



(12) 发明专利申请

(10) 申请公布号 CN 116501789 A

(43) 申请公布日 2023. 07. 28

(21) 申请号 202310770768.6

(22) 申请日 2023.06.28

(71) 申请人 本原数据(北京)信息技术有限公司
地址 100000 北京市海淀区知春路27号15
层1702-1703室

(72) 发明人 陈泽

(74) 专利代理机构 广州嘉权专利商标事务所有
限公司 44205
专利代理师 洪铭福

(51) Int. Cl.

G06F 16/25 (2019.01)

G06F 16/242 (2019.01)

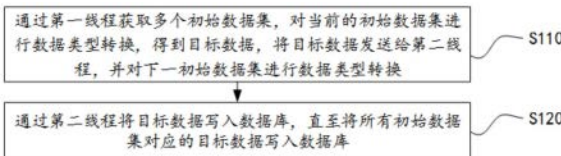
权利要求书2页 说明书10页 附图4页

(54) 发明名称

数据库数据写入方法和装置、电子设备及存储介质

(57) 摘要

本申请提供了一种数据库数据写入方法和装置、电子设备及存储介质,属于大数据处理技术领域,通过第一线程获取多个初始数据集,对当前的初始数据集进行数据类型转换,得到目标数据,将目标数据发送给第二线程,并对下一初始数据集进行数据类型转换,通过第二线程将目标数据写入数据库,直至将所有初始数据集对应的目标数据写入数据库,提高了数据写入数据库的效率。



1. 数据库数据写入方法,其特征在于,所述方法包括:

通过第一线程获取多个初始数据集,对当前的所述初始数据集进行数据类型转换,得到目标数据,将所述目标数据发送给第二线程,并对下一所述初始数据集进行数据类型转换;

通过所述第二线程将所述目标数据写入数据库,直至将所有初始数据集对应的所述目标数据写入所述数据库。

2. 根据权利要求1所述的数据库数据写入方法,其特征在于,所述通过第一线程获取多个初始数据集,对当前的所述初始数据集进行数据类型转换,得到目标数据,包括:

通过所述第一线程获取数据插入语句,对所述数据插入语句进行语法解析,生成语法树;所述数据插入语句包括预设数据;

对所述预设数据进行分批,得到多个初始数据集;

通过所述语法树对当前的所述初始数据集进行数据类型转换,得到所述目标数据。

3. 根据权利要求1所述的数据库数据写入方法,其特征在于,所述通过所述第二线程将所述目标数据写入数据库,包括:

通过所述第二线程将所述目标数据写入预设缓存;

当所述预设缓存中所述目标数据的数量达到预设数量,将所述预设数量个目标数据写入所述数据库。

4. 根据权利要求3所述的数据库数据写入方法,其特征在于,所述预设缓存包括多个缓存分区,所述通过所述第二线程将所述目标数据写入预设缓存,包括:

获取所述目标数据的数据事务类型;

通过所述第二线程,根据所述数据事务类型将所述目标数据写入所述缓存分区。

5. 根据权利要求3所述的数据库数据写入方法,其特征在于,在所述通过所述第二线程将所述目标数据写入预设缓存之后,所述数据库数据写入方法还包括:

当所述预设缓存中所述目标数据的数量达到预设数量,启动第三线程;

通过所述第三线程将所述预设数量个目标数据写入所述数据库。

6. 根据权利要求1至5任一项所述的数据库数据写入方法,其特征在于,所述通过所述第二线程将所述目标数据写入数据库,包括:

对所述目标数据进行加密;

通过所述第二线程将加密后的所述目标数据写入所述数据库。

7. 根据权利要求1至5任一项所述的数据库数据写入方法,其特征在于,所述通过所述第二线程将所述目标数据写入数据库,包括:

获取所述数据库的剩余存储容量以及所述目标数据的数据容量;

若所述数据容量小于或者等于所述剩余存储容量,则通过所述第二线程将所述目标数据写入所述数据库。

8. 数据库数据写入装置,其特征在于,所述装置包括:

数据解析模块,用于通过第一线程获取多个初始数据集,对当前的所述初始数据集进行数据类型转换,得到目标数据,将所述目标数据发送给第二线程,并对下一所述初始数据集进行数据类型转换;

数据写入模块,用于通过所述第二线程将所述目标数据写入数据库,直至将所有初始

数据集对应的所述目标数据写入所述数据库。

9. 电子设备,其特征在于,所述电子设备包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现权利要求1至7任一项所述的数据库数据写入方法。

10. 计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至7中任一项所述的数据库数据写入方法。

数据库数据写入方法和装置、电子设备及存储介质

技术领域

[0001] 本申请涉及大数据处理技术领域,尤其涉及一种数据库数据写入方法和装置、电子设备及存储介质。

背景技术

[0002] 在数据库领域通常需要面临大量数据不断写入的场景,由于客户端发送给数据库的数据为字符串,数据库接收到字符串后需要对海量数据进行数据类型转换,必须等待全部数据解析完毕后,才执行数据的写入操作,但是这种数据写入方式比较耗时,降低了数据写入数据库的效率。

发明内容

[0003] 本申请实施例的主要目的在于提出一种数据库数据写入方法和装置、电子设备及存储介质,旨在提高数据写入数据库的效率。

[0004] 为实现上述目的,本申请实施例的第一方面提出了一种数据库数据写入方法,所述方法包括:

通过第一线程获取多个初始数据集,对当前的所述初始数据集进行数据类型转换,得到目标数据,将所述目标数据发送给第二线程,并对下一所述初始数据集进行数据类型转换;

通过所述第二线程将所述目标数据写入数据库,直至将所有初始数据集对应的所述目标数据写入所述数据库。

[0005] 在一些实施例,所述通过第一线程获取多个初始数据集,对当前的所述初始数据集进行数据类型转换,得到目标数据,包括:

通过所述第一线程获取数据插入语句,对所述数据插入语句进行语法解析,生成语法树;所述数据插入语句包括预设数据;

对所述预设数据进行分批,得到多个初始数据集;

通过所述语法树对当前的所述初始数据集进行数据类型转换,得到所述目标数据。

[0006] 在一些实施例,所述通过所述第二线程将所述目标数据写入数据库,包括:

通过所述第二线程将所述目标数据写入预设缓存;

当所述预设缓存中所述目标数据的数量达到预设数量,将所述预设数量个目标数据写入所述数据库。

[0007] 在一些实施例,所述预设缓存包括多个缓存分区,所述通过所述第二线程将所述目标数据写入预设缓存,包括:

获取所述目标数据的数据事务类型;

通过所述第二线程,根据所述数据事务类型将所述目标数据写入所述缓存分区。

[0008] 在一些实施例,在所述通过所述第二线程将所述目标数据写入预设缓存之后,所

述数据库数据写入方法还包括：

当所述预设缓存中所述目标数据的数量达到预设数量，启动第三线程；
通过所述第三线程将所述预设数量个目标数据写入所述数据库。

[0009] 在一些实施例，所述通过所述第二线程将所述目标数据写入数据库，包括：
对所述目标数据进行加密；
通过所述第二线程将加密后的所述目标数据写入所述数据库。

[0010] 在一些实施例，所述通过所述第二线程将所述目标数据写入数据库，包括：
获取所述数据库的剩余存储容量以及所述目标数据的数据容量；
若所述数据容量小于或者等于所述剩余存储容量，则通过所述第二线程将所述目标数据写入所述数据库。

[0011] 为实现上述目的，本申请实施例的第二方面提出了一种数据库数据写入装置，所述装置包括：

数据解析模块，用于通过第一线程获取多个初始数据集，对当前的所述初始数据集进行数据类型转换，得到目标数据，将所述目标数据发送给第二线程，并对下一所述初始数据集进行数据类型转换；

数据写入模块，用于通过所述第二线程将所述目标数据写入数据库，直至将所有初始数据集对应的所述目标数据写入所述数据库。

[0012] 为实现上述目的，本申请实施例的第三方面提出了一种电子设备，所述电子设备包括存储器和处理器，所述存储器存储有计算机程序，所述处理器执行所述计算机程序时实现上述第一方面所述的数据库数据写入方法。

[0013] 为实现上述目的，本申请实施例的第四方面提出了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，所述计算机程序被处理器执行时实现上述第一方面所述的数据库数据写入方法。

[0014] 本申请提出的数据库数据写入方法、数据库数据写入装置、电子设备及计算机可读存储介质，通过第一线程对当前初始数据集进行数据类型转换，得到目标数据，将目标数据发送给第二线程，由第二线程将目标数据写入数据库，无需等待全部初始数据集均完成数据转换，才进行数据写入。在第二线程将目标数据写入数据库的同时，第一线程继续对下一初始数据集进行数据类型转换，通过两个异步线程并行对数据进行处理，以将海量数据批量写入数据库，提高了数据写入数据库的效率。

附图说明

[0015] 图1是本申请实施例提供的数据库数据写入方法的流程图；
图2是图1中的步骤S110的流程图；
图3是图1中的步骤S120的流程图；
图4是图3中的步骤S310的流程图；
图5是本申请实施例提供的数据库数据写入方法的另一流程图；
图6是图1中的步骤S120的另一流程图；
图7是图1中的步骤S120的另一流程图；
图8是本申请实施例提供的数据库数据写入方法的示意图；

图9是本申请实施例提供的数据库数据写入方法的另一示意图；
图10是本申请实施例提供的数据库数据写入装置的结构示意图；
图11是本申请实施例提供的电子设备的硬件结构示意图。

具体实施方式

[0016] 为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本申请，并不用于限定本申请。

[0017] 需要说明的是，虽然在装置示意图中进行了功能模块划分，在流程图中示出了逻辑顺序，但是在某些情况下，可以以不同于装置中的模块划分，或流程图中的顺序执行所示出或描述的步骤。说明书和权利要求书及上述附图中的术语“第一”、“第二”等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。

[0018] 除非另有定义，本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请实施例的目的，不是旨在限制本申请。

[0019] 在数据库领域通常需要面临大量数据不断写入的场景，大部分主流关系型数据库支持insert和copy两种方式进行数据写入。通过copy进行数据写入时，需要数据全部写入完成才会提交事务，数据库日志才会落盘。在copy导入数据的过程中一旦出现错误，全部的数据都不会写入数据库。copy还要求数据格式需要与表结构一致，需要用户提前组织数据格式，使得copy只适合在特定的情况下使用，例如从文件中加载数据，不能用于实时数据写入。此外，copy操作需要文件系统访问权限，这可能会使其在某些环境中受到限制。insert可以选择一次性写入一定数目的数据，若一次insert出现错误，只会影响该次需要写入的数据。这样一次insert是一个事务，并且每条数据都有一条日志。copy虽然数据写入效率高，但因为是单个事务，导致其使用限制比较多。相较于copy，insert比较灵活，因此通常使用insert方式进行数据导入。

[0020] 对于大批量数据的写入，通常会使用“INSERT INTO xxx VALUES (xx,xx), (xx,xx), (xx,xx)”这样形式的一条结构化查询语句(Structured Query Language,SQL)插入多条数据。但是这种数据插入方式的解析次数随着数据量的增加而线性增长。若数据总条数为N，一条insert语句每次插入的数据条数为10，则需要解析inset语句的次数为N/10次。

[0021] 为了减少SQL语句的解析次数，通过使用parse、bind和execute分开的执行模式。分开后的模型会让之前解析过的SQL语句跳过解析流程。若后续再来一个insert语句也是插入10条数据，那么可以不解析这条SQL语句，直接使用之前解析该insert语句保存的语法树。但若不是10个，而是15个或其它未解析过的数量，则仍然需要解析SQL语句并保存语法树。分开模型需要解析的次数取决于insert语句每批数据的数量变化，如果都是10条数据，那么SQL语句只需解析一次。

[0022] 于是写入流程在客户端表现为：“PREPARE xxx AS INSERT INTO xxx(column1, column2,...) VALUES (\$1,\$2,...), (\$10,\$11,...) ...;EXECUTE xxx(value1, value2,...,value10,value11,...);”数据库接收到该SQL语句后，在parse阶段，对PREPARE中的SQL语句进行解析，生成语法树并缓存。在bind阶段，对EXECUTE语句进行解析，

将传入的各位置上的参数值字符串转换为对应的数据类型。在execute阶段,执行数据插入操作,将所有参数值按行依次插入数据库的表中。

[0023] 为了优化这种批量插入方式,oracle等数据库提供了batch方式绑定写入。具体来说,就是在PREPARE阶段仅解析一条数据,在EXECUTE时仍然传入多条数据,数据库使用第一次解析得到的语法树,重复执行bind、execute完成批量写入。对于优化了的批量插入方式,parse、bind、execute阶段由多事务优化为单事务。在数据插入的过程中,由于单条数据的类型以及单批数据的长度,bind阶段的时间有时候也会比较长,甚至能与execute阶段耗时差不多。

[0024] 由于客户端发送给数据库的数据为字符串,数据库接收到字符串后在bind阶段需要对海量数据进行数据类型转换,必须等待全部数据解析完毕后,才由execute执行数据的写入操作。但是这种数据写入方式比较耗时,降低了数据写入数据库的效率。

[0025] 基于此,本申请实施例提供了一种数据库数据写入方法、数据库数据写入装置、电子设备及计算机可读存储介质,旨在提高数据写入数据库的效率。

[0026] 本申请实施例提供的数据库数据写入方法、数据库数据写入装置、电子设备及计算机可读存储介质,具体通过如下实施例进行说明,首先描述本申请实施例中的数据库数据写入方法。

[0027] 本申请实施例提供的数据库数据写入方法,涉及大数据处理技术领域。本申请实施例提供的数据库数据写入方法可应用于终端中,也可应用于服务器端中,还可以是运行于终端或服务器端中的软件。在一些实施例中,终端可以是智能手机、平板电脑、笔记本电脑、台式计算机等;服务器端可以配置成独立的物理服务器,也可以配置成多个物理服务器构成的服务器集群或者分布式系统,还可以配置成提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN以及大数据和人工智能平台等基础云计算服务的云服务器;软件可以是实现数据库数据写入方法的应用等,但并不局限于以上形式。

[0028] 本申请可用于众多通用或专用的计算机系统环境或配置中。例如:个人计算机、服务器计算机、手持设备或便携式设备、平板型设备、多处理器系统、基于微处理器的系统、置顶盒、可编程的消费电子设备、网络PC、小型计算机、大型计算机、包括以上任何系统或设备的分布式计算环境等等。本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0029] 图1是本申请实施例提供的数据库数据写入方法的一个可选的流程图,图1中的方法可以包括但不限于包括步骤S110至步骤S120。

[0030] 步骤S110,通过第一线程获取多个初始数据集,对当前的初始数据集进行数据类型转换,得到目标数据,将目标数据发送给第二线程,并对下一初始数据集进行数据类型转换;

步骤S120,通过第二线程将目标数据写入数据库,直至将所有初始数据集对应的目标数据写入数据库。

[0031] 本申请实施例所示意的步骤S110至步骤S120,通过第一线程进行数据类型转换,通过第二线程将数据类型转换后的数据写入数据库,在第二线程进行数据写入的过程中,第一线程仍然进行数据类型转换,以异步线程的方式提高了数据写入数据库的效率。

[0032] 请参阅图2,在一些实施例中,步骤S110可以包括但不限于包括步骤S210至步骤S230:

步骤S210,通过第一线程获取数据插入语句,对数据插入语句进行语法解析,生成语法树;数据插入语句包括预设数据;

步骤S220,对预设数据进行分批,得到多个初始数据集;

步骤S230,通过语法树对当前的初始数据集进行数据类型转换,得到目标数据。

[0033] 在一些实施例的步骤S210中,数据库开启第一线程和第二线程。客户端向数据库发送数据插入语句,数据库的第一线程接收到数据插入语句后,在parse阶段对数据插入语句进行SQL语句解析,生成语法树并缓存。SQL语句解析是利用flex、bison这类工具识别SQL语句中的关键字,将字符串转变为特定结构体,该特定结构体便为语法树。数据插入语句为:“PREPARE xxx AS INSERT INTO xxx(column1,column2,...) VALUES(\$1,\$2,...),(\$10,\$11,...),...;EXECUTE xxx(value1,value2,...,value10,value11,...);”,对PREPARE中的SQL语句进行解析,得到一个insert结构体。inset结构体可以为:

```
{type = "insert",
target = "xxx",
column {
column1: type = xxx, name = xxx, ...,
column2: type = xxx, name = xxx, ...,
.... }
}
```

insert结构体包括目标表target,目标表target包含的参数数量以及每个参数位置上参数的类型等等。在之后流程中直接使用该结构体可以快速取得insert语句所需的信息。

[0034] 预设数据为数据插入语句的多个数据,例如,(\$1,\$2,...)、(\$10,\$11,...)…等均为预设数据,其中VALUE后一个括号内的数据为一个数据。客户端需要将大批量数据插入数据库,数据的个数是一个比较大的数值。

[0035] 在一些实施例的步骤S220中,数据库设置数据阈值,根据数据阈值对预设数据进行分批,以控制批量插入时每批插入数据库的数据数量。与数据阈值相等数量的数据为一个批,得到多个初始数据集。可以理解的是,每批的数据阈值可以相同,也可以不同。若数据阈值相同,则不需要执行parse阶段的SQL语句解析,直接使用解析生成的语法树即可,然后重复执行bind以及execute。若数据阈值不同,且该数据阈值的SQL语句从未被解析过,则需要对SQL语句进行解析。

[0036] 在一些实施例的步骤S230中,客户端发送给数据库的数据为字符串,数据库接收到数据后,需要将字符串转换为插入数据的数据库表各个列使用的数据类型。例如,插入数据的数据库表为成绩单表,成绩单表有姓名列、分数列,姓名列的数据类型是字符类型varchar,分数列的数据类型是浮点型float。通过语法树中包含的每个参数位置上参数的

类型对当前的初始数据集进行数据类型转换,得到目标数据。可以理解的是,一个数据的数据长度为各列数据类型所占字节之和。单批数据的数据长度为单批数据包括的各个数据的数据长度之和,也就是单批数据的数据个数与一个数据的数据长度的乘积。若各列数据类型分别为varchar和float类型,float占4个字节,varchar占4个字节,该数据的数据长度为8。

[0037] 在数据类型转换过程中,类型转换耗时受数据类型以及数据长度的影响。当表含有的列比较少、数据类型为简单格式时,类型转换耗时相对较小。当表含有的列比较多,达到百列千列,数据类型是timetz或其它解析代价比较高的类型时,类型转换耗时较长。若execute一直等待bind完成全部数据类型转换后,才将转换完成的数据写入数据库,需要耗费大量的时间。为了解决bind耗时的问题,对数据进行分批,当第一线程完成对当前批初始数据集的解析、数据类型转换任务后,将当前批初始数据集对应的目标数据发送给第二线程,并继续执行对下一批初始数据集进行解析、数据类型转换的任务。execute无需等待全部数据完成数据类型转换后,才进行数据写入操作。

[0038] 通过上述步骤S210至步骤S230,第一线程可以持续对初始数据集进行解析、数据类型转换等操作,无需等待当前初始数据集写入数据库后,才进行下一初始数据集的解析、数据类型转换,通过异步线程的方式减少了execute的等待时间,提高了数据写入的效率。同时,对数据进行分批处理,当某行插入失败的情况下,减少对其他批数据写入操作的影响。

[0039] 请参阅图3,在一些实施例中,步骤S120可以包括但不限于包括步骤S310至步骤S320:

步骤S310,通过第二线程将目标数据写入预设缓存;

步骤S320,当预设缓存中目标数据的数量达到预设数量,将预设数量个目标数据写入数据库。

[0040] 在一些实施例的步骤S310中,通过第二线程将目标数据写入数据库,直至将所有初始数据集对应的目标数据写入数据库,向客户端返回成功写入的提示。在大批量数据写入的场景中,用于分批的数据阈值设置得过于小时,必然会频繁触发数据库的检查点,将日志中的数据写入磁盘数据文件。频繁的刷新缓冲区以及切换日志,会导致大批量写入数据性能的降低。为了提高数据批量写入的性能,通过第二线程将目标数据写入预设缓存,预设缓存可以是数据库所在服务器的本地缓存,也可以是远程缓存。

[0041] 在一些实施例的步骤S320中,当预设缓存中目标数据的数量等于预设数量,将预设数量个目标数据写入数据库,避免频繁写入造成数据写入性能的降低。

[0042] 通过上述步骤S310至步骤S320,能够避免频繁写入造成的数据写入性能降低。

[0043] 请参阅图4,在一些实施例中,预设缓存包括多个缓存分区,步骤S310可以包括但不限于包括步骤S410至步骤S420:

步骤S410,获取目标数据的数据事务类型;

步骤S420,通过第二线程,根据数据事务类型将目标数据写入缓存分区。

[0044] 在一些实施例的步骤S410中,数据事务类型为目标数据所属的类别,可以是成绩、课程。具体地,在获取目标数据的数据事务类型时,可以根据目标数据对应的数据名称,查询目标数据所属的分组,根据目标数据所属的分组,确定目标数据的数据事物类型。

[0045] 在一些实施例的步骤S420中,为了快速定位目标数据在预设缓存中的存储区域,从而提高数据写入的效率,根据数据事务类型对目标数据进行分区存储。每一数据事务类型对应一个缓存分区,通过第二线程,确定数据事务类型对应的缓存分区,将目标数据写入该缓存分区。当缓存分区中目标数据的数量达到数量阈值时,则将与数量阈值相等数目个目标数据写入数据库。由于缓存分区存储的是同一数据事务类型的目标数据,可直接将目标数据写入数据库对应的表,避免根据各目标数据进行数据库表结构查询,才将目标数据写入到对应的表中。

[0046] 通过上述步骤S410至步骤S420,能够提高数据写入预设缓存的效率,从而提高数据写入数据库的效率。

[0047] 请参阅图5,在一些实施例中,在步骤S310之后,数据库数据写入方法还可以包括但不限于包括步骤S510至步骤S520:

步骤S510,当预设缓存中目标数据的数量达到预设数量,启动第三线程;

步骤S520,通过第三线程将预设数量个目标数据写入数据库。

[0048] 在一些实施例的步骤S510中,为了进一步提高数据写入数据库的效率,开启第三线程,通过第二线程将目标数据写入预设缓存,通过第三线程将预设缓存中的目标数据写入数据库。通过分离目标数据写入预设缓存的任务和预设缓存中的目标数据写入数据库的任务,提高了目标数据写入数据库的效率。当预设缓存中目标数据的数量达到预设数量,启动第三线程。

[0049] 在一些实施例的步骤S520中,通过第三线程将预设数量个目标数据写入数据库。

[0050] 通过上述步骤S510至步骤S520,能够大幅度提升目标数据写入数据库的效率。

[0051] 请参阅图6,在一些实施例中,步骤S120还可以包括但不限于包括步骤S610至步骤S620:

步骤S610,对目标数据进行加密;

步骤S620,通过第二线程将加密后的目标数据写入数据库。

[0052] 在一些实施例的步骤S610中,为了保证目标数据的安全性,对目标数据进行加密。具体地,获取客户端对外公布的公钥,利用公钥对目标数据进行加密。客户端从数据库读取加密后的目标数据,利用与该公钥匹配的私钥对加密后的目标数据进行解密。

[0053] 在一些实施例的步骤S620中,通过第二线程将加密后的目标数据写入数据库。可获取目标数据的数据事务类型,通过第二线程根据数据事务类型将加密后的目标数据写入预设缓存的缓存分区,当预设缓存中目标数据的数量达到预设数量,通过第三线程将预设缓存中的目标数据写入数据库。或者,缓存分区中目标数据的数量达到数量阈值,通过第三线程将该缓存分区中的目标数据写入数据库。

[0054] 通过上述步骤S610至步骤S620,能够保证数据写入数据库时的安全性。

[0055] 请参阅图7,在一些实施例中,步骤S120还可以包括但不限于包括步骤S710至步骤S720:

步骤S710,获取数据库的剩余存储容量以及目标数据的数据容量;

步骤S720,若数据容量小于或者等于剩余存储容量,则通过第二线程将目标数据写入数据库。

[0056] 在一些实施例的步骤S710中,为了避免数据库存储空间不足,导致数据写入失败,

需要获取数据库的剩余存储容量以及目标数据的数据容量,以判断目标数据是否可以写入数据库。

[0057] 在一些实施例的步骤S720中,若数据容量小于或者等于剩余存储容量,说明目标数据可以写入数据库,则通过第二线程将目标数据写入数据库。若数据容量大于剩余存储容量,说明数据库存储空间不足,需要重新开辟数据库存储空间,以将目标数据写入新的数据库存储空间。

[0058] 通过上述步骤S710至步骤S720,能够避免因数据库存储空间不足,导致数据写入失败。

[0059] 在步骤S120之后,当目标数据写入失败时,可以提示目标数据所属的批次,以使客户端可以对更少的插入失败数据进行重新插入,提升用户体验。

[0060] 如图8所示,一个客户端发来N条数据,需要先等待bind完成N条数据的参数值解析转换,才能继续execute。数据写入过程始终只使用了一个cpu核心,数据写入数据库的效率比较低。为了提高数据写入数据库的效率,本申请实施例灵活控制批量插入时每批数据的数量,并分离bind和execute的执行线程。

[0061] 如图9所示,将N条数据拆分为M份,当bind完成了第一份数据的解析转换工作后,execute就可以直接开始第一份数据的写入,同时bind可以继续第二份数据的处理。一个线程对应一个cpu核心,通过bind与execute的异步,使用了两个cpu核心,大幅提升原有的批量数据写入流程中bind和execute阶段的效率。

[0062] 请参阅图10,本申请实施例还提供一种数据库数据写入装置,可以实现上述数据库数据写入方法,该装置包括:

数据解析模块1010,用于通过第一线程获取多个初始数据集,对当前的初始数据集进行数据类型转换,得到目标数据,将目标数据发送给第二线程,并对下一初始数据集进行数据类型转换;

数据写入模块1020,用于通过第二线程将目标数据写入数据库,直至将所有初始数据集对应的目标数据写入数据库。

[0063] 该数据库数据写入装置的具体实施方式与上述数据库数据写入方法的具体实施例基本相同,在此不再赘述。

[0064] 本申请实施例还提供了一种电子设备,电子设备包括括存储器和处理器,存储器存储有计算机程序,处理器执行计算机程序时实现上述数据库数据写入方法。该电子设备可以为包括平板电脑、车载电脑等任意智能终端。

[0065] 请参阅图11,图11示意了另一实施例的电子设备的硬件结构,电子设备包括:

处理器1110,可以采用通用的CPU(CentralProcessingUnit,中央处理器)、微处理器、应用专用集成电路(ApplicationSpecificIntegratedCircuit,ASIC)、或者一个或多个集成电路等方式实现,用于执行相关程序,以实现本申请实施例所提供的技术方案;

存储器1120,可以采用只读存储器(ReadOnlyMemory,ROM)、静态存储设备、动态存储设备或者随机存取存储器(RandomAccessMemory,RAM)等形式实现。存储器1120可以存储操作系统和其他应用程序,在通过软件或者固件来实现本说明书实施例所提供的技术方案时,相关的程序代码保存在存储器1120中,并由处理器1110来调用执行本申请实施例的数据库数据写入方法;

输入/输出接口1130,用于实现信息输入及输出;

通信接口1140,用于实现本设备与其他设备的通信交互,可以通过有线方式(例如USB、网线等)实现通信,也可以通过无线方式(例如移动网络、WIFI、蓝牙等)实现通信;

总线1150,在设备的各个组件(例如处理器1110、存储器1120、输入/输出接口1130和通信接口1140)之间传输信息;

其中处理器1110、存储器1120、输入/输出接口1130和通信接口1140通过总线1150实现彼此之间在设备内部的通信连接。

[0066] 本申请实施例还提供了一种计算机可读存储介质,该计算机可读存储介质存储有计算机程序,该计算机程序被处理器执行时实现上述数据库数据写入方法。

[0067] 存储器作为一种非暂态计算机可读存储介质,可用于存储非暂态软件程序以及非暂态性计算机可执行程序。此外,存储器可以包括高速随机存取存储器,还可以包括非暂态存储器,例如至少一个磁盘存储器件、闪存器件、或其他非暂态固态存储器件。在一些实施方式中,存储器可选包括相对于处理器远程设置的存储器,这些远程存储器可以通过网络连接至该处理器。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0068] 本申请实施例提供的数据库数据写入方法、数据库数据写入装置、电子设备及计算机可读存储介质,通过第一线程进行数据类型转换,通过第二线程将数据类型转换后的数据写入数据库,在第二线程进行数据写入的过程中,第一线程仍然进行数据类型转换,以异步线程的方式提高了数据写入数据库的效率。

[0069] 本申请实施例描述的实施例是为了更加清楚的说明本申请实施例的技术方案,并不构成对于本申请实施例提供的技术方案的限定,本领域技术人员可知,随着技术的演变和新应用场景的出现,本申请实施例提供的技术方案对于类似的技术问题,同样适用。

[0070] 本领域技术人员可以理解的是,图中示出的技术方案并不构成对本申请实施例的限定,可以包括比图示更多或更少的步骤,或者组合某些步骤,或者不同的步骤。

[0071] 以上所描述的装置实施例仅仅是示意性的,其中作为分离部件说明的单元可以是或者也可以不是物理上分开的,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0072] 本领域普通技术人员可以理解,上文中所公开方法中的全部或某些步骤、系统、设备中的功能模块/单元可以被实施为软件、固件、硬件及其适当的组合。

[0073] 本申请的说明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,以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,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。

[0074] 应当理解,在本申请中,“至少一个(项)”是指一个或者多个,“多个”是指两个或两个以上。“和/或”,用于描述关联对象的关联关系,表示可以存在三种关系,例如,“A和/或B”可以表示:只存在A,只存在B以及同时存在A和B三种情况,其中A,B可以是单数或者复数。字

符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项(个)”或其类似表达,是指这些项中的任意组合,包括单项(个)或复数项(个)的任意组合。例如,a,b或c中的至少一项(个),可以表示:a,b,c,“a和b”,“a和c”,“b和c”,或“a和b和c”,其中a,b,c可以是单个,也可以是多个。

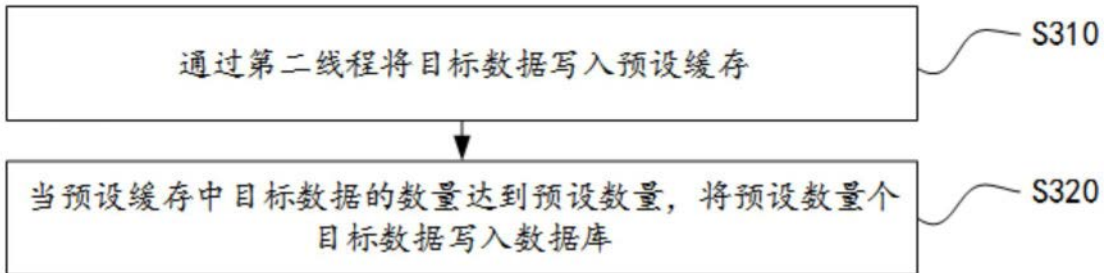
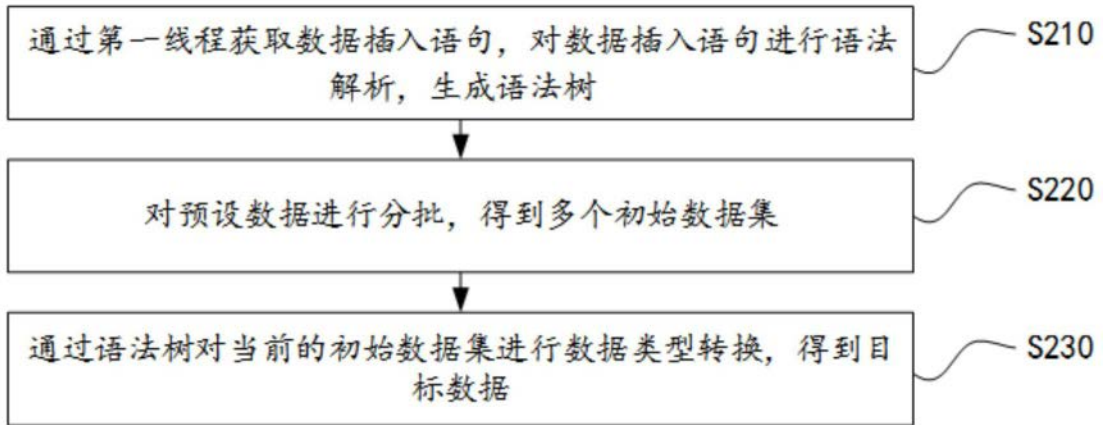
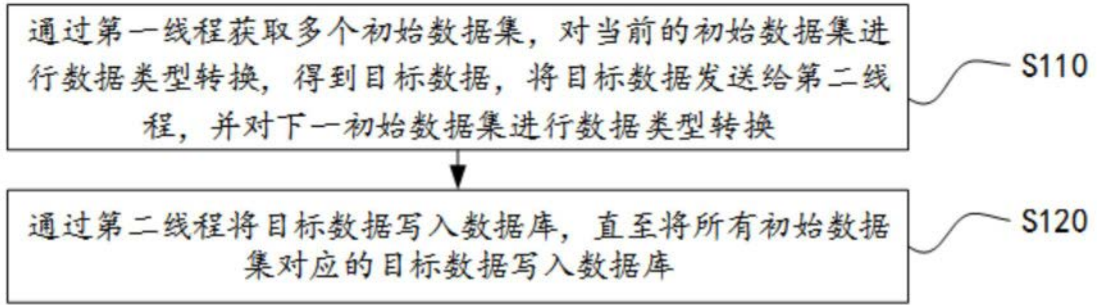
[0075] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,上述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0076] 上述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0077] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0078] 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,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,本申请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,包括多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)执行本申请各个实施例的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,简称ROM)、随机存取存储器(Random Access Memory,简称RAM)、磁碟或者光盘等各种可以存储程序的介质。

[0079] 以上参照附图说明了本申请实施例的优选实施例,并非因此局限本申请实施例的权利范围。本领域技术人员不脱离本申请实施例的范围和实质内所作的任何修改、等同替换和改进,均应在本申请实施例的权利范围之内。



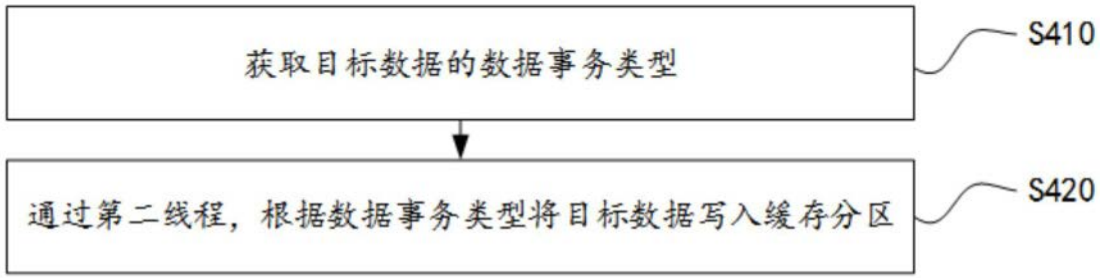


图4

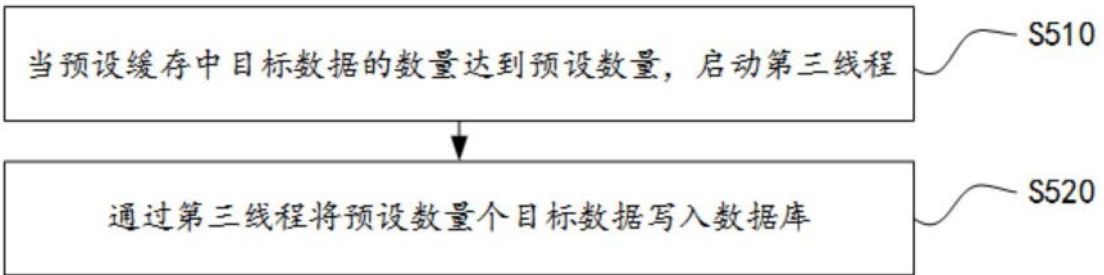


图5

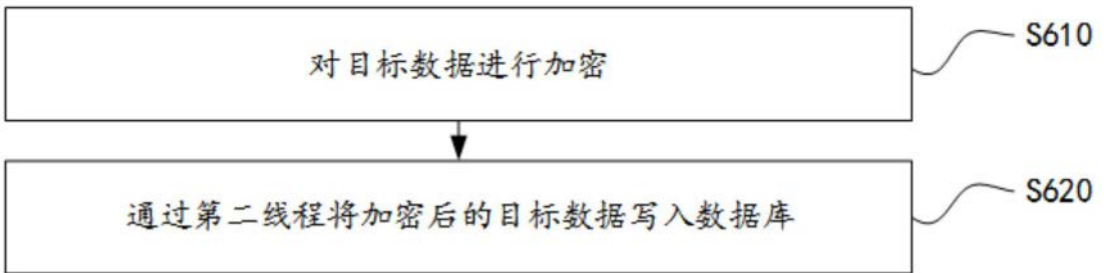


图6

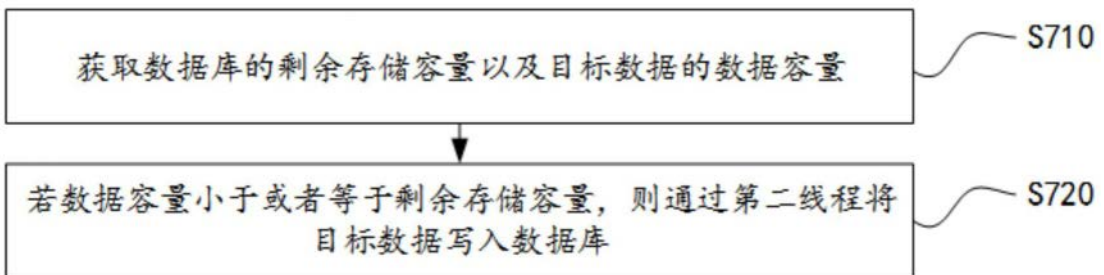


图7

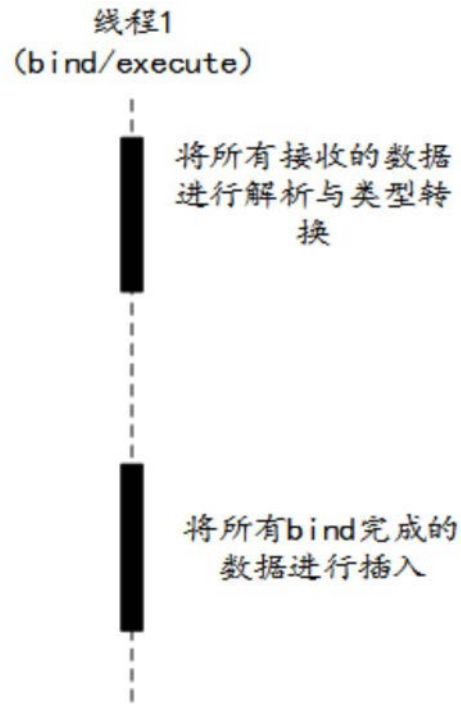


图8

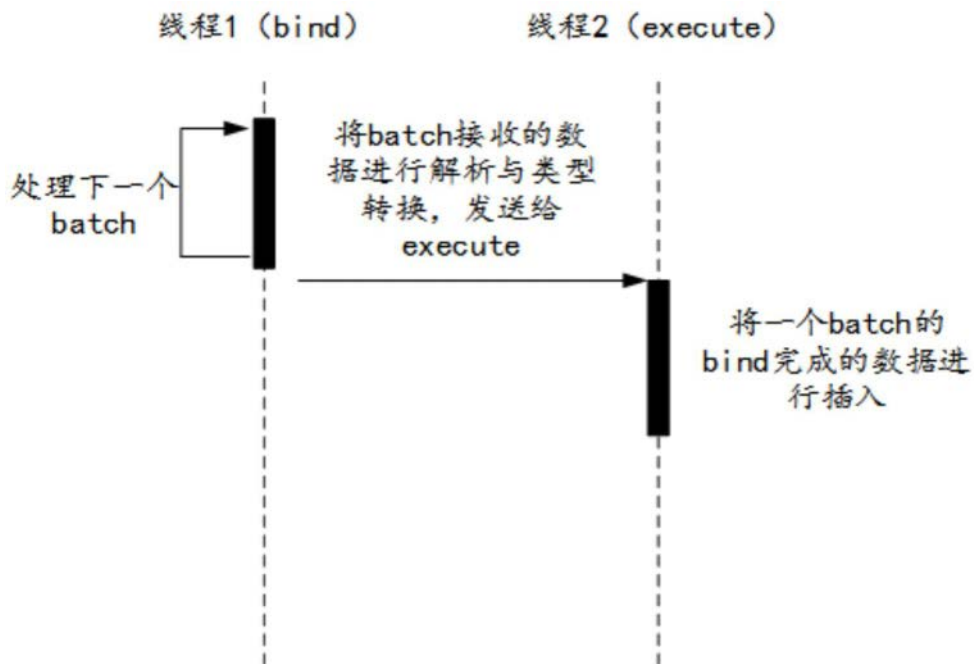


图9

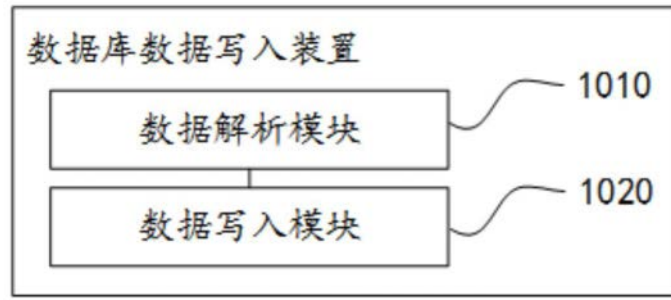


图10

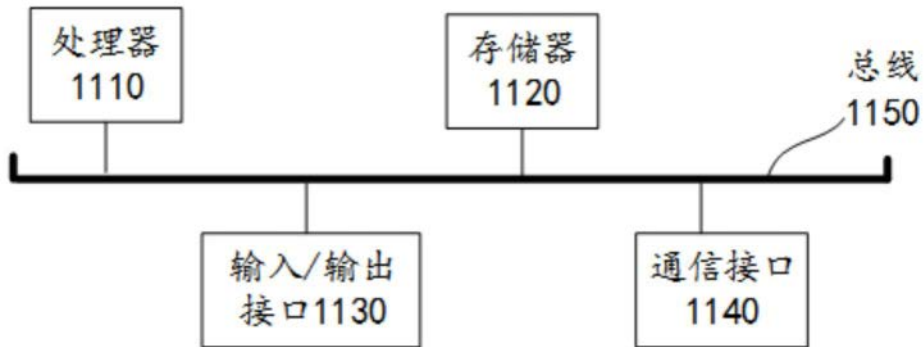


图11