



(12)发明专利申请

(10)申请公布号 CN 108829413 A

(43)申请公布日 2018. 11. 16

(21)申请号 201810426188.4

(22)申请日 2018.05.07

(71)申请人 北京达佳互联信息技术有限公司

地址 100084 北京市海淀区中关村东路1号
院8号楼20层B2201

(72)发明人 刘硕

(74)专利代理机构 北京市立方律师事务所

11330

代理人 刘延喜

(51)Int.Cl.

G06F 8/65(2018.01)

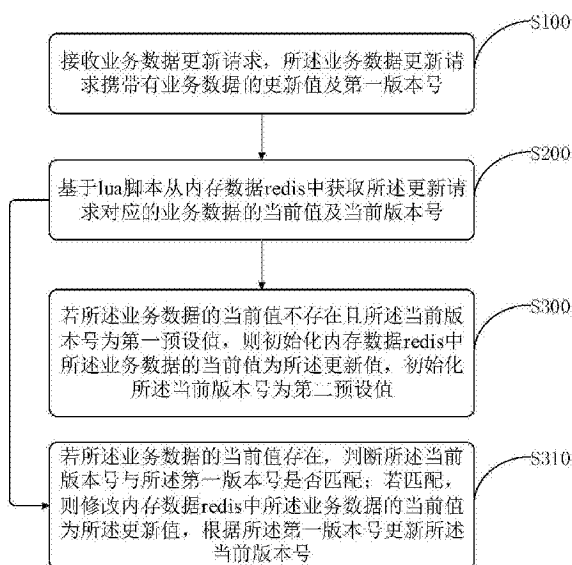
权利要求书2页 说明书7页 附图2页

(54)发明名称

数据更新方法、装置及计算机可读存储介质、服务器

(57)摘要

本发明涉及数据处理领域,具体涉及一种数据更新方法、装置及存储介质、服务器,该方法包括:接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号;基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号;若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。提高在并发环境下对数据的更新的性能及效率。



1. 一种数据更新方法,其特征在于,包括:

接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号;

基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号;

若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;

若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。

2. 根据权利要求1所述的方法,其特征在于,所述根据所述第一版本号更新所述当前版本号之后,还包括:

经过内存数据redis预设的缓存时间后,将所述业务数据的当前值及当前版本号写入持久化数据库中。

3. 根据权利要求1所述的方法,其特征在于,所述内存数据redis中业务数据的数据结构通过lua脚本定义,数据结构的前4个字节为业务数据的版本号,后N个字节为业务数据的值,N为预设正整数。

4. 根据权利要求1所述的方法,其特征在于,所述判断所述当前版本号与所述第一版本号是否匹配之后,还包括:

若不匹配,则向业务数据更新请求方返回业务数据更新失败的提示信息。

5. 根据权利要求1所述的方法,其特征在于,所述根据所述第一版本号更新所述当前版本号,包括:

将所述第一版本号加1作为所述当前版本号。

6. 一种数据更新装置,其特征在于,包括:

接收模块:用于接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号;

获取模块:用于基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号;

第一更新模块:用于若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;

第二更新模块:用于若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。

7. 根据权利要求6所述的装置,其特征在于,所述装置还包括:

写入模块:用于经过内存数据redis预设的缓存时间后,将所述业务数据的当前值及当前版本号写入持久化数据库中。

8. 根据权利要求6所述的装置,其特征在于,所述内存数据redis中业务数据的数据结构通过lua脚本定义,数据结构的前4个字节为业务数据的版本号,后N个字节为业务数据的

值,N为预设正整数。

9.根据权利要求6所述的装置,其特征在于,所述装置还包括:

提示模块:用于在判断所述当前版本号与所述第一版本号是否匹配之后,若不匹配,则向业务数据更新请求方返回业务数据更新失败的提示信息。

10.根据权利要求6所述的装置,其特征在于,所述第二更新模块根据所述第一版本号更新所述当前版本号,包括:

将所述第一版本号加1作为所述当前版本号。

11.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,该程序被处理器执行时实现权利要求1至5任一项所述的数据更新方法。

12.一种服务器,其特征在于,包括:

一个或多个处理器;

存储器;

一个或多个应用程序,其中所述一个或多个应用程序被存储在所述存储器中并被配置为由所述一个或多个处理器执行,所述一个或多个应用程序配置用于执行根据权利要求1至5任一项所述的数据更新方法。

数据更新方法、装置及计算机可读存储介质、服务器

技术领域

[0001] 本发明涉及数据处理领域，具体涉及一种数据更新方法、装置及计算机可读存储介质、服务器。

背景技术

[0002] 在并发环境下对数据的更新操作一般都需要上锁，只有对数据加上锁并且成功之后才能够对其进行操作，否则要等待别的已经获取锁的操作释放才能继续，这种叫悲观锁；还有一种叫乐观锁，即我抢不到锁就不抢了，等我下次想用的时候再来试一下。乐观锁在高并发的应用中有大量的场景需要使用，如购物网站的下单，秒杀等。现有技术方案基于数据库表记录的乐观锁，主要使用一个额外的字段来表示本记录当前的版本号version，只有来加锁的请求带来的版本号和该版本号一致才能够对该记录成功更新。如果有多个线程通知执行，且它们的version都是数据库里对应记录的version，也只能有一个线程操作成功，另外一个失败。数据库的操作是磁盘操作，耗时较高，需要增加额外字段对其进行检索和存储消耗较大。

发明内容

[0003] 本发明的目的在于提供一种数据更新方法、装置及计算机可读存储介质、服务器，解决目前数据更新操作耗时较高、性能及效率较低的问题。

[0004] 为实现该目的，本发明采用如下技术方案：

[0005] 第一方面，本发明提供了一种数据更新方法，包括以下步骤：

[0006] 接收业务数据更新请求，所述业务数据更新请求携带有业务数据的更新值及第一版本号；

[0007] 基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号；

[0008] 若所述业务数据的当前值不存在且所述当前版本号为第一预设值，则初始化内存数据redis中所述业务数据的当前值为所述更新值，初始化所述当前版本号为第二预设值；

[0009] 若所述业务数据的当前值存在，判断所述当前版本号与所述第一版本号是否匹配；若匹配，则修改内存数据redis中所述业务数据的当前值为所述更新值，根据所述第一版本号更新所述当前版本号。

[0010] 一种实施例中，所述根据所述第一版本号更新所述当前版本号之后，还包括：

[0011] 经过内存数据redis预设的缓存时间后，将所述业务数据的当前值及当前版本号写入持久化数据库中。

[0012] 另一种实施例中，所述内存数据redis中业务数据的数据结构通过lua脚本定义，数据结构的前4个字节为业务数据的版本号，后N个字节为业务数据的值，N为预设正整数。

[0013] 进一步的，所述判断所述当前版本号与所述第一版本号是否匹配之后，还包括：

[0014] 若不匹配，则向业务数据更新请求方返回业务数据更新失败的提示信息。

- [0015] 具体的,所述根据所述第一版本号更新所述当前版本号,包括:
- [0016] 将所述第一版本号加1作为所述当前版本号。
- [0017] 第二方面,本发明提供一种数据更新装置,包括:
- [0018] 接收模块:用于接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号;
- [0019] 获取模块:用于基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号;
- [0020] 第一更新模块:用于若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;
- [0021] 第二更新模块:用于若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。
- [0022] 一种实施例中,所述装置还包括:
- [0023] 写入模块:用于经过内存数据redis预设的缓存时间后,将所述业务数据的当前值及当前版本号写入持久化数据库中。
- [0024] 另一种实施例中,所述内存数据redis中业务数据的数据结构通过lua脚本定义,数据结构的前4个字节为业务数据的版本号,后N个字节为业务数据的值,N为预设正整数。
- [0025] 进一步的,所述装置还包括:
- [0026] 提示模块:用于在判断所述当前版本号与所述第一版本号是否匹配之后,若不匹配,则向业务数据更新请求方返回业务数据更新失败的提示信息。
- [0027] 具体的,所述第二更新模块根据所述第一版本号更新所述当前版本号,包括:
- [0028] 将所述第一版本号加1作为所述当前版本号。
- [0029] 第三方面,本发明还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上存储有计算机程序,该程序被处理器执行时实现上述的数据更新方法。
- [0030] 第四方面,本发明还提供了一种服务器,所述服务器包括一个或多个处理器、存储器、一个或多个应用程序,其中所述一个或多个应用程序被存储在所述存储器中并被配置为由所述一个或多个处理器执行,所述一个或多个应用程序配置用于执行上述的数据更新方法。
- [0031] 与现有技术相比,本发明具备如下优点:
- [0032] 本发明提供一种基于内存数据redis和lua插件来实现一个无需固定格式的高效的带版本的业务数据更新功能,在接收业务数据更新请求时获取业务数据更新请求携带的业务数据的更新值及第一版本号,再基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号,然后根据数据的当前值及当前版本号的不同状态更新业务数据的值及版本号,所有操作均在内存redis中执行,没有额外的存储开销,更新操作速度快,避免由于对数据库的操作是磁盘操作而耗时较高、需要增加额外字段对其进行检索和存储消耗较大的问题,提高在并发环境下对数据的更新的性能及效率。
- [0033] 显然,上述有关本发明优点的描述是概括性的,更多的优点描述将体现在后续的实施例揭示中,以及,本领域技术人员也可以本发明所揭示的内容合理地发现本发明的其

他诸多优点。

[0034] 本发明附加的方面和优点将在下面的描述中部分给出,这些将从下面的描述中变得明显,或通过本发明的实践了解到。

附图说明

[0035] 本发明上述的和/或附加的方面和优点从下面结合附图对实施例的描述中将变得明显和容易理解,其中:

[0036] 图1为本发明数据更新方法的一实施例流程示意图;

[0037] 图2为本发明数据更新装置的一实施例示意图;

[0038] 图3为本发明服务器的一实施例结构示意图。

具体实施方式

[0039] 下面详细描述本发明的实施例,所述实施例的示例在附图中示出,其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的,仅用于解释本发明,而不能解释为对本发明的限制。

[0040] 本技术领域技术人员可以理解,除非特意声明,这里使用的单数形式“一”、“一个”、“所述”和“该”也可包括复数形式。应该进一步理解的是,本发明的说明书中使用的措辞“包括”是指存在所述特征、整数、步骤、操作,但是并不排除存在或添加一个或多个其他特征、整数、步骤、操作。

[0041] 本技术领域技术人员可以理解,除非另外定义,这里使用的所有术语(包括技术术语和科学术语),具有与本发明所属领域中的普通技术人员的一般理解相同的意义。还应该理解的是,诸如通用字典中定义的那些术语,应该被理解为具有与现有技术的上下文中的意义一致的意义,并且除非像这里一样被特定定义,否则不会用理想化或过于正式的含义来解释。

[0042] 在一种实施例中,本发明提供一种数据更新方法,如图1所示,包括:

[0043] S100:接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号。

[0044] 本实施例可以应用在购物网站的下单,秒杀等应用场景,商品库存随着用户下单而需要作相应地调整,在该些应用场景下,用户的每一次下单行为都会触发业务数据更新请求,服务器端存储了业务数据如商品的库存量,然后接收所述业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号,以让服务器端根据所述业务数据更新请求对服务器端的业务数据进行更新。

[0045] S200:基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号。

[0046] 本实施例中,服务器端将业务数据存储在内存数据redis中,服务器端接收到所述业务数据更新请求后,基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号,在服务器端,业务数据的版本号以version表征,业务数据的当前值以value表示。

[0047] S300:若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始

化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;

[0048] S310:若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。

[0049] 服务器端在对业务数据进行更新之前,尤其是在高并发场景下对业务数据进行更新之前,需要根据业务数据的当前值及当前版本号判断对所述业务数据更新请求采取怎样的数据更新操作。

[0050] 一种实现逻辑是,若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值。例如,若业务数据的当前值不存在,即只定义了业务数据的名称,而没有定义业务数据的当前值,如只定义了“string len”,而没有定义“string len=1”,此时业务数据len 的当前值不存在,当有针对业务数据len的更新请求进来时,判断业务数据len的当前版本号是否为第一预设值如0,当为0时,则初始化该业务数据,将业务数据的当前值初始化为业务数据更新请求中业务数据的更新值,将业务数据的当前版本号初始化为第二预设值如1。而在此步骤中,若业务数据的当前值不存在且所述当前版本号不为第一预设值如不为0,则返回数据更新失败的信息。此步骤应用在对业务数据值作第一次处理的情况。

[0051] 另一种实现逻辑是,当业务数据初始化完成后,所述业务数据的当前值存在,则判断所述当前版本号与所述第一版本号是否匹配,若匹配,则表示此时的业务数据更新请求合法、有效,服务器端修改内存数据redis 中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。

[0052] 本实施例提供一种基于内存数据redis和lua插件来实现一个无需固定格式的高效的带版本的业务数据更新功能,在接收业务数据更新请求时获取业务数据更新请求携带的业务数据的更新值及第一版本号,再基于lua 脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号,然后根据数据的当前值及当前版本号的不同状态更新业务数据的值及版本号,所有操作均在内存redis中执行,没有额外的存储开销,更新操作速度快,避免由于对数据库的操作是磁盘操作而耗时较高、需要增加额外字段对其进行检索和存储消耗较大的问题,提高在并发环境下对数据的更新的性能及效率。

[0053] 本发明的一种实施例,所述根据所述第一版本号更新所述当前版本号之后,还包括:

[0054] 经过内存数据redis预设的缓存时间后,将所述业务数据的当前值及当前版本号写入持久化数据库中。

[0055] 上述实施例均通过内存数据redis对业务数据进行更新,使用redis可以有效减少对数据库的操作,避免由于对数据库的操作是磁盘操作而耗时较高的问题,在内存数据redis对业务数据进行更新后的预设的缓存时间,再将所述业务数据的当前值及当前版本号写入到持久化数据库,完成业务数据的保存。

[0056] 优选的,本发明实施例中,所述内存数据redis中业务数据的数据结构通过lua脚本定义,数据结构的前4个字节为业务数据的版本号,后N 个字节为业务数据的值,N为预设正整数。

[0057] 本实施例中,使用lua脚本来定义业务数据的数据结构,数据结构的前4个字节为业务数据的版本号,所述业务数据的当前版本号、第一版本号均为业务数据的版本号的概念范畴,后N个字节为业务数据的值,所述业务数据当前值,更新时均为业务数据的值的概念范畴。

[0058] 进一步的,本发明的一种实施例中,所述判断所述当前版本号与所述第一版本号是否匹配之后,还包括:

[0059] 若不匹配,则向业务数据更新请求方返回业务数据更新失败的提示信息。

[0060] 服务器端在对业务数据更新之前,若业务更新请求的所述第一版本号与业务数据的当前版本号不匹配,则表示此时的业务数据更新请求无效,可能是当前业务数据正在更新而无法再接受其它更新请求,然后向业务数据更新请求方返回业务数据更新失败的提示信息。

[0061] 本发明的实施例中,所述根据所述第一版本号更新所述当前版本号,包括:

[0062] 将所述第一版本号加1作为所述当前版本号。

[0063] 在根据第一版本号更新业务数据的当前版本号时,将所述第一版本号加1作为所述当前版本号。

[0064] 例如,接收第一次业务数据更新请求,如果业务数据不存在则初始化为业务数据的当前版本号version为1,当前值value为传递进来的value;以后的每次业务数据更新请求时,unpack解码的时候获取业务数据的当前版本号version,并和业务数据更新请求传递进来的第一版本号 expectVersion对比,如果不相同则不能更新业务数据,如果相同则更新业务数据的当前版本号为expectVersion+1并保存当前值为传递进来的 value。

[0065] 在另一种实施例中,本发明提供一种数据更新装置,如图2所示,包括:

[0066] 接收模块100:用于接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号;

[0067] 获取模块200:用于基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号;

[0068] 第一更新模块300:用于若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;

[0069] 第二更新模块310:用于若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。

[0070] 本发明的一种实施例,所述装置还包括:

[0071] 写入模块:用于经过内存数据redis预设的缓存时间后,将所述业务数据的当前值及当前版本号写入持久化数据库中。

[0072] 本发明的数量中,所述内存数据redis中业务数据的数据结构通过lua 脚本定义,数据结构的前4个字节为业务数据的版本号,后N个字节为业务数据的值,N为预设正整数。

[0073] 本发明的一种实施例,所述装置还包括:

[0074] 提示模块:用于在判断所述当前版本号与所述第一版本号是否匹配之后,若不匹配,则向业务数据更新请求方返回业务数据更新失败的提示信息。

[0075] 本发明的一种实施例中,所述第二更新模块根据所述第一版本号更新所述当前版本号,包括:

[0076] 将所述第一版本号加1作为所述当前版本号。

[0077] 在另一种实施例中,本发明提供了一种计算机可读存储介质,计算机可读存储介质上存储有计算机程序,该程序被处理器执行时实现上述实施例所述的视频处理方法。其中,所述计算机可读存储介质包括但不限于任何类型的盘(包括软盘、硬盘、光盘、CD-ROM、和磁光盘)、ROM(Read-Only Memory,只读存储器)、RAM(Random Access Memory,随机存取存储器)、EPROM(Erasable Programmable Read-Only Memory,可擦写可编程只读存储器)、EEPROM(Electrically Erasable Programmable Read-Only Memory,电可擦可编程只读存储器)、闪存、磁性卡片或光线卡片。也就是,存储设备包括由设备(例如,计算机、手机)以能够读的形式存储或传输信息的任何介质,可以是只读存储器,磁盘或光盘等。

[0078] 本发明实施例提供的一种计算机可读存储介质,可实现接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号;基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号;若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。通过提供一种基于内存数据redis和lua插件来实现一个无需固定格式的高效的带版本的业务数据更新功能,在接收业务数据更新请求时获取业务数据更新请求携带的业务数据的更新值及第一版本号,再基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号,然后根据数据的当前值及当前版本号的不同状态更新业务数据的值及版本号,所有操作均在内存redis中执行,没有额外的存储开销,更新操作速度快,避免由于对数据库的操作是磁盘操作而耗时较高、需要增加额外字段对其进行检索和存储消耗较大的问题,提高在并发环境下对数据的更新的性能及效率。

[0079] 本发明实施例提供的计算机可读存储介质可以实现上述数据更新方法的实施例,具体功能实现请参见方法实施例中的说明,在此不再赘述。

[0080] 此外,在又一种实施例中,本发明还提供一种服务器,如图3所示,所述服务器处理器303、存储器305、输入单元307以及显示单元309等器件。本领域技术人员可以理解,图3示出的结构器件并不构成对所有服务器的限定,可以包括比图示更多或更少的部件,或者组合某些部件。存储器305可用于存储应用程序301以及各功能模块,处理器303运行存储在存储器305的应用程序301,从而执行设备的各种功能应用以及数据处理。存储器305可以是内存储器或外存储器,或者包括内存储器 and 外存储器两者。内存储器可以包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦写可编程ROM(EEPROM)、快闪存储器、或者随机存储器。外存储器可以包括硬盘、软盘、ZIP盘、U盘、磁带等。本发明所公开的存储器包括但不限于这些类型的存储器。本发明所公开的存储器305只作为例子而非作为限定。

[0081] 输入单元307用于接收信号的输入,以及接收用户输入的关键字。输入单元307可包括触控面板以及其它输入设备。触控面板可收集用户在其上或附近的触摸操作(比如用

户使用手指、触笔等任何适合的物体或附件在触控面板上或在触控面板附近的操作),并根据预先设定的程序驱动相应的连接装置;其它输入设备可以包括但不限于物理键盘、功能键(比如播放控制按键、开关按键等)、轨迹球、鼠标、操作杆等中的一种或多种。显示单元309可用于显示用户输入的信息或提供给用户的信息以及计算机设备的各种菜单。显示单元309可采用液晶显示器、有机发光二极管等形式。处理器303是计算机设备的控制中心,利用各种接口和线路连接整个电脑的各个部分,通过运行或执行存储在存储器303内的软件程序和/或模块,以及调用存储在存储器内的数据,执行各种功能和处理数据。图3中所示的一个或多个处理器303能够执行、实现图2中所示的接收模块100、获取模块200、第一更新模块300、第二更新模块310的功能。

[0082] 在一种实施方式中,所述服务器包括一个或多个处理器303,以及一个或多个存储器305,一个或多个应用程序301,其中所述一个或多个应用程序301被存储在存储器305中并被配置为由所述一个或多个处理器303执行,所述一个或多个应用程序301配置用于执行以上实施例所述的数据更新方法。

[0083] 本发明实施例提供的一种服务器,可实现接收业务数据更新请求,所述业务数据更新请求携带有业务数据的更新值及第一版本号;基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号;若所述业务数据的当前值不存在且所述当前版本号为第一预设值,则初始化内存数据redis中所述业务数据的当前值为所述更新值,初始化所述当前版本号为第二预设值;若所述业务数据的当前值存在,判断所述当前版本号与所述第一版本号是否匹配;若匹配,则修改内存数据redis中所述业务数据的当前值为所述更新值,根据所述第一版本号更新所述当前版本号。通过提供一种基于内存数据redis和lua插件来实现一个无需固定格式的高效的带版本的业务数据更新功能,在接收业务数据更新请求时获取业务数据更新请求携带的业务数据的更新值及第一版本号,再基于lua脚本从内存数据redis中获取所述更新请求对应的业务数据的当前值及当前版本号,然后根据数据的当前值及当前版本号的不同状态更新业务数据的值及版本号,所有操作均在内存redis中执行,没有额外的存储开销,更新操作速度快,避免由于对数据库的操作是磁盘操作而耗时较高、需要增加额外字段对其进行检索和存储消耗较大的问题,提高在并发环境下对数据的更新的性能及效率。

[0084] 本发明实施例提供的服务器可以实现上述提供的数据更新方法的实施例,具体功能实现请参见方法实施例中的说明,在此不再赘述。

[0085] 以上所述仅是本发明的部分实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

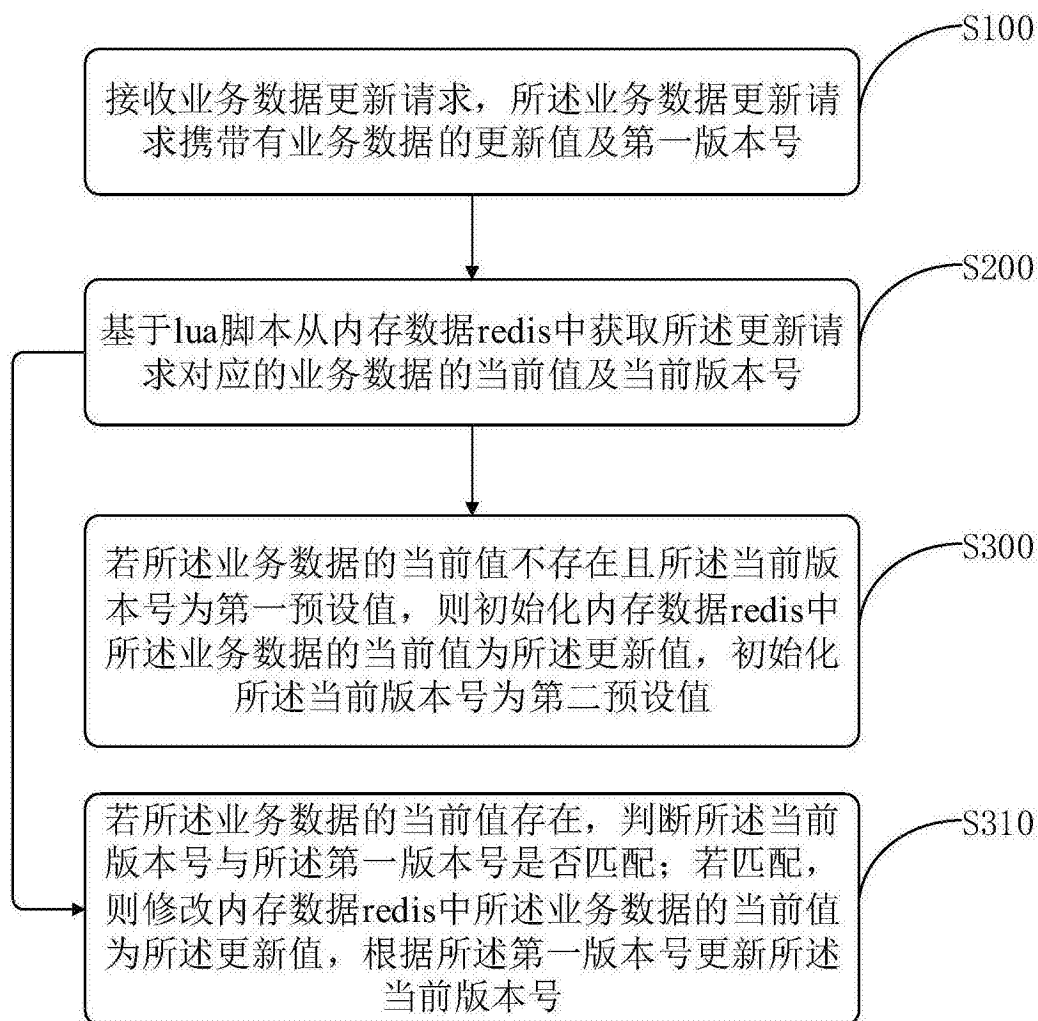


图1

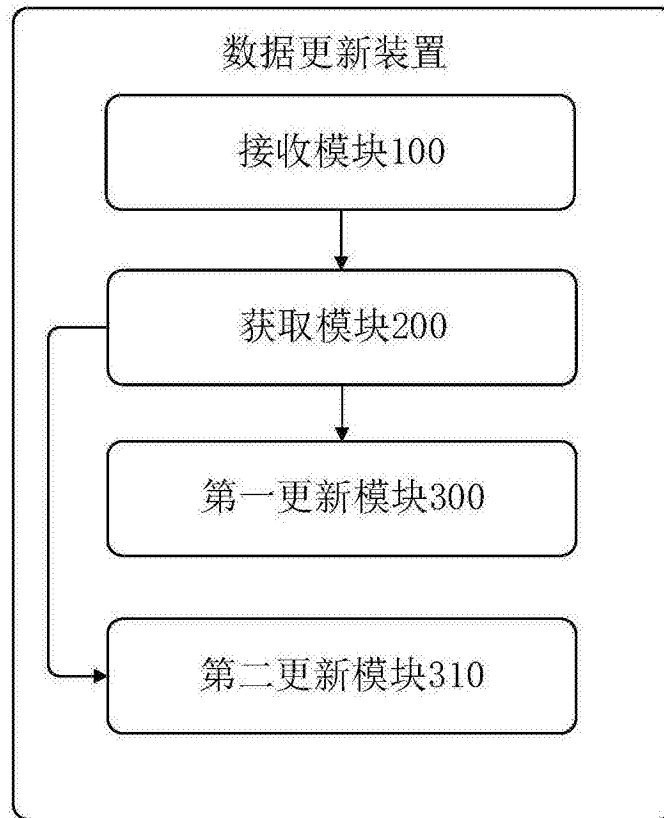


图2

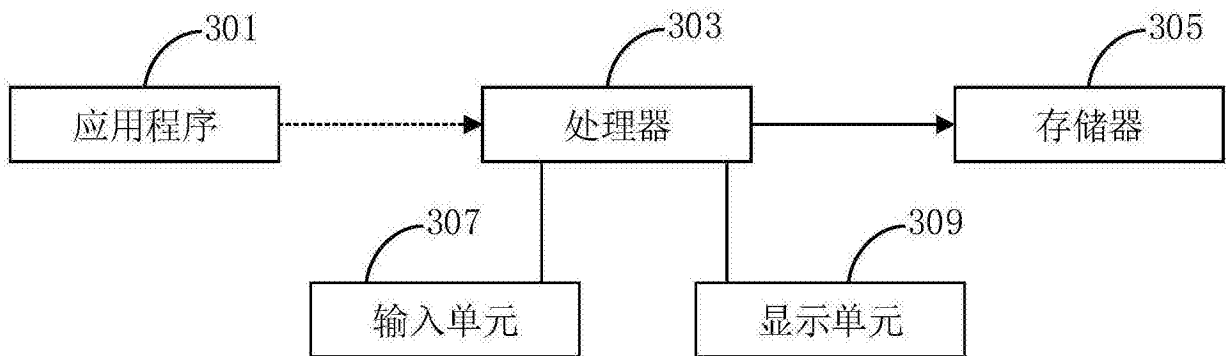


图3