程序,其中,所述程序运行时执行上述实施例中所述的方法。

[0061] 本发明实施例的另一方面还提供了一种设备,设备包括处理器、存储器及存储在存储器上并可在处理器上运行的程序,处理器执行程序时实现上述实施例中所述的方法。

[0062] 本发明实施例的另一方面还提供了一种计算机程序产品,包括计算机程序/指令,该计算机程序/指令被处理器执行时实现上述实施例中所述的方法的步骤。

[0063] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0064] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0065] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0066] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0067] 在一个典型的配置中,计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0068] 存储器可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。存储器是计算机可读介质的示例。

[0069] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[0070] 还需要说明的是,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括要素的过程、方法、商品或者设备中还存在另外的相同要素。

[0071] 以上仅为本申请的实施例而已,并不用于限制本申请。对于本领域技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本申请的权利要求范围之内。

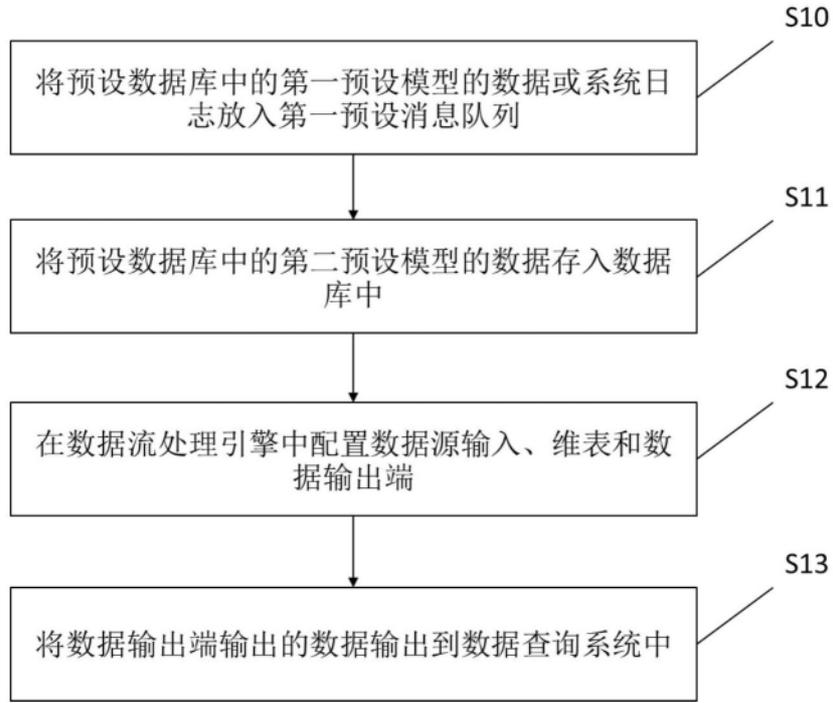


图1

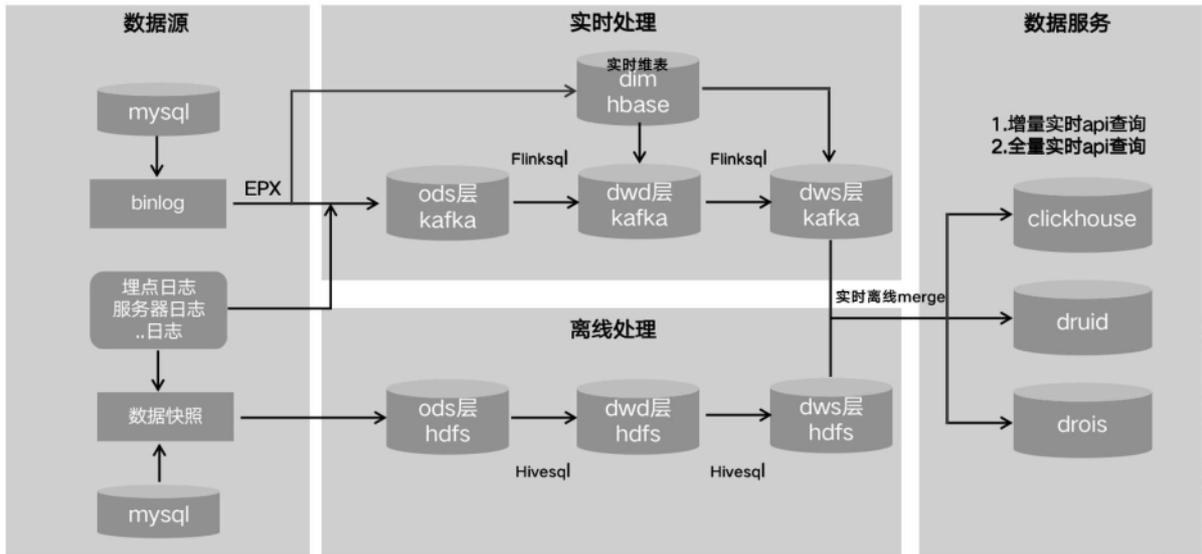


图2

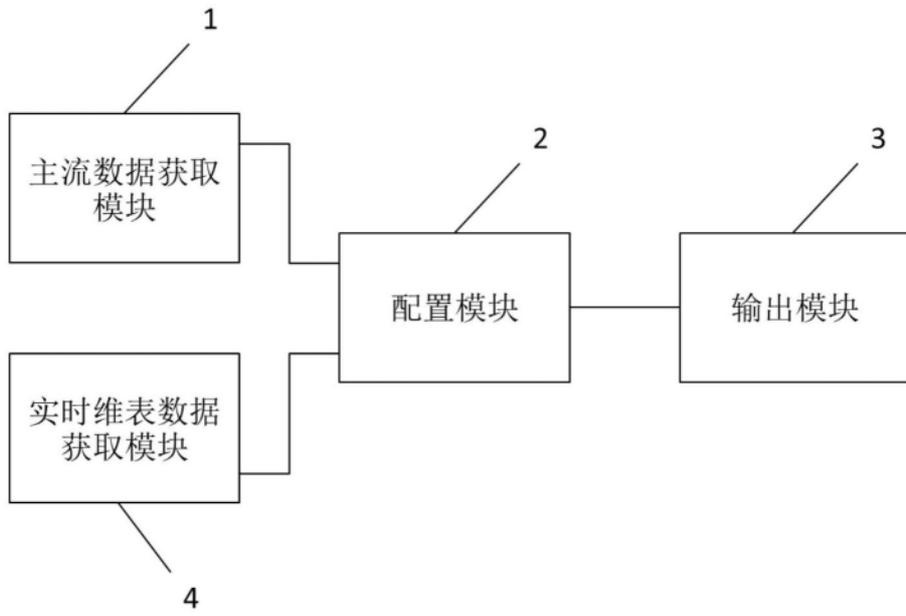


图3