(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 115328928 B
(45)授权公告日 2023.07.25

(21)申请号 202210972152.2

(22)申请日 2022.08.15

(65)同一申请的已公布的文献号
申请公布号 CN 115328928 A

(43)申请公布日 2022.11.11

(73)专利权人 深圳大道云科技有限公司
地址 518000 广东省深圳市前海深港合作区前湾一路1号A栋201室(入驻深圳市前海商务秘书有限公司)

(72)发明人 徐约可 谢国斌 马明 李环良

(74)专利代理机构 深圳市特讯知识产权代理事务所(普通合伙) 44653
专利代理师 孟智广

(51)Int.Cl.
G06F 16/23(2019.01)
G06F 16/22(2019.01)
G06F 16/27(2019.01)
G06F 16/28(2019.01)

审查员 杨洁

权利要求书2页 说明书7页 附图2页

(54)发明名称

kudu表更新方法、装置、设备及存储介质

(57)摘要

本发明涉及表格更新领域，公开了一种kudu表更新方法、装置、设备及存储介质。该方法包括：基于预置table.json配置文件，触发FlinkCDC任务，生成预置kudu表的同步指令；根据所述同步指令，对预置监控日志进行解析转换处理，得到所述监控日志对应的JSON串；根据预置转换算法，将所述JSON串转换为Java对象数据；判断所述table.json配置文件中的配置字段与kudu表中对应字段是否存在修改；若存在修改，则根据所述Java对象数据，对所述kudu表进行映射匹配修改处理，生成更新的kudu表。

CN 115328928 B

1.一种kudu表更新方法,其特征在于,包括步骤:

基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令;

根据所述同步指令,对预置监控日志进行解析转换处理,得到所述监控日志对应的JSON串;

根据预置转换算法,将所述JSON串转换为Java对象数据;

判断所述table.json配置文件中的配置字段与kudu表中对应字段是否存在修改;

若存在修改,则根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表;

其中,在所述基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令之后,在所述根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表之前,还包括:

基于所述table.json配置文件,获取kudu服务组件的会话数据;

将所述会话数据对应所述kudu表的数据转换为K-V形式存储在Map容器中;

其中,在所述基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令之后,在所述根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表之前,还包括:

根据所述table.json配置文件,对预置kudu创建操作组件进行设置处理,获取数据操作设置;

将所述数据操作设置转换为K-V形式存储在Map容器中。

2.根据权利要求1所述的kudu表更新方法,其特征在于,在所述基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令之前,还包括:

获取table.json配置文件,访问预置MySQL数据库,解析所述table.json配置文件对应的数据,生成kudu建表语句;

基于预置Impala组件,对所述kudu建表语句进行执行处理,生成kudu表;

执行预置DataX任务数据,将历史数据同步至所述kudu表中。

3.根据权利要求1所述的kudu表更新方法,其特征在于,所述根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表包括:

根据所述Java对象数据,对所述Map容器进行查询处理,得到所述Java对象数据对应的数据操作设置、所述kudu表的数据;

将所述Java对象数据转换为K-V形式数据,得到Java对象K-V数据;

根据所述Java对象K-V数据,对所述Java对象数据对应的数据操作设置、所述kudu表的数据进行修改处理,得到Map容器的映射修改数据;

基于所述映射修改数据,对所述kudu表进行修改处理,生成更新的kudu表。

4.根据权利要求1所述的kudu表更新方法,其特征在于,在所述根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表之后,还包括:

基于预置Impala组件,对所述更新的kudu表进行查询处理,得到更新日志,以及将所述更新日志上传至云端数据库。

5.根据权利要求1所述的kudu表更新方法,其特征在于,所述基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令包括:

基于预置table.json配置文件，通过StreamX组件读取FlinkCDC任务，生成所述table.json配置文件对应的kudu表的同步指令。

6.一种kudu表更新装置，其特征在于，所述kudu表更新装置包括：

指令生成模块，用于基于预置table.json配置文件，触发FlinkCDC任务，生成预置kudu表的同步指令；

日志处理模块，用于根据所述同步指令，对预置监控日志进行解析转换处理，得到所述监控日志对应的JSON串；

对象转换模块，用于根据预置转换算法，将所述JSON串转换为Java对象数据；

判断模块，用于判断所述table.json配置文件中的配置字段与kudu表中对应字段是否存在修改；

匹配修改模块，用于若存在修改，则根据所述Java对象数据，对所述kudu表进行映射匹配修改处理，生成更新的kudu表；

其中，所述kudu表更新装置还包括会话转换模块，所述会话转换模块具体用于：

基于所述table.json配置文件，获取kudu服务组件的会话数据；

将所述会话数据对应所述kudu表的数据转换为K-V形式存储在Map容器中；

其中，所述kudu表更新装置还包括设置转换模块，所述设置转换模块具体用于：

根据所述table.json配置文件，对预置kudu创建操作组件进行设置处理，获取数据操作设置；

将所述数据操作设置转换为K-V形式存储在Map容器中。

7.一种kudu表更新设备，其特征在于，所述kudu表更新设备包括：存储器和至少一个处理器，所述存储器中存储有指令，所述存储器和所述至少一个处理器通过线路互连；

所述至少一个处理器调用所述存储器中的所述指令，以使得所述kudu表更新设备执行如权利要求1-5中任一项所述的kudu表更新方法。

8.一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现如权利要求1-5中任一项所述的kudu表更新方法。

kudu表更新方法、装置、设备及存储介质

技术领域

[0001] 本发明涉及表格更新领域，尤其涉及一种kudu表更新方法、装置、设备及存储介质。

背景技术

[0002] 现有的基于mysql binlog数据实时同步方案中mysql-Streamsets-kudu，框架较重，耗费资源，不可定制，出现同步问题不太方便查找；mysql-canal/maxwell/debezium-kafka-flink-kudu，组件较多，开发和维护成本较高；Flinksql (Flink CDC + Apache Bahir Flink Kudu Connector)一个Job中同步多张表受限且需要重复写Schema比较麻烦等；现有方案在数据实时同步过程中，源数据一有变更便会触发kudu的写操作，而在我们实际数据分析中并不需要一有binlog产生就触发写操作，而是按需(我们关心的字段信息发生变更)触发写操作，这样也可以大大降低数据洪峰时，写操作产生压力；再者现有方案实现数据同步一般是对于数据表的增删改，也就是kudu表中存储的是此时此刻的一份快照，而我们对有一些重要的信息表的数据需要详细记录每条信息增删改的详细变更过程以便后续审计溯源等。因此，针对当前kudu表格的修改内容变更无法简易被溯源的技术问题，需要一种新的更新kudu表的技术。

发明内容

[0003] 本发明的主要目的在于解决当前kudu表格的修改内容变更无法简易被溯源的技术问题。

[0004] 本发明第一方面提供了一种kudu表更新方法，所述kudu表更新方法包括：

[0005] 基于预置table.json配置文件，触发FlinkCDC任务，生成预置kudu表的同步指令；

[0006] 根据所述同步指令，对预置监控日志进行解析转换处理，得到所述监控日志对应的JSON串；

[0007] 根据预置转换算法，将所述JSON串转换为Java对象数据；

[0008] 判断所述table.json配置文件中的配置字段与kudu表中对应字段是否存在修改；

[0009] 若存在修改，则根据所述Java对象数据，对所述kudu表进行映射匹配修改处理，生成更新的kudu表。

[0010] 可选的，在本发明第一方面的第一种实现方式中，在所述基于预置table.json配置文件，触发FlinkCDC任务，生成预置kudu表的同步指令之前，还包括：

[0011] 获取table.json配置文件，访问预置MySQL数据库，解析所述table.json配置文件对应的数据，生成kudu建表语句；

[0012] 基于预置Impala组件，对所述kudu建表语句进行执行处理，生成kudu表；

[0013] 执行预置DataX任务数据，将历史数据同步至所述kudu表中。

[0014] 可选的，在本发明第一方面的第二种实现方式中，在所述基于预置table.json配置文件，触发FlinkCDC任务，生成预置kudu表的同步指令之后，在所述根据所述Java对象数

据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表之前,还包括:

[0015] 基于所述table.json配置文件,获取kudu服务组件的会话数据;

[0016] 将所述会话数据对应所述kudu表的数据转换为K-V形式存储在Map容器中。

[0017] 可选的,在本发明第一方面的第三种实现方式中,在所述基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令之后,在所述根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表之前,还包括:

[0018] 根据所述table.json配置文件,对预置kudu创建操作组件进行设置处理,获取数据操作设置;

[0019] 将所述数据操作设置转换为K-V形式存储在Map容器中。

[0020] 可选的,在本发明第一方面的第四种实现方式中,所述根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表包括:

[0021] 根据所述Java对象数据,对所述Map容器进行查询处理,得到所述Java对象数据对应的数据操作设置、所述kudu表的数据;

[0022] 将所述Java对象数据转换为K-V形式数据,得到Java对象K-V数据;

[0023] 根据所述Java对象K-V数据,对所述Java对象数据对应的数据操作设置、所述kudu表的数据进行修改处理,得到Map容器的映射修改数据;

[0024] 基于所述映射修改数据,对所述kudu表进行修改处理,生成更新的kudu表。

[0025] 可选的,在本发明第一方面的第五种实现方式中,在所述根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表之后,还包括:

[0026] 基于预置Impala组件,对所述更新的kudu表进行查询处理,得到更新日志,以及将所述更新日志上传至云端数据库。

[0027] 可选的,在本发明第一方面的第六种实现方式中,所述基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令包括:

[0028] 基于预置table.json配置文件,通过StreamX组件读取FlinkFlinkCDC任务,生成所述table.json配置文件对应的kudu表的同步指令。

[0029] 本发明第二方面提供了一种kudu表更新装置,所述kudu表更新装置包括:

[0030] 指令生成模块,用于基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令;

[0031] 日志处理模块,用于根据所述同步指令,对预置监控日志进行解析转换处理,得到所述监控日志对应的JSON串;

[0032] 对象转换模块,用于根据预置转换算法,将所述JSON串转换为Java对象数据;

[0033] 判断模块,用于判断所述table.json配置文件中的配置字段与kudu表中对应字段是否存在修改;

[0034] 匹配修改模块,用于若存在修改,则根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表。

[0035] 本发明第三方面提供了一种kudu表更新设备,包括:存储器和至少一个处理器,所述存储器中存储有指令,所述存储器和所述至少一个处理器通过线路互连;所述至少一个处理器调用所述存储器中的所述指令,以使得所述kudu表更新设备执行上述的kudu表更新方法。

[0036] 本发明的第四方面提供了一种计算机可读存储介质，所述计算机可读存储介质中存储有指令，当其在计算机上运行时，使得计算机执行上述的kudu表更新方法。

[0037] 在本发明实施例中，当有新的多张数据库表需要加入实时同步时，无需再有代码开发，只需简单地配置，便可轻松实现mysql-kudu的实时同步；再者本方案可以按需字段信息发生变更触发写操作，这样也可以大大降低数据洪峰时，写操作产生压力，该方案集成阿里云日志可以方便定位数据从产生到落盘的完整过程，并且集成了StreamX可以很方便的部署和管理，并且通过Impala可以快速的实时查阅Kudu表数据情况。

附图说明

[0038] 图1为本发明实施例中kudu表更新方法的一个实施例示意图；

[0039] 图2为本发明实施例中kudu表更新装置的一个实施例示意图；

[0040] 图3为本发明实施例中kudu表更新装置的另一个实施例示意图；

[0041] 图4为本发明实施例中kudu表更新设备的一个实施例示意图。

具体实施方式

[0042] 本发明实施例提供了一种kudu表更新方法、装置、设备及存储介质。

[0043] 本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”、“第三”、“第四”等(如果存在)是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实施。此外，术语“包括”或“具有”及其任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0044] 为便于理解，下面对本发明实施例的具体流程进行描述，请参阅图1，本发明实施例中kudu表更新方法的一个实施例包括：

[0045] 101、基于预置table.json配置文件，触发FlinkCDC任务，生成预置kudu表的同步指令；

[0046] 在本实施例中，该方案分为两部分，一部分是初始化过程，根据配置文件进行的批量历史数据同步过程；另一部分是基于mysql binlog的实时数据同步过程。首先通过初始化建表之后，开启FlinkCDC实时同步任务，完成保存点savepoint之后关闭FlinkCDC任务，执行历史数据同步DataX任务，完成历史数据同步之后，再从保存点savepoint将FlinkCDC任务启动起来。

[0047] table.json配置文件包含库，表，字段名称等信息，解析eble.json配置文件，即可触发开启FlinkCDC任务，生成对目标kudu表的同步指令。

[0048] 进一步的，在101之前还包括以下步骤：

[0049] 1011、获取table.json配置文件，访问预置MySQL数据库，解析table.json配置文件对应的数据，生成kudu建表语句；

[0050] 1012、基于预置Impala组件，对kudu建表语句进行执行处理，生成kudu表；

[0051] 1013、执行预置DataX任务数据，将历史数据同步至kudu表中。

[0052] 在1011-1013步骤中，根据配置文件信息访问MySQL数据库，解析对应表的元数据信息，映射其字段数据类型，生成Kudu的建表语句，以及批量数据同步的DataX任务，完成之后采用Impala组件执行建表语句生成Kudu表，接着启动DataX任务完成历史数据同步kudu表中从而完成初始化。

[0053] 进一步的，101可以执行以下步骤：

[0054] 1014、基于预置table.json配置文件，通过StreamX组件读取FlinkFlinkCDC任务，生成table.json配置文件对应的kudu表的同步指令。

[0055] 在1014步骤中，mysql binlog的实时数据同步过程中，通过StreamX组件管理Flink的实时同步任务，然后基于StreamX组件读取FlinkFlinkCDC任务，通过table.json配置文件监控指定库的日志信息。

[0056] 102、根据同步指令，对预置监控日志进行解析转换处理，得到监控日志对应的JSON串；

[0057] 在本实施例中，在同步指令下，通过对table.json配置文件对应的库、表、字段，监控对应库的监控日志，转换成JOSN串。例如：mysql binlog通过反序列化器生成指定格式的Json串。

[0058] 103、根据预置转换算法，将JSON串转换为Java对象数据；

[0059] 在本实施例中，JSON串并通过扁平化映射转成Java对象，接着在过滤器中，读取配置文件表字段信息，对比变更前后配置字段的信息是否发生了变化，没有变化则舍弃，有变化才继续向下游传输，减小数据洪峰时数据的写压力，之后输出Kudu。

[0060] 104、判断table.json配置文件中的配置字段与kudu表中对应字段是否存在修改；

[0061] 在本实施例中，在过滤器中，读取配置文件表字段信息，对比变更前后配置字段的信息是否发生了变化，没有变化则舍弃，有变化才继续向下游传输。

[0062] 进一步的，在101步骤之后，在105步骤之前，还可以执行以下步骤：

[0063] 1041、基于table.json配置文件，获取kudu服务组件的会话数据；

[0064] 1042、将会话数据对应kudu表的数据转换为K-V形式存储在Map容器中。

[0065] 在1041-1042步骤中，输出Kudu时，首先会初始化Kudu Client以及会话数据，接着通过会话数据获取要同步的Kudu表的表格信息并以k-v的形式保存于Map容器中。

[0066] 进一步的，在101步骤之后，在105步骤之前，还可以执行以下步骤：

[0067] 1043、根据table.json配置文件，对预置kudu创建操作组件进行设置处理，获取数据操作设置；

[0068] 1044、将数据操作设置转换为K-V形式存储在Map容器中。

[0069] 该1043-1044步骤与1041-1042步骤可以并行处理，主要是创建各表的Kudu Operation Mapper确定以何种方式进行数据操作并以k-v的形式保存于Map容器中。可以增删改实时同步，也可以将这些记录都转成插入，并在目标表中增加操作类型及操作时间等字段记录数据的详细变化过程。

[0070] 105、若存在修改，则根据Java对象数据，对kudu表进行映射匹配修改处理，生成更新的kudu表。

[0071] 在本实施例中，Java对象数据中包含映射对象的库名、表名、字段，对kudu表在匹配的库名、表名、字段进行更新，生成更新的kudu表。

[0072] 进一步的,在105步骤可以执行以下步骤:

[0073] 1051、根据Java对象数据,对Map容器进行查询处理,得到Java对象数据对应的数据操作设置、kudu表的数据;

[0074] 1052、将Java对象数据转换为K-V形式数据,得到Java对象K-V数据;

[0075] 1053、根据Java对象K-V数据,对Java对象数据对应的数据操作设置、kudu表的数据进行修改处理,得到Map容器的映射修改数据;

[0076] 1054、基于映射修改数据,对kudu表进行修改处理,生成更新的kudu表。

[0077] 在1051-1054步骤中,完成Kudu Sink初始化之后,基于上游传输来的Java对象获取原表名称然后去初始化的Map容器中获取对应的Kudu表格数据和数据操作设置(Kudu Operation Mapper)。根据Java对象中的业务数据获取字段名称对数据类型进行转化映射使之与Kudu表数据类型匹配,完成之后触发Kudu的写操作,实时写入的数据,可以通过Impala进行查询;整个实时同步过程通过StreamX进行部署管理,当其他库表需要同步时,仅需在配置文件中添加相关信息并通过StreamX的CICD即可,无需再有代码开发。

[0078] 进一步的,在105步骤之后,还可以执行以下步骤:

[0079] 106、基于预置Impala组件,对更新的kudu表进行查询处理,得到更新日志,以及将更新日志上传至云端数据库。

[0080] 在本实施例中,该过程集成了阿里云日志,mysql binlog产生之后在反序列化的过程中会实时的加上一个唯一的trace ID,在之后的扁平化映射、过滤、写Kudu的过程中都会保留,当我们需要查看具体的一条记录的增删改情况,只需登录阿里云日志平台,输入对应的traceID便可查看完整的数据变更过程,方便我们定位问题。

[0081] 在本发明实施例中,当有新的多张数据库表需要加入实时同步时,无需再有代码开发,只需简单地配置,便可轻松实现mysql-kudu的实时同步;再者本方案可以按需字段信息发生变更触发写操作,这样也可以大大降低数据洪峰时,写操作产生压力,该方案集成阿里云日志可以方便定位数据从产生到落盘的完整过程,并且集成了StreamX可以很方便的部署和管理,并且通过Impala可以快速的实时查阅Kudu表数据情况。

[0082] 上面对本发明实施例中kudu表更新方法进行了描述,下面对本发明实施例中kudu表更新装置进行描述,请参阅图2,本发明实施例中kudu表更新装置一个实施例包括:

[0083] 指令生成模块201,用于基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令;

[0084] 日志处理模块202,用于根据所述同步指令,对预置监控日志进行解析转换处理,得到所述监控日志对应的JSON串;

[0085] 对象转换模块203,用于根据预置转换算法,将所述JSON串转换为Java对象数据;

[0086] 判断模块204,用于判断所述table.json配置文件中的配置字段与kudu表中对应字段是否存在修改;

[0087] 匹配修改模块205,用于若存在修改,则根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表。

[0088] 在本发明实施例中,当有新的多张数据库表需要加入实时同步时,无需再有代码开发,只需简单地配置,便可轻松实现mysql-kudu的实时同步;再者本方案可以按需字段信息发生变更触发写操作,这样也可以大大降低数据洪峰时,写操作产生压力,该方案集成阿

里云日志可以方便定位数据从产生到落盘的完整过程,并且集成了StreamX可以很方便的部署和管理,并且通过Impala可以快速的实时查阅Kudu表数据情况。

[0089] 请参阅图3,本发明实施例中kudu表更新装置的另一个实施例包括:

[0090] 指令生成模块201,用于基于预置table.json配置文件,触发FlinkCDC任务,生成预置kudu表的同步指令;

[0091] 日志处理模块202,用于根据所述同步指令,对预置监控日志进行解析转换处理,得到所述监控日志对应的JSON串;

[0092] 对象转换模块203,用于根据预置转换算法,将所述JSON串转换为Java对象数据;

[0093] 判断模块204,用于判断所述table.json配置文件中的配置字段与kudu表中对应字段是否存在修改;

[0094] 匹配修改模块205,用于若存在修改,则根据所述Java对象数据,对所述kudu表进行映射匹配修改处理,生成更新的kudu表。

[0095] 其中,所述kudu表更新装置还包括初始化模块206,所述初始化模块206具体用于:

[0096] 获取table.json配置文件,访问预置MySQL数据库,解析所述table.json配置文件对应的数据,生成kudu建表语句;

[0097] 基于预置Impala组件,对所述kudu建表语句进行执行处理,生成kudu表;

[0098] 执行预置DataX任务数据,将历史数据同步至所述kudu表中。

[0099] 其中,所述kudu表更新装置还包括会话转换模块207,所述会话转换模块207具体用于:

[0100] 基于所述table.json配置文件,获取kudu服务组件的会话数据;

[0101] 将所述会话数据对应所述kudu表的数据转换为K-V形式存储在Map容器中。

[0102] 其中,所述kudu表更新装置还包括设置转换模块208,所述设置转换模块208具体用于:

[0103] 根据所述table.json配置文件,对预置kudu创建操作组件进行设置处理,获取数据操作设置;

[0104] 将所述数据操作设置转换为K-V形式存储在Map容器中。

[0105] 其中,所述匹配修改模块205具体用于:

[0106] 根据所述Java对象数据,对所述Map容器进行查询处理,得到所述Java对象数据对应的数据操作设置、所述kudu表的数据;

[0107] 将所述Java对象数据转换为K-V形式数据,得到Java对象K-V数据;

[0108] 根据所述Java对象K-V数据,对所述Java对象数据对应的数据操作设置、所述kudu表的数据进行修改处理,得到Map容器的映射修改数据;

[0109] 基于所述映射修改数据,对所述kudu表进行修改处理,生成更新的kudu表。

[0110] 其中,所述kudu表更新装置还包括云端上传模块209,所述云端上传模块209具体用于:

[0111] 基于预置Impala组件,对所述更新的kudu表进行查询处理,得到更新日志,以及将所述更新日志上传至云端数据库。

[0112] 其中,所述指令生成模块201具体用于:

[0113] 基于预置table.json配置文件,通过StreamX组件读取FlinkCDC任务,生成所述

table.json配置文件对应的kudu表的同步指令。

[0114] 在本发明实施例中，当有新的多张数据库表需要加入实时同步时，无需再有代码开发，只需简单地配置，便可轻松实现mysql-kudu的实时同步；再者本方案可以按需字段信息发生变更触发写操作，这样也可以大大降低数据洪峰时，写操作产生压力，该方案集成阿里云日志可以方便定位数据从产生到落盘的完整过程，并且集成了StreamX可以很方便的部署和管理，并且通过Impala可以快速的实时查阅Kudu表数据情况。

[0115] 上面图2和图3从模块化功能实体的角度对本发明实施例中的kudu表更新装置进行详细描述，下面从硬件处理的角度对本发明实施例中kudu表更新设备进行详细描述。

[0116] 图4是本发明实施例提供的一种kudu表更新设备的结构示意图，该kudu表更新设备400可因配置或性能不同而产生比较大的差异，可以包括一个或一个以上处理器(central processing units，CPU)410(例如，一个或一个以上处理器)和存储器420，一个或一个以上存储应用程序433或数据432的存储介质430(例如一个或一个以上海量存储设备)。其中，存储器420和存储介质430可以是短暂存储或持久存储。存储在存储介质430的程序可以包括一个或一个以上模块(图示没标出)，每个模块可以包括对kudu表更新设备400中的一系列指令操作。更进一步地，处理器410可以设置为与存储介质430通信，在kudu表更新设备400上执行存储介质430中的一系列指令操作。

[0117] 基于kudu表更新设备400还可以包括一个或一个以上电源440，一个或一个以上有线或无线网络接口450，一个或一个以上输入输出接口460，和/或，一个或一个以上操作系统431，例如Windows Serve，Mac OS X，Unix，Linux，FreeBSD等等。本领域技术人员可以理解，图4示出的kudu表更新设备结构并不构成对基于kudu表更新设备的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件布置。

[0118] 本发明还提供一种计算机可读存储介质，该计算机可读存储介质可以为非易失性计算机可读存储介质，该计算机可读存储介质也可以为易失性计算机可读存储介质，所述计算机可读存储介质中存储有指令，当所述指令在计算机上运行时，使得计算机执行所述kudu表更新方法的步骤。

[0119] 所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统或装置、单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

[0120] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备(可以是个人计算机，服务器，或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U盘、移动硬盘、只读存储器(read-only memory，ROM)、随机存取存储器(random access memory，RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0121] 以上所述，以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

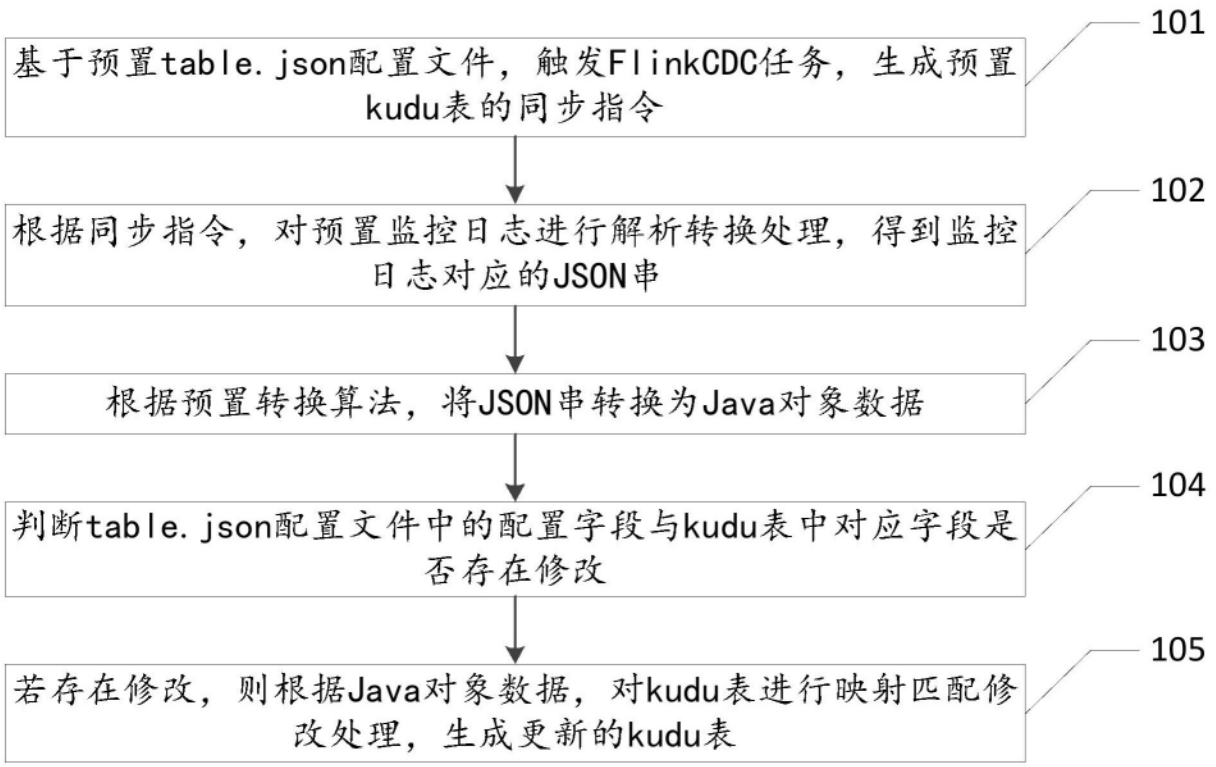
101
基于预置table.json配置文件，触发FlinkCDC任务，生成预置kudu表的同步指令
102
根据同步指令，对预置监控日志进行解析转换处理，得到监控日志对应的JSON串
103
根据预置转换算法，将JSON串转换为Java对象数据
104
判断table.json配置文件中的配置字段与kudu表中对应字段是否存在修改
105
若存在修改，则根据Java对象数据，对kudu表进行映射匹配修改处理，生成更新的kudu表

图1

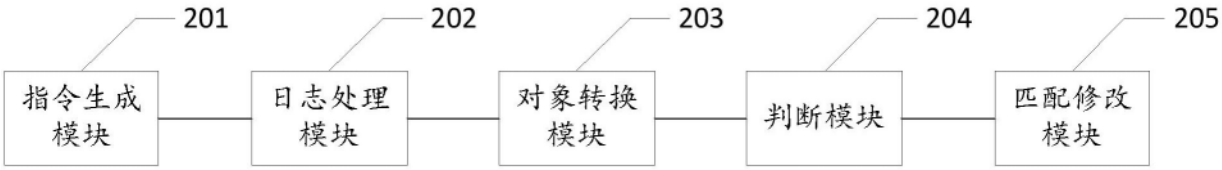
201
指令生成模块
202
日志处理模块
203
对象转换模块
204
判断模块
205
匹配修改模块

图2

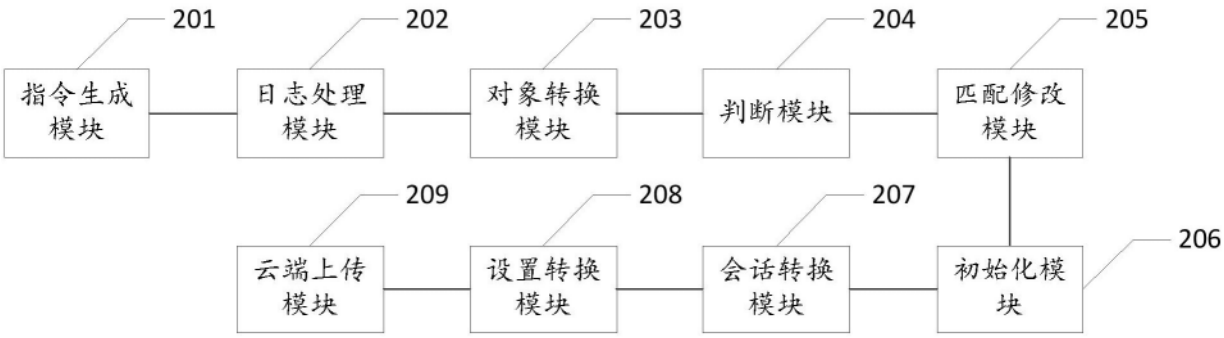
201
指令生成模块
202
日志处理模块
203
对象转换模块
204
判断模块
205
匹配修改模块
209
云端上传模块
208
设置转换模块
207
会话转换模块
206
初始化模块

图3

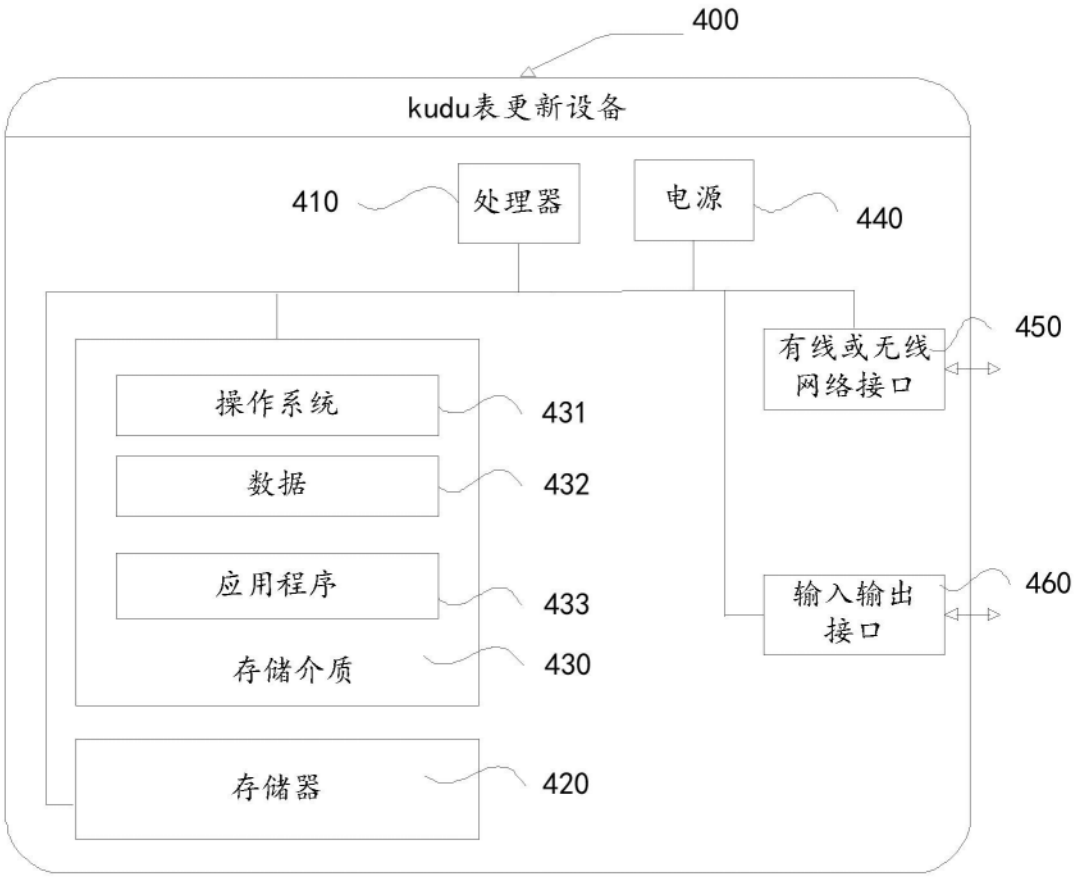
400
kudu表更新设备
410
处理器
电源
440
450
有线或无线
网络接口
操作系统
431
数据
432
应用程序
433
存储介质
430
460
输入输出
接口
存储器
420

图4