



(12) 发明专利申请

(10) 申请公布号 CN 116974983 A

(43) 申请公布日 2023. 10. 31

(21) 申请号 202210449462.6

(22) 申请日 2022.04.24

(71) 申请人 腾讯科技(深圳)有限公司  
地址 518000 广东省深圳市南山区高新区  
科技中一路腾讯大厦35层

(72) 发明人 刘国旭

(74) 专利代理机构 深圳市联鼎知识产权代理有  
限公司 44232  
专利代理师 徐明霞

(51) Int. Cl.

G06F 16/11 (2019.01)

G06F 16/176 (2019.01)

G06F 16/18 (2019.01)

G06F 9/52 (2006.01)

权利要求书2页 说明书15页 附图5页

(54) 发明名称

数据处理方法、装置、计算机可读介质及电子设备

(57) 摘要

本申请属于数据处理领域,涉及数据处理方法、装置、计算机可读介质及电子设备。方法包括:响应数据核查请求,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号,所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号;将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,并根据所述数据核查结果执行目标操作。本申请能够实现系统无锁化数据核查,减少了系统资源的占用,同时用于数据核查的数据量少,提高了数据核查的效率。

S310

响应数据核查请求,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号,所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号

S320

将所述第一主键信息和所述第一版本号分部与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,并根据所述数据核查结果执行目标操作

1. 一种数据处理方法,其特征在于,包括:

响应数据核查请求,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号,所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号;

将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,并根据所述数据核查结果执行目标操作。

2. 根据权利要求1所述的方法,其特征在于,所述获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,包括:

基于多版本并发控制机制对所述源节点对应的数据和所述目的节点对应的数据进行快照读,以获取与所述源节点对应的第一快照和与所述目的节点对应的第二快照;

从所述第一快照中抽取与各条数据对应的所述第一主键信息和所述第一版本号,从所述第二快照中抽取与各条数据对应的所述第二主键信息和所述第二版本号;

根据所述第一主键信息和所述第一版本号生成所述第一数据副本,根据所述第二主键信息和所述第二版本号生成所述第二数据副本。

3. 根据权利要求1或2所述的方法,其特征在于,所述将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,包括:

将所述第一主键信息与所述第二主键信息进行对比;

当所述第一主键信息与所述第二主键信息相同时,获取与所述第一主键信息对应的第一目标版本号与与所述第二主键信息对应的第二目标版本号;

将所述第一目标版本号与所述第二目标版本号进行对比;

当所述第一目标版本号与所述第二目标版本号不同时,确定所述数据核查结果为数据传输异常。

4. 根据权利要求1或2所述的方法,其特征在于,所述将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,包括:

将所述第一主键信息与所述第二主键信息进行对比;

当不存在与所述第一主键信息相同的第二主键信息时,确定所述数据核查结果为数据传输异常。

5. 根据权利要求1所述的方法,其特征在于,在响应数据核查请求之前,所述方法还包括:

响应所述源节点发送的查询请求,获取与所述查询请求中的主键信息对应的锁的状态;

当所述锁的状态为解除状态时,通过所述源节点将数据传输至所述目的节点。

6. 根据权利要求1所述的方法,其特征在于,所述根据所述数据核查结果执行目标操作,包括:

当所述数据核查结果为数据传输异常时,获取与异常数据对应的目标主键信息;

将与所述目标主键信息对应的锁的状态切换为上锁状态,以停止所述源节点和所述目的节点之间的数据传输;

根据与所述目标主键信息对应的源节点数据对所述目的节点进行数据修复。

7. 根据权利要求6所述的方法,其特征在于,所述根据与所述目标主键信息对应的源节点数据对所述目的节点进行数据修复,包括:

将所述目标主键信息与所述源节点对应的数据表中的主键信息进行对比,以获取与所述目标主键信息对应的目标数据;

根据所述目标数据对所述目的节点进行数据修复。

8. 根据权利要求7所述的方法,其特征在于,当所述第二数据副本中与所述目标主键信息对应的版本号不同于所述第一数据副本中与所述目标主键信息对应的版本号时;

所述根据所述目标数据对所述目的节点进行数据修复,包括:

根据所述目标主键信息从所述目的节点对应的数据表中获取待更新数据;

根据所述目标数据对所述待更新数据进行更新。

9. 根据权利要求7所述的方法,其特征在于,当所述第二数据副本中不存在与所述目标主键信息相同的第二主键信息时;

所述根据所述目标数据对所述目的节点进行数据修复,包括:

将所述目标数据插入所述目的节点对应的数据表中。

10. 根据权利要求7所述的方法,其特征在于,所述源节点和所述目的节点是包含多个节点的业务线程中的任意两相邻节点;

在根据所述目标数据对所述目的节点进行数据修复之后,所述方法还包括:

通过所述源节点将缓存的数据传送至所述目的节点,其中所述缓存的数据为与所述源节点相邻的上一节点发送至所述源节点的数据。

11. 一种数据处理装置,其特征在于,包括:

副本获取模块,用于响应数据核查请求,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号,所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号;

数据对比模块,用于将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,并根据所述数据核查结果执行目标操作。

12. 一种计算机可读介质,其特征在于,包括指令,当其在计算机上运行时,使得所述计算机执行权利要求1至10中任意一项所述的数据处理方法。

13. 一种电子设备,其特征在于,包括:

处理器;以及

存储器,用于存储所述处理器的可执行指令;

其中,所述处理器用于调用所述可执行指令来实现权利要求1至10中任意一项所述的数据处理方法。

14. 一种包含指令的计算机程序产品,其特征在于,当所述指令在计算机上运行时,使得所述计算机执行权利要求1至10中任意一项所述的数据处理方法。

## 数据处理方法、装置、计算机可读介质及电子设备

### 技术领域

[0001] 本申请属于数据处理技术领域，具体涉及一种数据处理方法、数据处理装置、计算机可读介质以及电子设备。

### 背景技术

[0002] 对于一个线上的业务系统，时刻都有大量的数据在系统各节点间传递，数据传递过程中会经过如Kafka、Redis等中间件，如果其中某一环节出现异常，会导致源端和目的端的数据不一致，必然会影响系统的运行。因此必须进行节点间的数据核查，然而系统中的数据无时无刻不在变化，在动态的情况下如何进行数据核查就成为了一个难点。

[0003] 目前，主要通过加锁的方式进行数据核查，首先向想要进行数据核查的源端发送核查请求，源端在接收到核查请求后，暂停从上游接收新的数据，并在将现有的数据传送到目的端后返回，然后从源端和目的端获取数据进行核查，最后将有差异的数据发送到目的端，完成核查。这种数据核查方法对于数据量大的系统而言，为了进行数据核查而暂停系统中的数据传输，会占用大量资源，影响正常的数据传输，进而影响用户业务的顺利进行。

### 发明内容

[0004] 本申请提供一种数据处理方法、数据处理装置、计算机可读介质以及电子设备，能够克服相关技术中存在的在系统数据动态变化的情况下进行数据核查需要加锁暂停数据传输的问题。

[0005] 本申请的其他特性和优点将通过下面的详细描述变得显然，或部分地通过本申请的实践而习得。

[0006] 第一方面，本申请实施例提供一种数据处理方法，该方法包括：响应数据核查请求，获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本，所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号，所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号；将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比，以获取数据核查结果，并根据所述数据核查结果执行目标操作。

[0007] 第二方面，本申请实施例提供一种数据处理装置，该装置包括：副本获取模块，用于响应数据核查请求，获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本，所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号，所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号；数据对比模块，用于将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比，以获取数据核查结果，并根据所述数据核查结果执行目标操作。

[0008] 第三方面，本申请实施例提供一种计算机可读介质，其上存储有计算机程序，该计算机程序被处理器执行时实现如以上技术方案中的数据处理方法。

[0009] 第四方面，本申请实施例提供一种电子设备，该电子设备包括：处理器；以及存储

器,用于存储所述处理器的可执行指令;其中,所述处理器被配置为经由执行所述可执行指令来执行如以上技术方案中的数据处理方法。

[0010] 第五方面,本申请实施例提供一种包含指令的计算机程序产品,当该指令在计算机上运行时,使得该计算机执行上述数据处理方法。

[0011] 本申请实施例提供的数据处理方法,一方面可以在响应数据核查请求后,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,通过将第一数据副本中的第一主键信息好第一版本号与第二数据副本中的第二主键信息和第二版本号进行对比,以判断源节点发出的数据和目的节点接收到的数据是否相同,由于采用的是数据副本,因此不需要停止数据传输,也无需对传输数据的线路进行加锁,实现了系统无锁化数据核查,减少了系统资源的占用;另一方面,由于本申请中仅根据每行数据中的主键信息和版本号形成第一数据副本和第二数据副本,因此减少了数据核查时所采用的数据量,提高了数据核查的效率。

[0012] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本申请。

## 附图说明

[0013] 图1示意性地示出了一个实施例中应用本申请技术方案的系统的架构框图。

[0014] 图2示意性地示出了相关技术中采用加锁的方式进行数据核查的流程示意图。

[0015] 图3示意性地示出了一个实施例中数据处理方法的步骤流程示意图。

[0016] 图4示意性地示出了一个实施例中获取第一数据副本和第二数据副本的流程示意图。

[0017] 图5示意性地示出了一个实施例中获取数据核查结果的流程示意图。

[0018] 图6示意性地示出了一个实施例中根据数据核查结果执行目标操作的流程示意图。

[0019] 图7示意性地示出了一个实施例中基于Redis的分布式锁的界面示意图。

[0020] 图8示意性地示出了一个实施例中进行数据核查和数据修复的流程示意图。

[0021] 图9示意性地示出了一个实施例中数据处理装置的结构框图。

[0022] 图10示意性示出了一个实施例中适于用来实现本申请实施例的电子设备的计算机系统结构框图。

## 具体实施方式

[0023] 现在将参考附图更全面地描述示例实施方式。然而,示例实施方式能够以多种形式实施,且不应被理解为限于在此阐述的范例;相反,提供这些实施方式使得本申请将更加全面和完整,并将示例实施方式的构思全面地传达给本领域的技术人员。

[0024] 此外,所描述的特征、结构或特性可以以任何合适的方式结合在一个或更多实施例中。在下面的描述中,提供许多具体细节从而给出对本申请的实施例的充分理解。然而,本领域技术人员将意识到,可以实践本申请的技术方案而没有特定细节中的一个或更多,或者可以采用其它的方法、组元、装置、步骤等。在其它情况下,不详细示出或描述公知方法、装置、实现或者操作以避免模糊本申请的各方面。

[0025] 附图中所示的方框图仅仅是功能实体,不一定必须与物理上独立的实体相对应。即,可以采用软件形式来实现这些功能实体,或在一个或多个硬件模块或集成电路中实现这些功能实体,或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

[0026] 附图中所示的流程图仅是示例性说明,不是必须包括所有的内容和操作 /步骤,也不是必须按所描述的顺序执行。例如,有的操作/步骤还可以分解,而有的操作/步骤可以合并或部分合并,因此实际执行的顺序有可能根据实际情况改变。

[0027] 为了便于理解本申请的技术方案,在此对本申请中涉及的技术名词进行解释说明。

[0028] 1、多版本并发控制(Multiversion concurrency control,简称MCC或MVCC)是数据库管理系统常用的一种并发控制,也用于程序设计语言实现事务内存。MVCC意图解决读写锁造成的多个、长时间的读操作和死写操作问题,每个事务读到的数据项都是一个历史快照,并依赖于实现的隔离级别。写操作不覆盖已有数据项,而是创建一个新的版本,直至所有操作提交时才变为课件。快照隔离使得事务看到它启动时的数据状态。

[0029] MVCC主要解决了以下问题:

[0030] 1. 让读写互相不阻塞,即读不阻塞写,写不阻塞读,这样就可以提升事务并发处理能力。提高并发的演进思路有:普通锁,只能串行执行;读写锁,可以实现读写并发;数据多版本并发控制,可以实现读写并发。

[0031] 2. 降低了死锁的概率,MVCC采用了乐观锁的方式,读取数据时并不需要加锁,对于写操作,也只对必要的数据行加锁。

[0032] 3. 解决了一致性读的问题,一致性读也被称为快照读,当查询数据库在某个时间点的快照时,只能看到该时间点之前事务提交更新的结果,而不能看到该时间点之后事务提交更新的结果。

[0033] 2、数据副本:对系统中数据流转的各节点进行一致性读生成快照,并从快照中提取主键信息和版本号所生成的数据。

[0034] 3、主键信息:可以将各数据行区分开的信息,例如可以是节点编号、节点名称等。

[0035] 4、版本号:用于指示数据行中数据的更新情况,该版本号是一个全局自增的版本号,数据行中任意一个或多个数据发生变化,版本号都会自增。

[0036] 5、源节点、目的节点:根据数据流向确定,相邻两节点中,发送数据的节点为源节点,接收数据的节点为目的节点。

[0037] 图1示意性地示出了应用本申请技术方案的示例性系统架构框图。

[0038] 如图1所示,系统架构100可以包括终端设备101、网络102和服务器 103。终端设备101可以包括智能手机、平板电脑、笔记本电脑、台式电脑、智能电视、智能车载终端等各种电子设备。服务器103可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,还可以是提供云计算服务的云服务器。网络102是能够在终端设备101和服务器 103之间提供通信链路的各种连接类型的通信介质,例如可以是有线通信链路或者无线通信链路。

[0039] 根据实现需要,本申请实施例中的系统架构可以具有任意数目的终端设备、网络和服务器。例如,服务器可以由多个服务器设备组成的服务器群组。另外,本申请实施例提供的技术方案可以应用于服务器103,也可以应用于终端设备101,当然也可以由终端设

备101和服务器103共同实施,本申请对此不做特殊限定。

[0040] 在本申请的一个实施例中,用户通过终端设备101进行业务操作,通过网络102将业务请求发送至服务器103,以便服务器103对该业务请求进行处理,并返回业务处理结果,该业务具体可以是在线购物、参加商品秒杀活动、在线预约挂号等等。在进行业务处理的过程中,业务系统中包含多个节点,业务数据会在该业务系统中的各个节点之间进行流转,为了保证用户提交的业务请求被正确地处理,就需要保证数据在传输过程中不会发生缺少、遗失、传错等异常情况,相应地,运维服务人员就需要适时地或者在发现系统异常时对数据进行核查,或者设定系统中的数据核查服务间隔预设时间对系统中的数据进行核查。在进行数据核查时,可以对业务系统所包含的各业务线路中任意相邻两节点之间的数据进行核查,其中发送数据的节点为源节点,接收数据的节点为目的节点,然后采用多版本并发控制机制对源节点和目的节点的数据进行一致性读,以获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,接着将第一数据副本和第二数据副本中的数据进行对比,以获取数据核查结果,并根据数据核查结果进行目标操作。该数据核查结果分为两类,数据传输异常和数据传输正常,如果数据传输正常,则无需进行任何处理,等待进行下次数据核查,如果数据传输异常,则需要根据源节点中正确的数据对目的节点中的数据进行修复。

[0041] 在本申请的一个实施例中,用于进行数据核查的数据核查服务可以搭建于服务器103中,也可以搭建与终端设备101中,在满足数据核查的触发条件时,便可调用该数据核查服务对待核查的源目的节点的数据进行核查。在本申请的实施例中,数据核查服务是外置服务,通过单独设置数据核查服务可以避免对系统中其它服务以及数据传输造成影响,并且减少了修改系统代码的工作量。

[0042] 进一步地,在形成第一数据副本和第二数据副本时,可以根据各行数据中的主键信息和版本号生成,其中主键信息是可以将每行数据区分开的数据,版本号则用于指示数据行中的数据发生过多少次的变更,正常情况下,源节点中某一主键信息对应的版本号与目的节点中该主键信息对应的版本号是相同的,如果存在不同或缺失,则说明在数据传输过程中出现了异常。

[0043] 在本申请的一个实施例中,可以采用提供云计算服务的云服务器执行本申请中的数据处理方法,相应地,本申请的技术方案涉及云技术中的云计算和云存储。

[0044] 云计算(cloud computing)是一种计算模式,它将计算任务分布在大量计算机构成的资源池上,使各种应用系统能够根据需要获取计算力、存储空间和信息服务。提供资源的网络被称为“云”。“云”中的资源在使用者看来是可以无限扩展的,并且可以随时获取,按需使用,随时扩展,按使用付费。

[0045] 作为云计算的基础能力提供商,会建立云计算资源池(简称云平台,一般称为IaaS (Infrastructure as a Service,基础设施即服务)平台,在资源池中部署多种类型的虚拟资源,供外部客户选择使用。云计算资源池中主要包括:计算设备(为虚拟化机器,包含操作系统)、存储设备、网络设备。

[0046] 按照逻辑功能划分,在IaaS (Infrastructure as a Service,基础设施即服务)层上可以部署PaaS (Platform as a Service,平台即服务)层,PaaS层之上再部署SaaS (Software as a Service,软件即服务)层,也可以直接将SaaS部署在 IaaS上。PaaS为软件运行的平台,如数据库、web容器等。SaaS为各式各样的业务软件,如web门户网站、短信群发

器等。一般来说,SaaS和PaaS 相对于IaaS是上层。

[0047] 云存储(cloud storage)是在云计算概念上延伸和发展出来的一个新的概念,分布式云存储系统(以下简称存储系统)是指通过集群应用、网格技术以及分布存储文件系统等功能,将网络中大量各种不同类型的存储设备(存储设备也称之为存储节点)通过应用软件或应用接口集合起来协同工作,共同对外提供数据存储和业务访问功能的一个存储系统。

[0048] 目前,存储系统的存储方法为:创建逻辑卷,在创建逻辑卷时,就为每个逻辑卷分配物理存储空间,该物理存储空间可能是某个存储设备或者某几个存储设备的磁盘组成。客户端在某一逻辑卷上存储数据,也就是将数据存储在文件系统上,文件系统将数据分成许多部分,每一部分是一个对象,对象不仅包含数据而且还包含数据标识(ID, ID entity)等额外的信息,文件系统将每个对象分别写入该逻辑卷的物理存储空间,且文件系统会记录每个对象的存储位置信息,从而当客户端请求访问数据时,文件系统能够根据每个对象的存储位置信息让客户端对数据进行访问。

[0049] 存储系统为逻辑卷分配物理存储空间的过程,具体为:按照对存储于逻辑卷的对象的容量估量(该估量往往相对于实际要存储的对象的容量有很大余量)和独立冗余磁盘阵列(RAID,Redundant Array of Independent Disk)的组别,预先将物理存储空间划分成分条,一个逻辑卷可以理解为一个分条,从而为逻辑卷分配了物理存储空间。

[0050] 图2示出了相关技术中采用加锁的方式进行数据核查的流程示意图,如图2所示,业务数据的流向是Node(节点)1→Node 2→Node 3,当运维服务人员想对Node 2和Node 3之间的数据进行核查时,先向Node 2发送数据核查请求;Node 2接收到数据核查请求后,暂停从Node 1接收新数据,并等待现有数据完成到Node 3的传输之后返回;运维服务人员分别从Node 2和Node 3处获取数据,并开始进行核查,最后将有差异的数据发送到 Node 3,完成核查。

[0051] 虽然图2所示的方案对数据量小的系统影响不大,但是对于数据量大的系统而言,通过加锁暂停两节点之间的数据传输,以进行数据核查,会占用大量资源,影响正常的数据传输,进而对用户业务造成严重影响。

[0052] 针对相关技术中存在的问题,下面结合具体实施方式对本申请提供的数据处理方法、数据处理装置、计算机可读介质以及电子设备等技术方案做出详细说明。

[0053] 图3示意性地示出了本申请一个实施例中的数据处理方法的步骤流程示意图,该数据处理方法可以由配置于服务器中的数据核查服务执行,该服务器具体可以是图1中的服务器103,当然也可以由配置于终端设备101中的数据核查服务执行。如图3所示,本申请实施例中的数据处理方法主要可以包括如下的S310至S320。

[0054] S310:响应数据核查请求,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号,所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号;

[0055] S320:将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,并根据所述数据核查结果执行目标操作。

[0056] 本申请实施例提供的数据处理方法,一方面可以在响应数据核查请求后,获取与



源节点对应的第一数据副本和与目的节点对应的第二数据副本,通过将第一数据副本中的第一主键信息和第一版本号分别与第二数据副本中的第二主键信息和第二版本号进行对比,以判断源节点发出的数据和目的节点接收到的数据是否相同,由于采用的是数据副本,因此不需要停止数据传输,也无需对传输的数据进行加锁,实现了系统无锁化数据核查,减少了系统资源的占用;另一方面,由于本申请中仅根据每行数据中的主键信息和版本号形成第一数据副本和第二数据副本,因此减少了数据核查时所采用的数据量,提高了数据核查的效率。

[0057] 下面对数据处理方法的各个方法步骤的具体实现方式进行详细说明。

[0058] 在S310中,响应数据核查请求,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号,所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号。

[0059] 在本申请的一个实施例中,触发数据核查任务的条件主要分为两种情况,一种是运维服务人员在系统运行过程中发现异常时,触发数据核查任务,一种是在预定时间,系统主动触发数据核查任务,例如在业务低峰时段主动触发数据核查任务。无论是运维服务人员触发还是系统主动触发,在响应该数据核查请求后,数据核查服务即可根据响应数据核查请求的时间点从源节点处维护的数据表和目的节点处维护的数据表中获取与待核查的源节点和目的节点对应的数据,以便进行数据核查。其中,待核查的源节点和目的节点可以是运维服务人员或系统指定的节点,也可以是系统中各业务线程中的部分或全部节点。

[0060] 在本申请的一个实施例中,系统具体可以是一个分布式系统,也可以是一个非分布式系统。在进行数据核查时,可以同时多个由源节点和目的节点组成的节点组进行数据核查。在根据获取的与待核查的源节点和目的节点对应的数据进行数据核查时,可以通过与源节点对应的第一数据副本和与目的节点对应的第二数据副本进行数据核查,其中,第一数据副本和第二数据副本包括与数据表中各条数据对应的主键信息和版本号,这样就可以避免为了对源节点和目的节点之间的数据进行核查而对源节点和目的节点之间的业务线程加锁,进而避免了对外部正常的数据传输造成影响,影响用户业务。

[0061] 图4示出了获取第一数据副本和第二数据副本的流程示意图,如图4所示,该流程至少包括S401-S403:

[0062] 在S401中,基于多版本并发控制机制对所述源节点对应的数据和所述目的节点对应的数据进行快照读,以获取与所述源节点对应的第一快照和与所述目的节点对应的第二快照。

[0063] 在本申请的一个实施例中,主要采用了MVCC中的一致性读机制,即快照读,通过对源节点对应的数据和目的节点对应的数据进行快照读,可以获取与源节点对应的第一快照和与目的节点对应的第二快照。根据一致性读(快照读)的机理可知,所获取的第一快照中所包含的数据是根据响应数据核查请求的时间点在与源节点对应的数据表中所能读取到的事务提交更新的数据,同样地,第二快照中所包含的数据也是根据响应数据核查请求的时间点在与目的节点对应的数据表中所能读取到的事务提交更新的数据。

[0064] 在S402中,从所述第一快照中抽取与各条数据对应的第一主键信息和第一版本号,从所述第二快照中抽取与各条数据对应的第二主键信息和第二版本号。

[0065] 在本申请的一个实施例中,由于各节点对应的数据表中存在多条数据,且各条数据包含对应多个维度的数据,为了减少数据核查时所使用的数据量,且不影响数据核查的结果,在获取第一快照和第二快照后,可以从中抽取可用于指示数据更新信息的关键维度的数据即可。在本申请的实施例中,关键维度的数据具体包括各条数据对应的主键信息和版本号,其中主键信息是用于区分各条数据的信息,例如可以是节点编号、节点名称等等,版本号是用于指示各数据行中数据的更新情况,其是一个全局自增的版本号,在数据行中的数据发生变化时,该数据行所对应的版本号可以以1为单位自增,也可以以其它固定的数值进行自增,本申请实施例对此不做具体限定。由于当某一行的数据发生变化时,都会更新该行对应的版本号,因此根据版本号的数值就能知道系统数据进行变更的位置。举例而言,当存在一条数据,其中的主键信息为A,版本号为3,那么就说明A对应的数据行中的数据在响应数据核查请求的时间点之前发生了3次变更。

[0066] 在本申请的一个实施例中,在获取第一快照和第二快照后,可以从第一快照中抽取其中各条数据所对应的第一主键信息和第一版本号,并且可以从第二快照中抽取其中各条数据所对应的第二主键信息和第二版本号,进而可以根据第一主键信息和第一版本号可以生成第一数据副本,根据第二主键信息和第二版本号可以生成第二数据副本。

[0067] 以网络控制器之间的路由数据为例,每个网络控制器控制多个虚拟路由转发(Virtual Routing Forwarding,简称VRF),将任意相邻的两个网络控制器分别作为源节点和目的节点,根据上述实施例中的方法可以获取与源节点和目的节点对应的快照以及数据副本,表1和表2示出了与源节点对应的第一快照和与目的节点对应的第二快照:

[0068] 表1与源节点对应的第一快照

VRF	Cidr	A	B	版本号
VRF1	10.206.0.1	Np1	16	1
	10.206.0.2	Np2	16	100
VRF2	172.21.0.1	Np3	16	80
	172.21.0.3	Np4	16	101

[0070] 表2与目的节点对应的第二快照

VRF	Cidr	A	B	版本号
VRF1	10.206.0.1	Np1	16	1
	10.206.0.2	Np2	16	100
VRF2	172.21.0.1	Np3	16	80

[0072] 分析表1和表2可知,所获取的第一快照和第二快照中包含的各数据行中包括五个维度的数据,分别为:VRF、Cidr(Classless Inter-Domain Routing,无类域间路由选择)、A、B和版本号,其中A和B可以是诸如下一跳IP地址(Nexthop)和前缀(Prefix)等与VRF相关的信息。由于根据VRF和Cidr可以唯一确定一个数据行,因此可以从第一快照和第二快照中抽取VRF、Cidr和版本号形成第一数据副本和第二数据副本,表3、表4 示出了第一数据副本和第二数据副本:

[0073] 表3与源节点对应的第一数据副本

VRF	Cidr	版本号
VRF1	10.206.0.1	1
	10.206.0.2	100
VRF2	172.21.0.1	80
	172.21.0.3	101

[0074]

[0075] 表4与目的节点对应的第二数据副本

VRF	Cidr	版本号
VRF1	10.206.0.1	1
	10.206.0.2	100
VRF2	172.21.0.1	80

[0076]

[0077] 在S403中,根据所述第一主键信息和所述第一版本号生成所述第一数据副本,根据所述第二主键信息和所述第二版本号生成所述第二数据副本。

[0078] 在本申请的一个实施例中,在获取第一主键信息、第一版本号、第二主键信息和第二版本号后,可以根据第一主键信息和第一版本号生成第一数据副本,根据第二主键信息和第二版本号生成第二数据副本。然后将第一数据副本和第二数据副本中的数据进行对比,以确定数据在由源节点流向目的节点的过程中是否存在异常。

[0079] 在本申请的一个实施例中,由于第一版本号和第二版本号是全局自增的,那么在数据传输过程不出现异常的情况下,源节点和目的节点的数据中应当存在相同的主键信息,并且同一主键信息对应的版本号是相同的,如果目的节点中的主键信息与源节点中的主键信息不同,或者源节点和目的节点中对应同一主键信息的版本号不同,则说明数据传输过程存在异常,因此在本申请的实施例中,可以通过对比源节点和目的节点中的主键信息是否相同,以及源节点和目的节点中对应同一主键信息的版本号,判断数据在由源节点流向目的节点时有没有出现异常。

[0080] 在本申请的一个实施例中,在根据数据行中数据的变更情况更新版本号的同时,还可以根据更新的数据生成日志,该日志记录了更新前的数据和更新后的数据。那么对于源节点和目的节点而言,都存在与之对应的日志,根据源节点对应的日志和目的节点对应的日志可以判断二者每次的数据变化是否一致,如果不一致,可以根据源节点对应的日志中的数据进行回滚,以对目的节点对应的数据行进行数据修复,使得在源节点和目的节点中对应同一主键信息的版本号相同时,所有小于该版本号的数据行的数据也都在源节点和目的节点保持一致。

[0081] 在S320中,将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,并根据所述数据核查结果执行目标操作。

[0082] 在本申请的一个实施例中,在获取第一数据副本和第二数据副本后,可以将第一数据副本中的第一主键信息和第一版本号与第二数据副本中的第二主键信息和第二版本

号进行对比,以获取数据核查结果。在本申请的实施例中,数据核查结果有两类,第一类是数据传输正常,第二类是数据传输异常,当数据传输正常时,无需对系统数据进行任何处理,当数据传输异常时,则需要根据正确的数据对错误的数据进行修复。进一步地,数据传输异常的情况也分为两类,第一类是数据丢失,如表3和表4所示的第一数据副本和第二数据副本所示,第二数据副本相对于第一数据副本缺失了对应(VRF2, 172.21.0.3)的数据行,还有一类数据传输异常是数据错误,例如当表4所示的第二数据副本中存在对应(VRF2, 172.21.0.3)的数据行,但是该行数据对应的版本号不同于第一数据副本中(VRF2, 172.21.0.3)对应的版本号时,即为数据传输时出现错误,因此在对数据传输异常的情况进行判断时,需要通过两种方式实现。

[0083] 图5示出了一种获取数据核查结果的流程示意图,如图5所示,在 S501中,将所述第一主键信息与所述第二主键信息进行对比;在S502中,当所述第一主键信息与所述第二主键信息相同时,获取与所述第一主键信息对应的第一目标版本号和与所述第二主键信息对应的第二目标版本号;在 S503中,将所述第一目标版本号与所述第二目标版本号进行对比;在S504 中,当所述第一目标版本号与所述第二目标版本号不同时,确定所述数据核查结果为数据传输异常。

[0084] 其中,第一主键信息和第二主键信息的数量为一个或多个,当第一主键信息和第二主键信息均为多个时,对第一主键信息进行轮询,将任意一个第一主键信息作为第一目标主键信息,然后将第一目标主键信息与各个第二主键信息进行对比,当存在与第一目标主键信息相同的第二目标主键信息时,获取与第一主键信息对应的第一目标版本号和与第二目标主键信息对应的第二目标版本号,然后将第一目标版本号和第二目标版本号进行对比,并根据对比结果确定数据核查结果。同样地,根据上述方法也可以在第一主键信息和/或第二主键信息的数量为一个时,获取数据核查结果。

[0085] 图5所示的方法可以判断得到数据传输错误所导致的数据传输异常。同时,将第一主键信息和第二主键信息进行对比,当不存在与第一主键信息相同的第二主键信息时,说明数据在从源节点流向目的节点时出现了数据丢失,也可以确定数据核查结果为数据传输异常。以表3和表4所示的第一数据副本和第二数据副本为例,将第一数据副本中的主键信息(VRF1,10.206.0.1)、(VRF1,10.206.0.2)、(VRF2,172.21.0.1)、(VRF2,172.21.0.3)分别与第二数据副本中的主键信息(VRF1,10.206.0.1)、(VRF1,10.206.0.2)、(VRF2,172.21.0.1)进行对比,对比发现,第一数据副本中的主键信息(VRF1,10.206.0.1)、(VRF1,10.206.0.2)、(VRF2,172.21.0.1)与第二数据副本中的主键信息(VRF1,10.206.0.1)、(VRF1,10.206.0.2)、(VRF2,172.21.0.1)都相同,接着可以获取这些主键信息对应的版本号进行对比,通过对比可知,各主键信息对应的版本号也都相同,说明这些数据在传输过程中未发生异常,但是第二数据副本中不存在主键信息(VRF2,172.21.0.3),说明在数据传输的过程中,该条数据丢失掉了,需要增加该条数据至目的节点的数据表中。

[0086] 在本申请的一个实施例中,无论是数据丢失还是数据错误,都是数据传输异常,都需要对目的节点对应的数据进行修复,以保证目的节点接收到的数据与源节点发送的数据相同,进而避免对用户业务的处理结果造成影响。

[0087] 在本申请的一个实施例中,在进行数据修复时,需要对待修复的数据行所对应的业务线程上锁,以暂停源节点和目的节点之间的数据传输,然后根据正确的源节点数据对

目的节点的数据表进行修复。图6示出了根据数据核查结果执行目标操作的流程示意图,如图6所示,在S601中,当所述数据核查结果为数据传输异常时,获取与异常数据对应的目标主键信息;在S602中,将与所述目标主键信息对应的锁的状态切换为上锁状态,以停止所述源节点和所述目的节点之间的数据传输;在S603中,根据与所述目标主键信息对应的源节点数据对所述目的节点进行数据修复。

[0088] 在根据与目标主键信息对应的源节点数据对目的节点进行数据修复时,首先可以将目标主键信息与源节点对应的数据表中的主键信息进行对比,以获取与该目标主键信息对应的目标数据;然后根据该目标数据对目的节点进行数据修复。

[0089] 由于数据传输异常包括数据丢失和数据错误,那么在进行数据修复时,所采用的方式也不同。具体地,当第二数据副本中存在与目标主键信息对应的版本号,且第二数据副本中与目标主键信息对应的版本号不同于第一数据副本中与目标主键信息对应的版本号时,也就是说,数据传输异常的类型是数据错误时,可以根据目标主键信息从目的节点对应的数据表中获取待更新数据,该待更新数据即为与目标主键信息对应的数据;然后根据所获取的目标数据对该待更新数据进行更新。当第二数据副本中不存在与目标主键信息对应的第二主键信息时,也就是说,数据传输异常的类型为数据丢失时,可以将目标数据插入目的节点对应的数据表中,以补全目的节点的数据表。

[0090] 在本申请的一个实施例中,所采用的锁为分布式锁,在确定目标主键信息后,可以通过对该目标主键信息对应的锁赋值以将该锁的状态切换为上锁状态,方便对目的节点进行数据修复。分布式锁是控制分布式系统之间同步访问共享资源的一种方式,如果不同的系统或是同一个系统的不同主机之间共享了一个或一组资源,那么访问这些资源的时候,往往需要互斥来防止彼此干扰来保证一致性,这便需要使用到分布式锁。

[0091] 在本申请的一个实施例中,该分布式锁可以是基于Redis的分布式锁,也可以是基于zookeeper的分布式锁。

[0092] 当为基于Redis的分布式锁时,可以通过数据核查服务借助Redis记录锁的key,各个源节点通过对redis进行周期轮询以获取该key,并根据key 对对应的业务线程上锁,进而对该业务线程中的数据进行修复。接下来基于表3和表4中所示的第一数据副本和第二数据副本,对基于Redis的分布式锁进行说明。

[0093] 图7示出了一种基于Redis的分布式锁的界面示意图,如图7所示,数据核查服务701借助Redis 702记录了一个key:lock-VRF2,有两个源节点703和704向同一目的节点705发送数据,在判定第二数据副本中缺少与主键信息(VRF2,172.21.0.3)对应的信息,需要进行数据修复时,数据核查服务701会对lock-VRF2设置一个值,以暂停VRF2业务线程的数据传输,并将源节点中与(VRF2,172.21.0.3)对应的数据插入至目的节点705的数据表中。

[0094] 在本申请的一个实施例汇总,如果源节点703和704与目的节点705之间的数据传输都包含VRF2业务线程的数据传输时,那么源节点703和704 需要周期轮询Redis中是否存在该key,如果不存在,则第一个查询的源节点,例如源节点703,获取该锁,并通过上锁将其与目的节点705之间对应 VRF2业务线程的数据传输暂停,而源节点704则继续周期轮询直至该锁被源节点703解除掉,然后源节点704拿到锁并上锁,以对目的节点705进行数据修复。如果只有源节点703/704和目的节点705之间包含VRF2业务线程的数据传输时,那么只需源节点703/704上锁,暂停源节点703/704和目的节点705之间对应VRF2业务线程的数据

传输。

[0095] 进一步地,待数据修复完成后,删除该分布式锁,恢复数据传输,并将源节点703、704缓存的来自上一节点的数据发送至目的节点705。

[0096] 在本申请的一个实施例中,当Redis中存在key时,说明该key对应的锁的状态为已上锁,无法进行数据传输,只有当Redis中不存在key时,说明锁已解除,能够进行正常的的数据传输,因此源节点在向目的节点发送数据之前,都需要发送一查询请求,通过根据该查询请求中的主键信息获取 redis中与该主键信息对应的锁的状态,并根据锁的状态确定是否可以向目的节点发送数据。

[0097] 在本申请的一个实施例中,根据业务类型的不同,可以设置多个对应不同业务的锁,例如存在VRF1、VRF2和VRF3时,那么可以在Redis中记录三个key:lock-VRF1、lock-VRF2和lock-VRF3,当需要对处理某一类型的业务的节点进行数据修复时,只需对该业务类型对应的key设置一个值,即可暂停对应的数据传输,并进行数据修复。

[0098] 当分布式锁为基于zookeeper的分布式锁时,当多个源节点想要暂停其与目的节点之间的数据传输进行数据修复时,可以在zookeeper的指定节点下创建临时顺序节点,对应第一位临时顺序节点的源节点首先获取分布式锁,其它未获取分布式锁的源节点从zookeeper中获取分布式锁删除事件,当存在锁删除事件时,判断其在临时顺序节点中是否为第一位,如果是,则获取分布式锁,如果不是则继续获取分布式锁删除事件。

[0099] 通过采用分布式锁将源节点和目的节点之间需要进行数据修复的业务线程锁住,待目的节点完成数据修复后再解锁进行新的数据传输,既保证了其它业务线程的正常数据传输,又对存在异常的数据传输线程进行了修复,保证了业务数据在源节点和目的节点的一致性,进而避免了对用户业务的影响。

[0100] 为了更直观、更全面的了解本申请中的进行数据核查,以及数据核查异常时对数据进行安全修复的流程,基于表1和表2示出的第一快照和第二快照进行具体说明。图8示出了进行数据核查和数据修复的流程示意图,如图8所示,通过从第一快照801中提取第一主键信息和第一版本号可以生成第一数据副本802,从第二快照803中提取第二主键信息和第二版本号可以生成第二数据副本804,通过对比第一数据副本802和第二数据副本804,可以确定差异数据805,即为对应主键信息(VRF2,172.21.0.3)的数据行;接着可以根据差异数据805从第一快照中获取对应的数据行,同时通过与VRF2对应分布式锁将与VRF2对应的业务线程上锁,暂停源节点和目的节点之间的数据传输,以根据与差异数据805对应的数据行对目的节点的数据表806进行修复,具体地,将该数据行插入至目的节点的数据表中。

[0101] 在本申请的一个实施例中,在采用分布式锁将业务线程锁住之后,且在对目的节点进行数据修复之前,还可以对第一数据副本和第二数据副本中的数据进行再对比,以根据再对比获得的数据核查结果对前次获取的数据核查结果进行确认,当确认数据核查结果一致,均为数据传输异常时,则对目的节点进行数据修复。这样可以避免由于数据量大导致数据核查结果不准确,进而导致占用大量系统资源进行无谓的数据修复。

[0102] 本申请中的数据处理方法,通过在响应数据核查请求后获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,该第一数据副本和该第二数据副本包含与各条数据对应的主键信息和版本号;接着通过将第一数据副本中的第一主键信息和第一版本号分别与第二数据副本中的第二主键信息和第二版本号进行对比,以获取数据核查结果,

并根据数据核查结果执行目标操作。本申请中的数据处理方法,一方面由于采用的是与源节点和目的节点对应的数据副本进行对比,以判断源节点发出的数据和目的节点接收到的数据是否相同,因此不需要停止外部的数据传输,也无需对传输数据得线路加锁,实现了系统无锁化数据核查,减少了系统资源的占用;另一方面,由于本申请中仅根据每行数据中的主键信息和版本号形成第一数据副本和第二数据副本,因此减少了数据核查时所采用的数据量,提高了数据核查的效率。

[0103] 应当注意,尽管在附图中以特定顺序描述了本申请中方法的各个步骤,但是,这并非要求或者暗示必须按照该特定顺序来执行这些步骤,或是必须执行全部所示的步骤才能实现期望的结果。附加的或备选的,可以省略某些步骤,将多个步骤合并为一个步骤执行,以及/或者将一个步骤分解为多个步骤执行等。

[0104] 以下介绍本申请的装置实施例,可以用于执行本申请上述实施例中的数据处理方法。图9示意性地示出了本申请实施例提供的数据处理装置的结构框图。如图9所示,数据处理装置900包括:副本获取模块910和数据对比模块920,具体地:

[0105] 副本获取模块910,用于响应数据核查请求,获取与源节点对应的第一数据副本和与目的节点对应的第二数据副本,所述第一数据副本包含与所述源节点中各条数据对应的第一主键信息和第一版本号,所述第二数据副本包含与所述目的节点中各条数据对应的第二主键信息和第二版本号;数据对比模块920,用于将所述第一主键信息和所述第一版本号分别与所述第二主键信息和所述第二版本号进行对比,以获取数据核查结果,并根据所述数据核查结果执行目标操作。

[0106] 在本申请的一些实施例中,基于以上技术方案,所述副本获取模块910 包括:快照生成单元,用于基于多版本并发控制机制对所述源节点对应的数据和所述目的节点对应的数据进行快照读,以获取与所述源节点对应的第一快照和与所述目的节点对应的第二快照;信息抽取单元,用于从所述第一快照中抽取与各条数据对应的所述第一主键信息和所述第一版本号,从所述第二快照中抽取与各条数据对应的所述第二主键信息和所述第二版本号;副本生成单元,用于根据所述第一主键信息和所述第一版本号生成所述第一数据副本,根据所述第二主键信息和所述第二版本号生成所述第二数据副本。

[0107] 在本申请的一些实施例中,基于以上技术方案,所述数据对比模块920 配置为:将所述第一主键信息与所述第二主键信息进行对比;当所述第一主键信息与所述第二主键信息相同时,获取与所述第一主键信息对应的第一目标版本号和与所述第二主键信息对应的第二目标版本号;将所述第一目标版本号与所述第二目标版本号进行对比;当所述第一目标版本号与所述第二目标版本号不同时,确定所述数据核查结果为数据传输异常。

[0108] 在本申请的一些实施例中,基于以上技术方案,所述数据对比模块920 配置为:将所述第一主键信息与所述第二主键信息进行对比,当不存在与所述第一主键信息相同的第二主键信息时,确定所述数据核查结果为数据传输异常。

[0109] 在本申请的一些实施例中,基于以上技术方案,所述数据处理装置900 配置为:在响应数据核查请求之前,响应所述源节点发送的查询请求,获取与所述查询请求中的主键信息对应的锁的状态;当所述锁的状态为解除状态时,通过所述源节点将数据传输至所述目的节点。

[0110] 在本申请的一些实施例中,基于以上技术方案,所述数据对比模块920 包括:获取

单元,用于当所述数据核查结果为数据传输异常时,获取与异常数据对应的目标主键信息;上锁单元,用于将与所述目标主键信息对应的锁的状态切换为上锁状态,以停止所述源节点和所述目的节点之间的数据传输;第一修复单元,用于根据与所述目标主键信息对应的源节点数据对所述目的节点进行数据修复。

[0111] 在本申请的一些实施例中,基于以上技术方案,所述修复单元包括:对比单元,用于将所述目标主键信息与所述源节点对应的数据表中的主键信息进行对比,以获取与所述目标主键信息对应的目标数据;第二修复单元,用于根据所述目标数据对所述目的节点进行数据修复。

[0112] 在本申请的一些实施例中,基于以上技术方案,当所述第二数据副本中与所述目标主键信息对应的版本号不同于所述第一数据副本中与所述目标主键信息对应的版本号时;所述第二修复单元配置为:根据所述目标主键信息从所述目的节点对应的数据表中获取待更新数据;根据所述目标数据对所述待更新数据进行更新。

[0113] 在本申请的一些实施例中,基于以上技术方案,当所述第二数据副本中不存在与所述目标主键信息相同的第二主键信息时;所述第二修复单元配置为:将所述目标数据插入所述目的节点对应的数据表中。

[0114] 在本申请的一些实施例中,基于以上技术方案,所述数据对比模块920 配置为:在根据与所述目标主键信息对应的源节点数据对所述目的节点进行数据修复之前,对所述第一数据副本与所述第二数据副本中的数据进行再对比,以对所述数据核查结果进行确认。

[0115] 在本申请的一些实施例中,所述源节点和所述目的节点是包含多个节点的业务线路中的任意两相邻节点;基于以上技术方案,所述数据处理装置 900还配置为:在根据所述目标数据对所述目的节点对应的数据表进行更新之后,通过所述源节点将缓存的数据传送至所述目的节点,其中所述缓存的数据为与所述源节点相邻的上一节点发送至所述源节点的数据。

[0116] 在本申请的一些实施例中,基于以上技术方案,所述锁为基于redis的分布式锁或者基于zookeeper的分布式锁。

[0117] 本申请各实施例中提供的数据处理装置的具体细节已经在对应的方法实施例中进行了详细的描述,此处不再赘述。

[0118] 图10示意性地示出了用于实现本申请实施例的电子设备的计算机系统结构框图,该电子设备可以是如图1中所示的终端设备101和服务器103。

[0119] 需要说明的是,图10示出的电子设备的计算机系统1000仅是一个示例,不应对本申请实施例的功能和使用范围带来任何限制。

[0120] 如图10所示,计算机系统1000包括中央处理器1001(Central Processing Unit, CPU),其可以根据存储在只读存储器1002(Read-Only Memory, ROM)中的程序或者从存储部分1008加载到随机访问存储器 1003(Random Access Memory, RAM)中的程序而执行各种适当的动作和处理。在随机访问存储器1003中,还存储有系统操作所需的各种程序和数据。中央处理器1001、在只读存储器1002以及随机访问存储器1003通过总线1004彼此相连。输入/输出接口1005(Input/Output接口,即I/O接口)也连接至总线1004。

[0121] 在一些实施例中,以下部件连接至输入/输出接口1005:包括键盘、鼠标等的输入部分1006;包括诸如阴极射线管(Cathode Ray Tube, CRT)、液晶显示器(Liquid Crystal



Display, LCD) 等以及扬声器等的输出部分 1007; 包括硬盘等的存储部分 1008; 以及包括诸如局域网卡、调制解调器等的网络接口卡的通信部分 1009。通信部分 1009 经由诸如因特网的网络执行通信处理。驱动器 1010 也根据需要连接至输入/输出接口 1005。可拆卸介质 1011, 诸如磁盘、光盘、磁光盘、半导体存储器等等, 根据需要安装在驱动器 1010 上, 以便于从其上读出的计算机程序根据需要被安装入存储部分 1008。

[0122] 特别地, 根据本申请的实施例, 各个方法流程图中所描述的过程可以被实现为计算机软件程序。例如, 本申请的实施例包括一种计算机程序产品, 其包括承载在计算机可读介质上的计算机程序, 该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中, 该计算机程序可以通过通信部分 1009 从网络上被下载和安装, 和/或从可拆卸介质 1011 被安装。在该计算机程序被中央处理器 1001 执行时, 执行本申请的系统中限定的各种功能。

[0123] 需要说明的是, 本申请实施例所示的计算机可读介质可以是计算机可读信号介质或者计算机可读介质或者是上述两者的任意组合。计算机可读介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件, 或者任意以上的组合。计算机可读介质的更具体的例子可以包括但不限于: 具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器 (RAM)、只读存储器 (ROM)、可擦式可编程只读存储器 (Erasable Programmable Read Only Memory, EPROM)、闪存、光纤、便携式紧凑磁盘只读存储器 (Compact Disc Read-Only Memory, CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本申请中, 计算机可读介质可以是任何包含或存储程序的有形介质, 该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本申请中, 计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号, 其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式, 包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读信号介质还可以是计算机可读介质以外的任何计算机可读介质, 该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输, 包括但不限于: 无线、有线等等, 或者上述的任意合适的组合。

[0124] 附图中的流程图和框图, 图示了按照本申请各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上, 流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分, 上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意, 在有些作为替换的实现中, 方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如, 两个接连地表示的方框实际上可以基本并行地执行, 它们有时也可以按相反的顺序执行, 这依所涉及的功能而定。也要注意, 框图或流程图中的每个方框、以及框图或流程图中的方框的组合, 可以用执行规定的功能或操作的专用的基于硬件的系统来实现, 或者可以用专用硬件与计算机指令的组合来实现。

[0125] 应当注意, 尽管在上文详细描述中提及了用于动作执行的设备的若干模块或者单元, 但是这种划分并非强制性的。实际上, 根据本申请的实施方式, 上文描述的两个或更多模块或者单元的特征和功能可以在一个模块或者单元中具体化。反之, 上文描述的一个模块或者单元的特征和功能可以进一步划分为由多个模块或者单元来具体化。

[0126] 通过以上的实施方式的描述,本领域的技术人员易于理解,这里描述的示例实施方式可以通过软件实现,也可以通过软件结合必要的硬件的方式来实现。因此,根据本申请实施方式的技术方案可以以软件产品的形式体现出来,该软件产品可以存储在一个非易失性存储介质(可以是CD-ROM,U 盘,移动硬盘等)中或网络上,包括若干指令以使得一台电子设备执行根据本申请实施方式的方法。

[0127] 本领域技术人员在考虑说明书及实践这里公开的发明后,将容易想到本申请的其它实施方案。本申请旨在涵盖本申请的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本申请的一般性原理并包括本申请未公开的本技术领域中的公知常识或惯用技术手段。

[0128] 应当理解的是,本申请并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本申请的范围仅由所附的权利要求来限制。

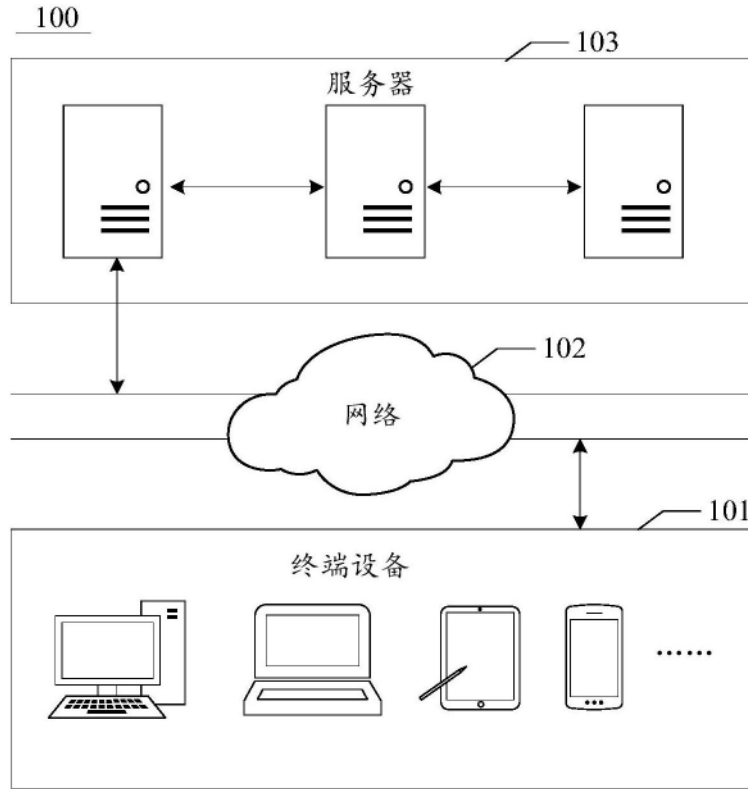


图1

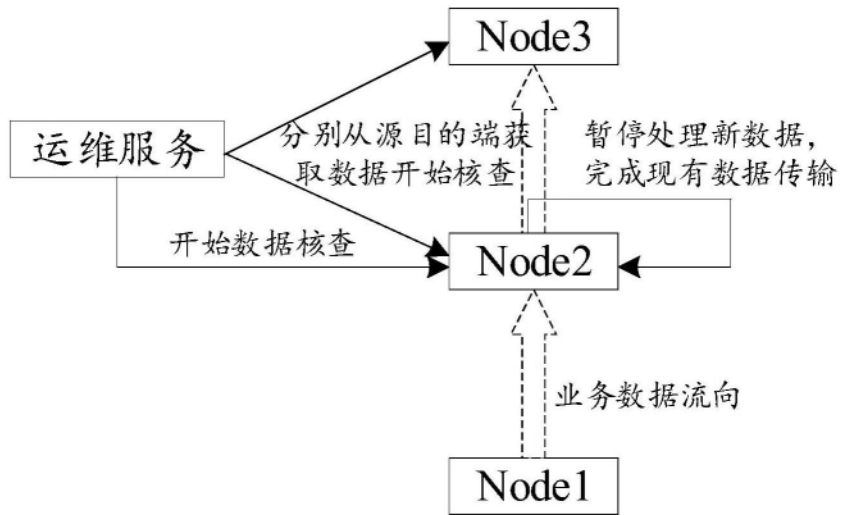


图2

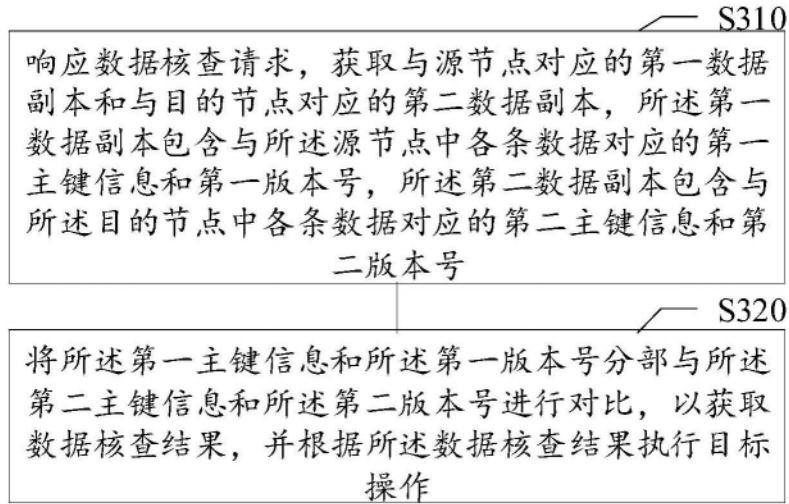


图3

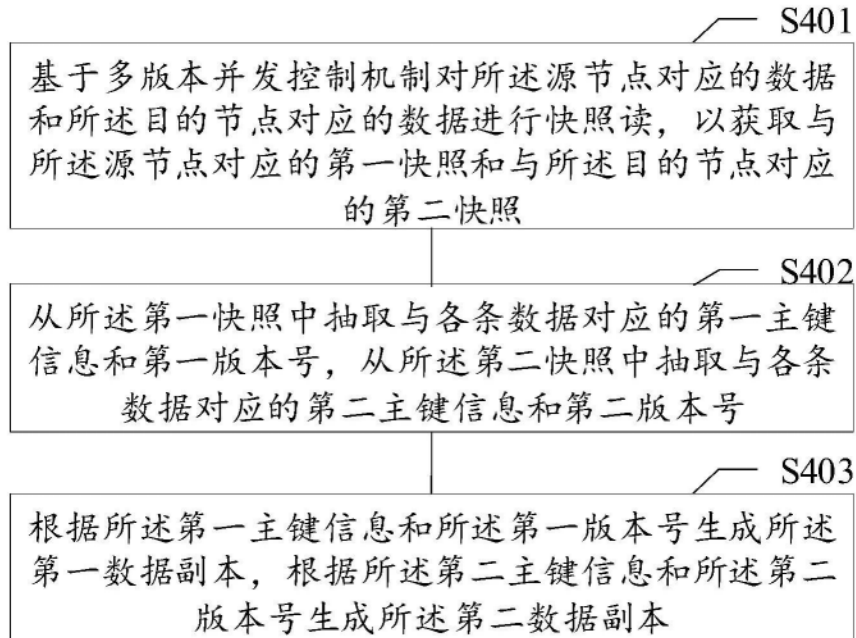


图4

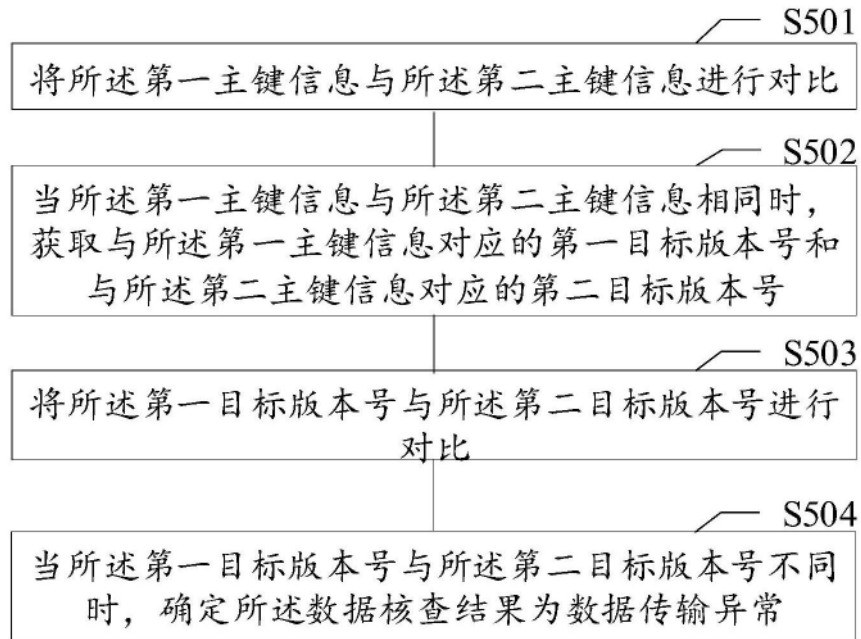


图5

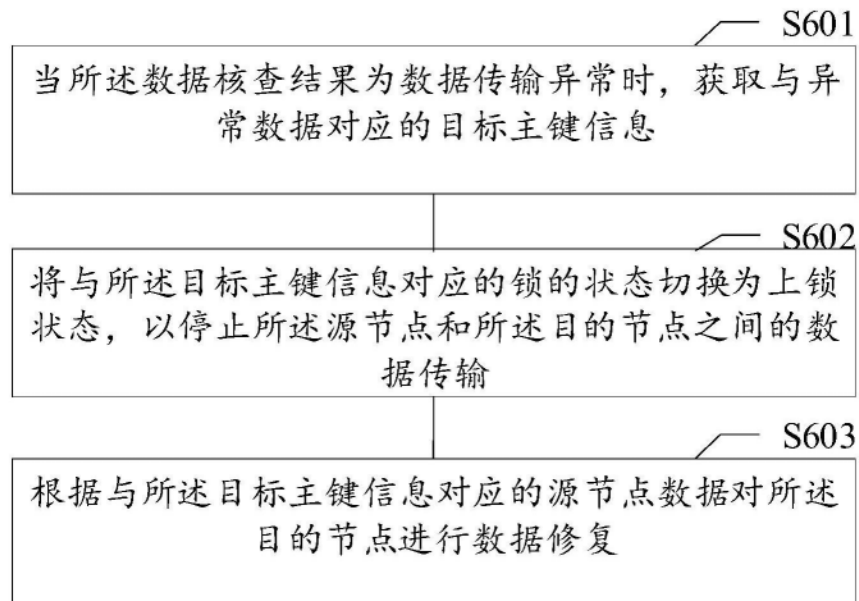


图6

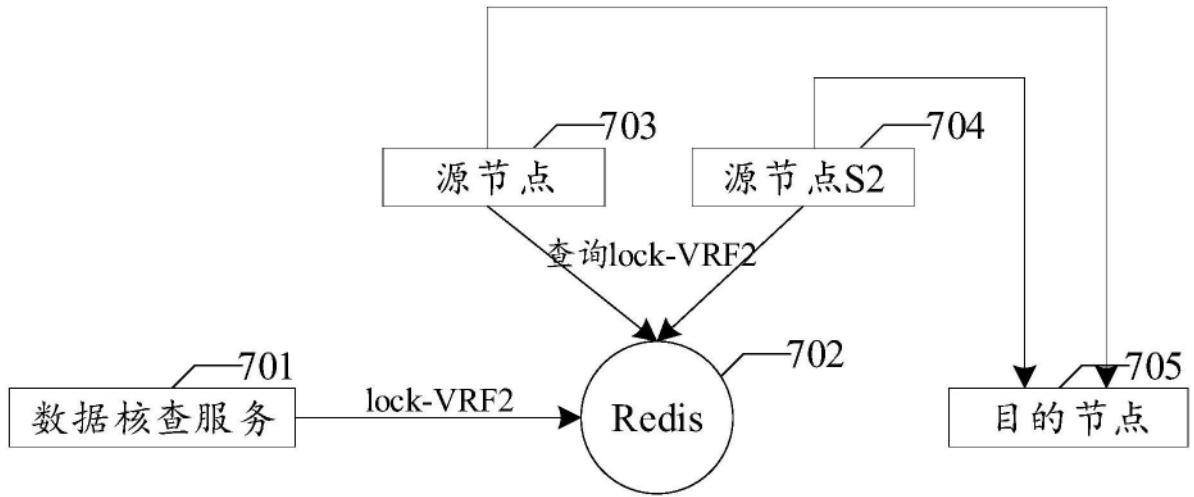


图7

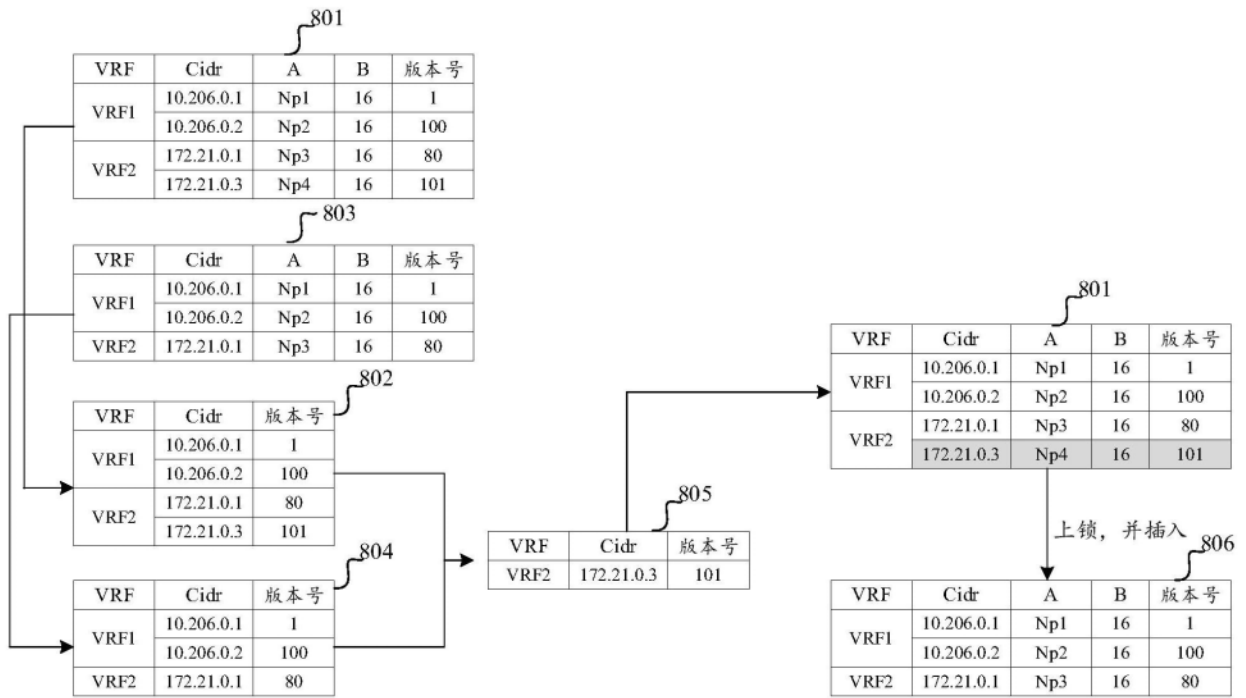


图8

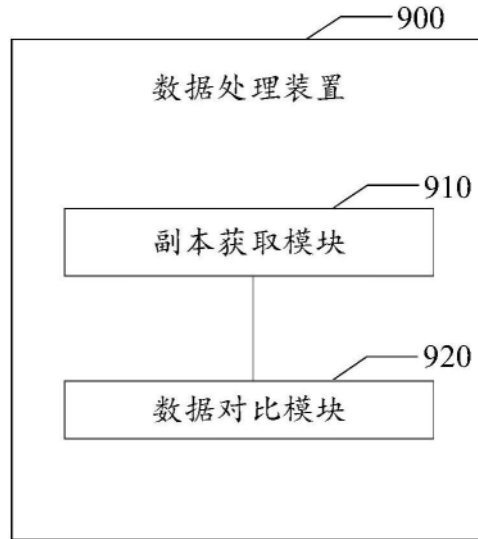


图9

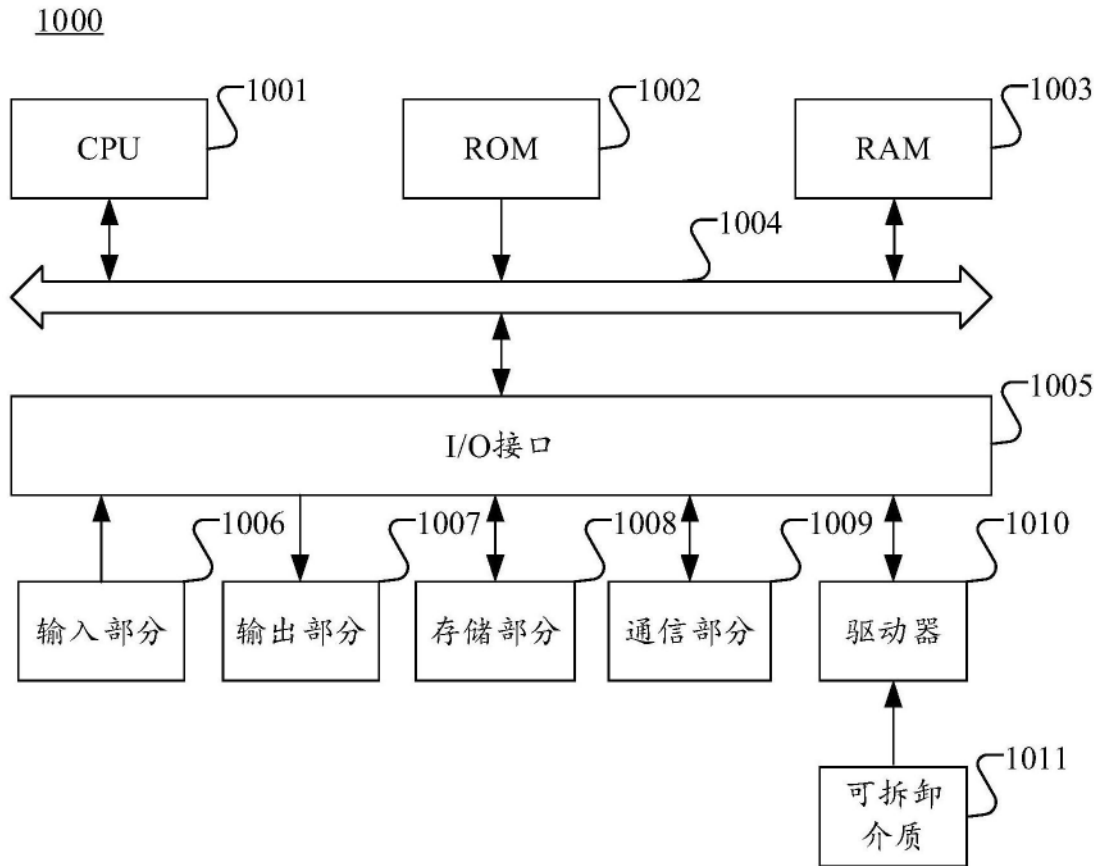


图10