(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112860794 B
(45)授权公告日 2024.08.13

(21)申请号 202110150945.1
(22)申请日 2021.02.03
(65)同一申请的已公布的文献号
申请公布号 CN 112860794 A
(43)申请公布日 2021.05.28
(73)专利权人 百果园技术(新加坡)有限公司
地址 新加坡巴西班让路枫树商业城30号楼15层31A
(72)发明人 唐小龙
(74)专利代理机构 北京泽方誉航专利代理事务所(普通合伙) 11884
专利代理师 陈照辉
(51)Int.Cl.
G06F 16/27(2019.01)
G06F 16/23(2019.01)
G06F 16/21(2019.01)
(56)对比文件
US 6578041 B1,2003.06.10
CN 105740260 A,2016.07.06
审查员 唐文俊

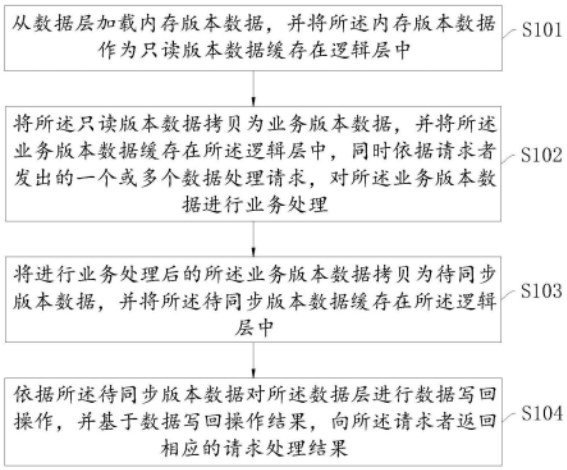
权利要求书2页 说明书9页 附图5页

(54)发明名称

基于缓存的并发能力提升方法、装置、设备及存储介质

(57)摘要

本申请实施例公开了基于缓存的并发能力提升方法、装置、设备及存储介质。本申请实施例提供的技术方案通过将从数据层中加载的内存版本数据作为只读版本数据缓存在逻辑层中,同时将只读版本数据在逻辑层中缓存为业务版本数据,在一个或多个数据处理请求到来时,在逻辑层中对业务版本数据进行业务处理,并将进行业务处理后的业务版本数据拷贝并缓存为待同步版本数据,在成功将待同步版本数据写回数据层时,向请求者返回相应的请求处理结果,实现对高并发请求的低延时响应,并保证了返回结果与数据层数据的一致性,同时有效提升对高并发请求的承载能力。



CN 112860794 B

1.一种基于缓存的并发能力提升方法,其特征在于,包括:

从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中;

将所述只读版本数据拷贝为业务版本数据,并将所述业务版本数据缓存在所述逻辑层中,同时依据请求者发出的一个或多个数据处理请求,对所述业务版本数据进行业务处理;

将进行业务处理后的所述业务版本数据拷贝为待同步版本数据,并将所述待同步版本数据缓存在所述逻辑层中;

确定所述待同步版本数据的待同步版本号和所述内存版本数据的内存版本号,判断所述待同步版本号是否对应所述内存版本号的下一个版本号;若是,则依据所述待同步版本数据对所述数据层进行数据写回操作;若否,则重新从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中;基于数据写回操作结果,向所述请求者返回相应的请求处理结果。

2.根据权利要求1所述的基于缓存的并发能力提升方法,其特征在于,所述从数据层加载内存版本数据,包括:

基于设定的加载时间间隔或基于数据写回操作失败,从数据层加载内存版本数据。

3.根据权利要求1所述的基于缓存的并发能力提升方法,其特征在于,所述将所述只读版本数据拷贝为业务版本数据,并将所述业务版本数据缓存在所述逻辑层中,包括:

将所述只读版本数据拷贝为业务版本数据,并增大所述业务版本数据的业务版本号,同时将所述业务版本数据缓存在所述逻辑层中。

4.根据权利要求1所述的基于缓存的并发能力提升方法,其特征在于,所述将所述待同步版本数据缓存在所述逻辑层中,包括:

将所述待同步版本数据缓存在所述逻辑层中,同时增大所述业务版本数据的业务版本号。

5.根据权利要求1所述的基于缓存的并发能力提升方法,其特征在于,所述将进行业务处理后的所述业务版本数据拷贝为待同步版本数据,包括:

确定逻辑层是否正在对所述数据层进行数据写回操作;

若是,则继续依据数据处理请求对所述业务版本数据进行业务处理;

若否,则将进行业务处理后的所述业务版本数据拷贝为待同步版本数据。

6.根据权利要求1所述的基于缓存的并发能力提升方法,其特征在于,所述向所述请求者返回相应的请求处理结果之后,还包括:

基于数据写回操作成功,依据所述待同步版本数据对所述只读版本数据进行更新,并依据所述业务版本数据对所述待同步版本数据进行更新以进行下一次的数据写回操作。

7.根据权利要求1所述的基于缓存的并发能力提升方法,其特征在于,所述将所述内存版本数据作为只读版本数据缓存在逻辑层中之后,还包括:

基于请求者发出的正确性业务请求,向所述请求者提供所述只读版本数据。

8.一种基于缓存的并发能力提升装置,其特征在于,包括数据加载模块、数据处理模块、数据同步模块和数据写回模块,其中:

所述数据加载模块,用于从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中;

所述数据处理模块,用于将所述只读版本数据拷贝为业务版本数据,并将所述业务版本数据缓存在所述逻辑层中,同时依据请求者发出的一个或多个数据处理请求,对所述业务版本数据进行业务处理;

所述数据同步模块,用于将进行业务处理后的所述业务版本数据拷贝为待同步版本数据,并将所述待同步版本数据缓存在所述逻辑层中;

所述数据写回模块,用于确定所述待同步版本数据的待同步版本号和所述内存版本数据的内存版本号,判断所述待同步版本号是否对应所述内存版本号的下一个版本号;若是,则依据所述待同步版本数据对所述数据层进行数据写回操作;若否,则重新从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中;基于数据写回操作结果,向所述请求者返回相应的请求处理结果。

9.一种基于缓存的并发能力提升设备,其特征在于,包括:存储器以及一个或多个处理器;

所述存储器,用于存储一个或多个程序;

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1-7任一所述的基于缓存的并发能力提升方法。

10.一种包含计算机可执行指令的存储介质,其特征在于,所述计算机可执行指令在由计算机处理器执行时用于执行如权利要求1-7任一所述的基于缓存的并发能力提升方法。

基于缓存的并发能力提升方法、装置、设备及存储介质

技术领域

[0001] 本申请实施例涉及计算机技术领域，尤其涉及基于缓存的并发能力提升方法、装置、设备及存储介质。

背景技术

[0002] 在部分互联网业务场景中，存在某些数据同时被大量用户访问与修改的情况，例如在直播连麦服务中的抢麦场景、电商服务中商品秒杀场景。一般情况下，会针对这些业务场景做数据层与逻辑层的服务分离，在请求到来时会从数据层加载数据到逻辑层，在逻辑层中根据请求对数据进行处理后，再将数据写回数据层，最后才向请求者返回处理结果。

[0003] 上述处理方式在每个请求来到时，都会进行数据的加载、处理与写回，因此一个请求的处理时间会高于逻辑层与数据层的数据传输时延，而下一个请求必须在前一个请求处理完毕才可以继续处理，这时候高并发量的请求与数据读写延时之间的矛盾就会显现出来，导致对高并发请求的处理能力受限。

发明内容

[0004] 本申请实施例提供基于缓存的并发能力提升方法、装置、设备及存储介质，以提升对高并发请求的处理能力。

[0005] 在第一方面，本申请实施例提供了一种基于缓存的并发能力提升方法，包括：

[0006] 从数据层加载内存版本数据，并将所述内存版本数据作为只读版本数据缓存在逻辑层中；

[0007] 将所述只读版本数据拷贝为业务版本数据，并将所述业务版本数据缓存在所述逻辑层中，同时依据请求者发出的一个或多个数据处理请求，对所述业务版本数据进行业务处理；

[0008] 将进行业务处理后的所述业务版本数据拷贝为待同步版本数据，并将所述待同步版本数据缓存在所述逻辑层中；

[0009] 依据所述待同步版本数据对所述数据层进行数据写回操作，并基于数据写回操作结果，向所述请求者返回相应的请求处理结果。

[0010] 在第二方面，本申请实施例提供了一种基于缓存的并发能力提升装置，包括数据加载模块、数据处理模块、数据同步模块和数据写回模块，其中：

[0011] 所述数据加载模块，用于从数据层加载内存版本数据，并将所述内存版本数据作为只读版本数据缓存在逻辑层中；

[0012] 所述数据处理模块，用于将所述只读版本数据拷贝为业务版本数据，并将所述业务版本数据缓存在所述逻辑层中，同时依据请求者发出的一个或多个数据处理请求，对所述业务版本数据进行业务处理；

[0013] 所述数据同步模块，用于将进行业务处理后的所述业务版本数据拷贝为待同步版本数据，并将所述待同步版本数据缓存在所述逻辑层中；

[0014] 所述数据写回模块,用于依据所述待同步版本数据对所述数据层进行数据写回操作,并基于数据写回操作结果,向所述请求者返回相应的请求处理结果。

[0015] 在第三方面,本申请实施例提供了一种基于缓存的并发能力提升设备,包括:存储器以及一个或多个处理器;

[0016] 所述存储器,用于存储一个或多个程序;

[0017] 当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如第一方面所述的基于缓存的并发能力提升方法。

[0018] 在第四方面,本申请实施例提供了一种包含计算机可执行指令的存储介质,所述计算机可执行指令在由计算机处理器执行时用于执行如第一方面所述的基于缓存的并发能力提升方法。

[0019] 本申请实施例通过将从数据层中加载的内存版本数据作为只读版本数据缓存在逻辑层中,同时将只读版本数据在逻辑层中缓存为业务版本数据,在一个或多个数据处理请求到来时,在逻辑层中对业务版本数据进行业务处理,并将进行业务处理后的业务版本数据拷贝并缓存为待同步版本数据,在成功将待同步版本数据写回数据层时,向请求者返回相应的请求处理结果,实现对高并发请求的低延时响应,并保证了返回结果与数据层数据的一致性,同时有效提升对高并发请求的承载能力。

附图说明

[0020] 图1是本申请实施例提供的一种基于缓存的并发能力提升方法的流程图;

[0021] 图2是本申请实施例提供的一种多并发请求处理流程示意图;

[0022] 图3是本申请实施例提供的另一种基于缓存的并发能力提升方法的流程图;

[0023] 图4是本申请实施例提供的一种数据处理请求到来前的数据保存状态示意图;

[0024] 图5是本申请实施例提供的一种第一次数据请求到来时的数据保存状态示意图;

[0025] 图6是本申请实施例提供的一种第二次数据请求到来时的数据保存状态示意图;

[0026] 图7是本申请实施例提供的一种基于缓存的并发能力提升装置的结构示意图;

[0027] 图8是本申请实施例提供的一种基于缓存的并发能力提升设备的结构示意图。

具体实施方式

[0028] 为了使本申请的目的、技术方案和优点更加清楚,下面结合附图对本申请具体实施例作进一步的详细描述。可以理解的是,此处所描述的具体实施例仅仅用于解释本申请,而非对本申请的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本申请相关的部分而非全部内容。在更加详细地讨论示例性实施例之前应当提到的是,一些示例性实施例被描述成作为流程图描绘的处理或方法。虽然流程图将各项操作(或步骤)描述成顺序的处理,但是其中的许多操作可以被并行地、并发地或者同时实施。此外,各项操作的顺序可以被重新安排。当其操作完成时所述处理可以被终止,但是还可以具有未包括在附图中的附加步骤。所述处理可以对应于方法、函数、规程、子例程、子程序等等。

[0029] 图1给出了本申请实施例提供的一种基于缓存的并发能力提升方法的流程图,本申请实施例提供的基于缓存的并发能力提升方法可以由基于缓存的并发能力提升装置来执行,该基于缓存的并发能力提升装置可以通过硬件和/或软件的方式实现,并集成在基于

缓存的并发能力提升设备中。

[0030] 下述以基于缓存的并发能力提升装置执行基于缓存的并发能力提升方法为例进行描述。参考图1,该基于缓存的并发能力提升方法包括:

[0031] S101:从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中。

[0032] 本实施例提供的基于缓存的并发能力提升装置设置有数据层和逻辑层,其中数据层和逻辑层分别用于对数据进行保存和业务处理,本实施例提供的内存版本数据保存在数据层中。

[0033] 示例性的,在需要进行多并发请求的数据处理服务时,根据数据处理服务所对应的数据内容,从数据层中加载相应的内存版本数据,然后将加载得到的内存版本数据作为只读版本数据缓存在逻辑层中。在一个可能的实施例中,还可在开启数据处理服务时预先将内存版本数据加载到逻辑层中并缓存为只读版本数据(例如主播开播时预先将麦位队列数据等加载到逻辑层)。

[0034] 在一个可能的实施例中,本申请实施例在将所述内存版本数据作为只读版本数据缓存在逻辑层中之后,还包括:基于请求者发出的正确性业务请求,向所述请求者提供所述只读版本数据。

[0035] 可以理解接的是,本实施例提供的只读版本数据为不可更改的只读数据,该只读版本数据从内存版本数据拷贝得到,可确保数据的正确性,在接收到请求者发出的有需要保证数据正确性的正确性业务请求时(例如在连麦服务中需要定时推送麦位数据给客户端时),可获取逻辑层中缓存只读版本数据并向对应请求者发送,保证获取数据的正确性。

[0036] S102:将所述只读版本数据拷贝为业务版本数据,并将所述业务版本数据缓存在所述逻辑层中,同时依据请求者发出的一个或多个数据处理请求,对所述业务版本数据进行业务处理。

[0037] 示例性的,在从数据层加载内存版本数据并缓存为只读版本数据后,将只读版本数据拷贝为业务版本数据,并在逻辑层中缓存该业务版本数据。本实施例提供的业务版本数据作为用于进行业务处理的数据,在业务请求到来时,都是对业务版本数据进行业务处理。

[0038] 可选的,在每次从数据层加载内存版本数据并缓存为只读版本数据后,都会重新拷贝新的只读版本数据,并替换原先的业务版本数据,保证进行业务处理的业务版本数据的正确性。

[0039] 进一步的,在接收到请求者发出的一个或多个数据处理请求时,依据这些数据处理请求对业务版本数据进行业务处理及修改。其中,一个或多个数据处理请求可由一个请求者发出,还可以是分别由多个请求者发出。

[0040] 可以理解的是,本申请实施例在接收到多个数据处理请求时,后续的数据处理请求不会被阻塞在逻辑层中,可同时或依次依据这些数据处理请求对业务版本数据进行业务处理,无需等待返回上一个数据处理请求对应的请求处理结果之后才进行下一个数据处理请求的处理,满足对该并发请求的承载能力需求。

[0041] S103:将进行业务处理后的所述业务版本数据拷贝为待同步版本数据,并将所述待同步版本数据缓存在所述逻辑层中。

[0042] 示例性的,在对业务版本数据进行业务处理后,将业务版本数据另外拷贝为待同步版本数据,并在逻辑层中对待同步版本数据进行缓存。

[0043] 其中,本实施例提供的待同步版本数据从进行业务处理后的业务版本数据进行拷贝得到,用于异步写回到数据层中。

[0044] S104:依据所述待同步版本数据对所述数据层进行数据写回操作,并基于数据写回操作结果,向所述请求者返回相应的请求处理结果。

[0045] 示例性的,在拷贝业务版本数据并缓存为待同步版本数据后,依据该待同步版本数据对数据层进行数据写回操作,即将逻辑层中缓存的待同步版本数据写回到数据层,并在数据层中利用待同步版本数据对内存版本数据进行更新。

[0046] 进一步的,在完成数据写回操作的执行后,确定数据写回操作对应的写回操作结果,并根据写回操作结果向各个数据处理请求对应的请求者返回相应的请求处理结果。

[0047] 例如,在写回操作结果指示数据写回操作成功时,向请求者返回的请求处理结果为对业务版本数据进行业务处理对应得到的数据处理结果。而在写回操作结果指示数据写回操作失败时,向请求者返回的请求处理结果为请求处理错误信息(例如服务故障错误码)。

[0048] 对于本实施例提供的任何数据处理请求,都会在相应的待同步版本数据成功写回数据层后,才会向请求者返回相应的数据处理结果,保证内存版本数据与返回的数据处理结果的一致性,减少出现向请求者返回数据修改成功而实际未修改成功的情况。

[0049] 图2给出了本申请实施例提供的一种多并发请求处理流程示意图,其应用于本实施例提供的基于缓存的并发能力提升方法对多并发请求进行处理。参考图2,在需要进行多并发请求的数据服务(例如用户直播多人连麦服务、线上商品秒杀服务等)时,从数据层中加载相应的内存版本数据到逻辑层中,并将加载得到的内存版本数据缓存为只读版本数据。在接收到数据处理请求1和数据处理请求2时,先在逻辑层中依据数据处理请求1和2对业务版本数据进行业务处理,然后将业务处理后的业务版本数据缓存为待同步版本数据,并向数据层写回待同步版本数据。

[0050] 若在数据写回操作过程中接收到数据处理请求3,则在逻辑层中依据数据处理请求3对业务版本数据进行业务处理,在针对待同步版本数据的数据写回操作成功时,向请求者返回数据处理请求1和2对应的数据处理结果。

[0051] 进一步的,利用依据数据处理请求3进行业务处理后的业务版本数据更新待同步版本数据,并向数据层写回该待同步版本数据,在数据写回操作成功时,在向请求者返回数据处理请求3对应的数据处理结果。

[0052] 在上述多并发请求处理流程中,任何请求都会等到数据写回成功才会返回给请求方,因此保证了数据与结果的一致性。同时,若同一时间内有多个请求到来,可以等待所有请求修改完逻辑层的业务版本数据后,以最终修改的业务版本数据作为待同步版本数据写回到数据层。并且在数据写回操作的过程中,逻辑层不会阻塞数据处理请求而是继续提供数据处理服务,从而达到任意一个数据处理请求的延时都不会超过两倍数据层时延(由于进行业务处理的时间相对数据层时延较短,可相对应忽略业务处理造成的时延)的效果。

[0053] 上述,通过将从数据层中加载的内存版本数据作为只读版本数据缓存在逻辑层中,同时将只读版本数据在逻辑层中缓存为业务版本数据,在一个或多个数据处理请求到

来时,在逻辑层中对业务版本数据进行业务处理,并将进行业务处理后的业务版本数据拷贝并缓存为待同步版本数据,在成功将待同步版本数据写回数据层时,向请求者返回相应的请求处理结果,实现对高并发请求的低延时响应,并保证了返回结果与数据层数据的一致性,同时有效提升对高并发请求的承载能力。

[0054] 在上述实施例的基础上,图3给出了本申请实施例提供的另一种基于缓存的并发能力提升方法的流程图,该是对上述基于缓存的并发能力提升方法的具体化。参考图3,该基于缓存的并发能力提升方法包括:

[0055] S201:基于设定的加载时间间隔或基于数据写回操作失败,从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中。

[0056] 具体的,本申请实施例按照设定的加载时间间隔从数据层加载内存版本数据并作为只读版本数据缓存在逻辑层中,尽可能保持只读版本数据的只读版本号与内存版本数据的内存版本号一致。在每次从数据层加载内存版本数据并缓存为只读版本数据时,开始对加载时间间隔的计时,在到达加载时间间隔时,重新从数据层加载内存版本数据,并对逻辑层中的只读版本数据缓存进行更新。

[0057] 进一步的,除了定时加载数据层的内存版本数据之外,本申请实施例还在基于待同步版本数据向数据层进行的数据写回操作失败时,重新从数据层加载内存版本数据,并对逻辑层中的只读版本数据缓存进行更新,保证只读版本数据与内存版本数据一致。

[0058] 可以理解的是,每次基于数据层加载内存版本数据得到的只读版本数据缓存对应的只读版本号与内存版本数据的内存版本号一致,直至内存版本数据被修改。在内存版本数据被修改时,对应的内存版本号做加一处理,若此时对内存版本数据进行修改的是其他服务或进程,此时内存版本号将先进于当前的只读版本号。

[0059] S202:将所述只读版本数据拷贝为业务版本数据,并增大所述业务版本数据的业务版本号,同时将所述业务版本数据缓存在所述逻辑层中。

[0060] 具体的,在每次只读版本数据更新后,将更新后的只读版本数据拷贝为业务版本数据,此时业务版本数据的业务版本号与只读版本数据的只读版本号一致。

[0061] 进一步的,增大业务版本数据的业务版本号,即对业务版本号进行加一处理,并将该业务版本数据缓存在逻辑层中。

[0062] S203:依据请求者发出的一个或多个数据处理请求,对所述业务版本数据进行业务处理。

[0063] S204:确定逻辑层是否正在对所述数据层进行数据写回操作。若是,跳转至步骤S203,否则跳转至步骤S205。

[0064] 在依据数据处理请求完成对业务版本数据的业务处理时,需要将业务版本数据拷贝到待同步版本数据中,等待进行数据写回操作。

[0065] 具体的,在依据数据处理请求完成对业务版本数据的业务处理时,判断逻辑层是否正在对数据层进行数据写回操作。

[0066] 若此时正在进行数据回写操作,需要等待数据回写操作完成后再进行对业务处理数据的拷贝,则跳转至步骤S203,继续依据数据处理请求对所述业务版本数据进行业务处理,即继续依据后续等待处理的数据处理请求对业务版本数据进行业务处理,以免造成对后续数据处理请求的阻塞。

[0067] S205:将进行业务处理后的所述业务版本数据拷贝为待同步版本数据,并将所述待同步版本数据缓存在所述逻辑层中,同时增大所述业务版本数据的业务版本号。

[0068] 若此时没有在进行数据回写操作,则将业务版本数据拷贝为待同步版本数据。可以理解的是,此时待同步版本数据的待同步版本号与业务版本数据的业务版本号一致。

[0069] 进一步的,在逻辑层中对上述从业务版本数据拷贝得到的待同步版本数据进行缓存,并增大业务版本数据的业务版本号,即对业务版本号进行加一处理。可以理解的是,本实施例在每次完成对业务版本数据的拷贝之后,都会增大业务版本数据的业务版本号,保证下次进行业务处理的业务版本数据拷贝为待同步版本数据后,待同步版本号与上一个待同步版本号的差为1,实现与内存版本号的同步。

[0070] S206:判断所述待同步版本号是否对应所述内存版本号的下一个版本号。若是,则跳转至步骤S207,否则重新从数据层加载内存版本数据并跳转至步骤S202。

[0071] 具体的,确定待同步版本数据的待同步版本号和内存版本数据的内存版本号,并将待同步版本号和内存版本号进行比较,在待同步版本号与内存版本号的下一个版本号一致时(此时待同步版本号等于内存版本号加一),跳转至步骤S207。

[0072] 而在待同步版本号与内存版本号的下一个版本号不一致时,确定数据写回操作失败,重新从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中,并跳转至步骤S202。

[0073] 具体的,在待同步版本号不等于内存版本号的下一个版本号时,认为此时内存版本数据已经被其他进程或服务修改(此时只读版本数据的只读版本号落后于内存版本数据的内存版本号),需要重新从数据层加载内存版本数据,并将该内存版本数据作为只读版本数据缓存在逻辑层中,保证只读版本数据与内存版本数据一致,此时只读版本数据的只读版本号与内存版本数据的内存版本号一致。

[0074] S207:依据所述待同步版本数据对所述数据层进行数据写回操作。

[0075] 具体的,在待同步版本号等于内存版本号的下一个版本号时,可确定此时内存版本数据未被其他进程或服务修改,则依据待同步版本数据对数据层进行数据写回操作,实现对内存版本数据的异步更新。可以理解的是,此时由于对内存版本数据进行了修改,内存版本号也进行了加一处理。

[0076] 在一个可能的实施例中,若进行数据写回操作失败,则跳转至步骤S201以重新从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中。若进行数据写回操作成功,则跳转至步骤S208。

[0077] S208:基于数据写回操作结果,向所述请求者返回相应的请求处理结果。

[0078] S209:基于数据写回操作成功,依据所述待同步版本数据对所述只读版本数据进行更新,并依据所述业务版本数据对所述待同步版本数据进行更新以进行下一次的数据写回操作。

[0079] 具体的,在数据写回操作成功时,直接拷贝逻辑层中的待同步版本数据,并依据待同步版本数据对只读版本数据进行更新,由于数据写回操作成功,该待同步版本数据与最新的内存版本数据一致,可保证更新后的只读版本数据与最新的内存版本数据一致,并有效减少数据传输的时延。

[0080] 进一步的,在完成对对只读版本数据的更新后,将只读版本数据拷贝为业务版本

数据进行缓存。进一步的,将业务版本数据拷贝为待同步版本数据,以对待同步版本数据进行更新,为下一次的数据写回操作做准备,并跳转至步骤S203(此时可同步进行下一次的数据写回操作)。

[0081] 图4为本申请实施例提供的一种数据处理请求到来前的数据保存状态示意图。如图4所示,假设最初数据层中保存的内存版本数据的内存版本号V0=x,逻辑层预先根据服务所需要的数据从数据层中加载内存版本数据,并作为只读版本数据进行缓存,此时只读版本号V1=y,并且y小于等于x(初始状态下y=x,在其他进程或服务修改内存版本数据时,y小于x)。然后将只读版本数据拷贝并缓存为业务版本数据,并将业务版本号进行加一,此时业务版本号V2=y+1。此时待同步版本数据的待同步版本号V3=0,此时待同步版本数据为无效数据。

[0082] 图5为本申请实施例提供的一种第一次数据请求到来时的数据保存状态示意图。如图5所示,假设有并发的数据处理请求R1-R8到达,假如在依据数据处理请求R1-R4完成对业务版本数据的修改后,将业务版本数据拷贝并缓存为待同步版本数据进行数据写回操作,并对业务版本号进行加一,此时只读版本号V1=y,待同步版本号V3=y+1,业务版本号V2=y+2,由于此时V3=V0+1(同步版本号对应内存版本号的下一个版本号),可正常进行数据写回操作,并在数据写回操作过程中继续依据数据处理请求R5-R8对业务版本数据进行修改。在数据写回操作成功时,向请求者返回对应请求处理结果,并将待同步版本数据拷贝作为只读版本数据。此时只读版本号V1=x+1,内存版本号V0=x+1。

[0083] 图6为本申请实施例提供的一种第二次数据请求到来时的数据保存状态示意图。如图6所示,假设此时接收到数据处理请求R9,此时依据数据处理请求R5-R8进行业务处理后的业务版本数据被拷贝到待同步版本数据进行数据写回操作,并在数据写回操作过程中继续依据数据处理请求R9对业务版本数据进行修改。此时只读版本号V1=x+1,待同步版本号V3=x+2,业务版本号V2=x+2。假设此时V3=V0+2,则正常进行数据写回操作,并在数据写回操作成功时,向请求者返回对应请求处理结果。假设此时V3<V0+2,则数据写回操作失败,需要让所有已执行的数据处理请求失败,此时数据处理请求R1-R9全部失败,并且强制从数据层重新加载最新的内存版本数据,等待数据加载完毕后才能处理新的数据处理请求。

[0084] 本申请实施例提供的并发能力提升方法可应用于要求具备高并发请求服务能力的应用场景中,例如用户直播连麦服务中,支持单个直播房间内大量用户同时抢麦、加入连麦等待队列等操作,同时可支持更广泛的产品设计,例如弹出通知让房间内的所有用户同时加入连麦等待队列,提升主播与用户连麦的活跃度。又如电商商品秒杀场景,通过在逻辑层缓存商品的数量数据,用户抢购商品的时候,根据数量数据判断内存中的剩余的数量是否满足抢购要求,如果满足则生成抢购任务,异步地串行执行逻辑完成订单。

[0085] 在一个可能的实施例中,还可以是通过限流机制来保证服务的可用性,例如在接收到大量的数据处理请求时,对超过设定数量的数据处理请求进行阻拦,减少大量数据处理请求对服务器造成的冲击。

[0086] 上述,通过将从数据层中加载的内存版本数据作为只读版本数据缓存在逻辑层中,同时将只读版本数据在逻辑层中缓存为业务版本数据,在一个或多个数据处理请求到来时,在逻辑层中对业务版本数据进行业务处理,并将进行业务处理后的业务版本数据拷

贝并缓存为待同步版本数据,在成功将待同步版本数据写回数据层时,向请求者返回相应的请求处理结果,实现对高并发请求的低延时响应,并保证了返回结果与数据层数据的一致性,同时有效提升对高并发请求的承载能力,满足对热点数据的高并发修改。同时,基于比较更新操作(CAS,Compare and Swap),利用单调递增的版本号,只有基于上一个版本号的数据修改才会最终在数据层更新成功,可有效保证更新的正确性。并且在数据写回操作成功时才向请求者返回相应的请求处理结果,减少由于网络或后端节点异常的原因(例如服务器因为断电而丢失来不及写回磁盘的内存数据)导致请求实际失败但是返回成功,而致使数据错乱的情况,有效保证数据的一致性。在用户直播连麦服务场景中,有效提升房间连麦的并发承载能力,提升对大量请求的攻击防护能力,减少由于大量请求的攻击而导致服务器异常或崩溃的情况,并支持高并发的业务场景,例如主播设置自由连麦后,给所有用户发送自由连麦通知,可以响应用户的上麦请求安排用户立即上麦,减少用户等待时长,优化用户体验。

[0087] 图7给出了本申请实施例提供的一种基于缓存的并发能力提升装置的结构示意图。参考图7,该基于缓存的并发能力提升装置包括数据加载模块31、数据处理模块32、数据同步模块33和数据写回模块34。

[0088] 其中,所述数据加载模块31,用于从数据层加载内存版本数据,并将所述内存版本数据作为只读版本数据缓存在逻辑层中;所述数据处理模块32,用于将所述只读版本数据拷贝为业务版本数据,并将所述业务版本数据缓存在所述逻辑层中,同时依据请求者发出的一个或多个数据处理请求,对所述业务版本数据进行业务处理;所述数据同步模块33,用于将进行业务处理后的所述业务版本数据拷贝为待同步版本数据,并将所述待同步版本数据缓存在所述逻辑层中;所述数据写回模块34,用于依据所述待同步版本数据对所述数据层进行数据写回操作,并基于数据写回操作结果,向所述请求者返回相应的请求处理结果。

[0089] 上述,通过将从数据层中加载的内存版本数据作为只读版本数据缓存在逻辑层中,同时将只读版本数据在逻辑层中缓存为业务版本数据,在一个或多个数据处理请求到来时,在逻辑层中对业务版本数据进行业务处理,并将进行业务处理后的业务版本数据拷贝并缓存为待同步版本数据,在成功将待同步版本数据写回数据层时,向请求者返回相应的请求处理结果,实现对高并发请求的低延时响应,并保证了返回结果与数据层数据的一致性,同时有效提升对高并发请求的承载能力。

[0090] 本申请实施例还提供了一种基于缓存的并发能力提升设备,该基于缓存的并发能力提升设备可集成本申请实施例提供的基于缓存的并发能力提升装置。图8是本申请实施例提供的一种基于缓存的并发能力提升设备的结构示意图。参考图8,该基于缓存的并发能力提升设备包括:输入装置43、输出装置44、存储器42以及一个或多个处理器41;所述存储器42,用于存储一个或多个程序;当所述一个或多个程序被所述一个或多个处理器41执行,使得所述一个或多个处理器41实现如上述实施例提供的基于缓存的并发能力提升方法。上述提供的基于缓存的并发能力提升装置、设备和计算机可用于执行上述任意实施例提供的基于缓存的并发能力提升方法,具备相应的功能和有益效果。

[0091] 本申请实施例还提供一种包含计算机可执行指令的存储介质,所述计算机可执行指令在由计算机处理器执行时用于执行如上述实施例提供的基于缓存的并发能力提升方法。当然,本申请实施例所提供的一种包含计算机可执行指令的存储介质,其计算机可执行

指令不限于如上所述的基于缓存的并发能力提升方法,还可以执行本申请任意实施例所提供的基于缓存的并发能力提升方法中的相关操作。上述实施例中提供的基于缓存的并发能力提升装置、设备及存储介质可执行本申请任意实施例所提供的基于缓存的并发能力提升方法,未在上述实施例中详尽描述的技术细节,可参见本申请任意实施例所提供的基于缓存的并发能力提升方法。

[0092] 上述仅为本申请的较佳实施例及所运用的技术原理。本申请不限于这里所述的特定实施例,对本领域技术人员来说能够进行的各种明显变化、重新调整及替代均不会脱离本申请的保护范围。因此,虽然通过以上实施例对本申请进行了较为详细的说明,但是本申请不仅仅限于以上实施例,在不脱离本申请构思的情况下,还可以包括更多其他等效实施例,而本申请的范围由权利要求的范围决定。

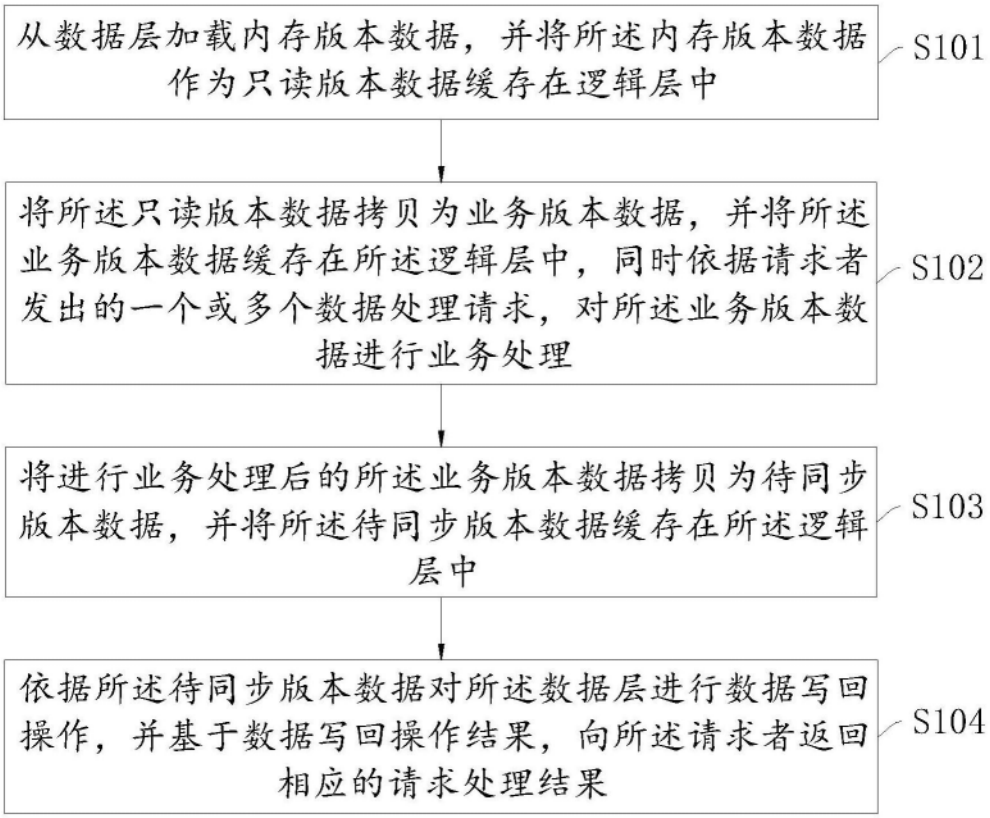

从数据层加载内存版本数据，并将所述内存版本数据作为只读版本数据缓存在逻辑层中
S101
将所述只读版本数据拷贝为业务版本数据，并将所述业务版本数据缓存在所述逻辑层中，同时依据请求者发出的一个或多个数据处理请求，对所述业务版本数据进行业务处理
S102
将进行业务处理后的所述业务版本数据拷贝为待同步版本数据，并将所述待同步版本数据缓存在所述逻辑层中
S103
依据所述待同步版本数据对所述数据层进行数据写回操作，并基于数据写回操作结果，向所述请求者返回相应的请求处理结果
S104

图1

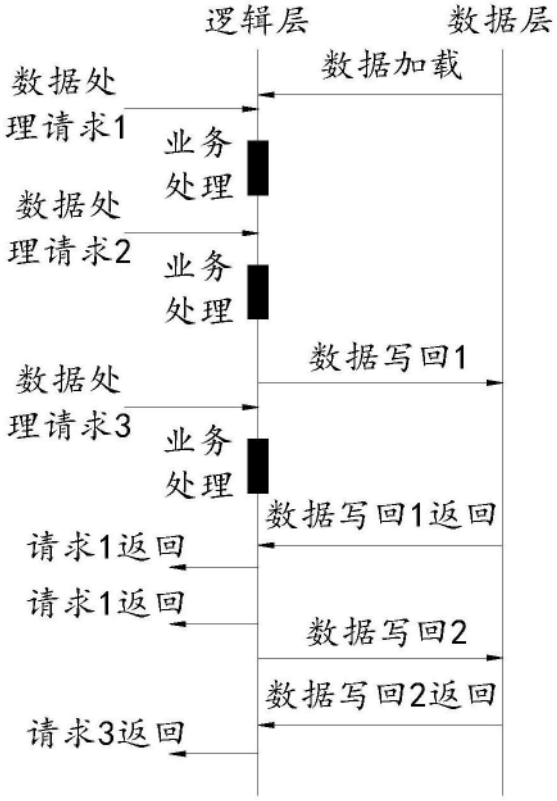

逻辑层
数据层
数据加载
数据处理请求1
业务处理
数据处理请求2
业务处理
数据写回1
数据处理请求3
业务处理
数据写回1返回
请求1返回
请求1返回
数据写回2
数据写回2返回
请求3返回

图2

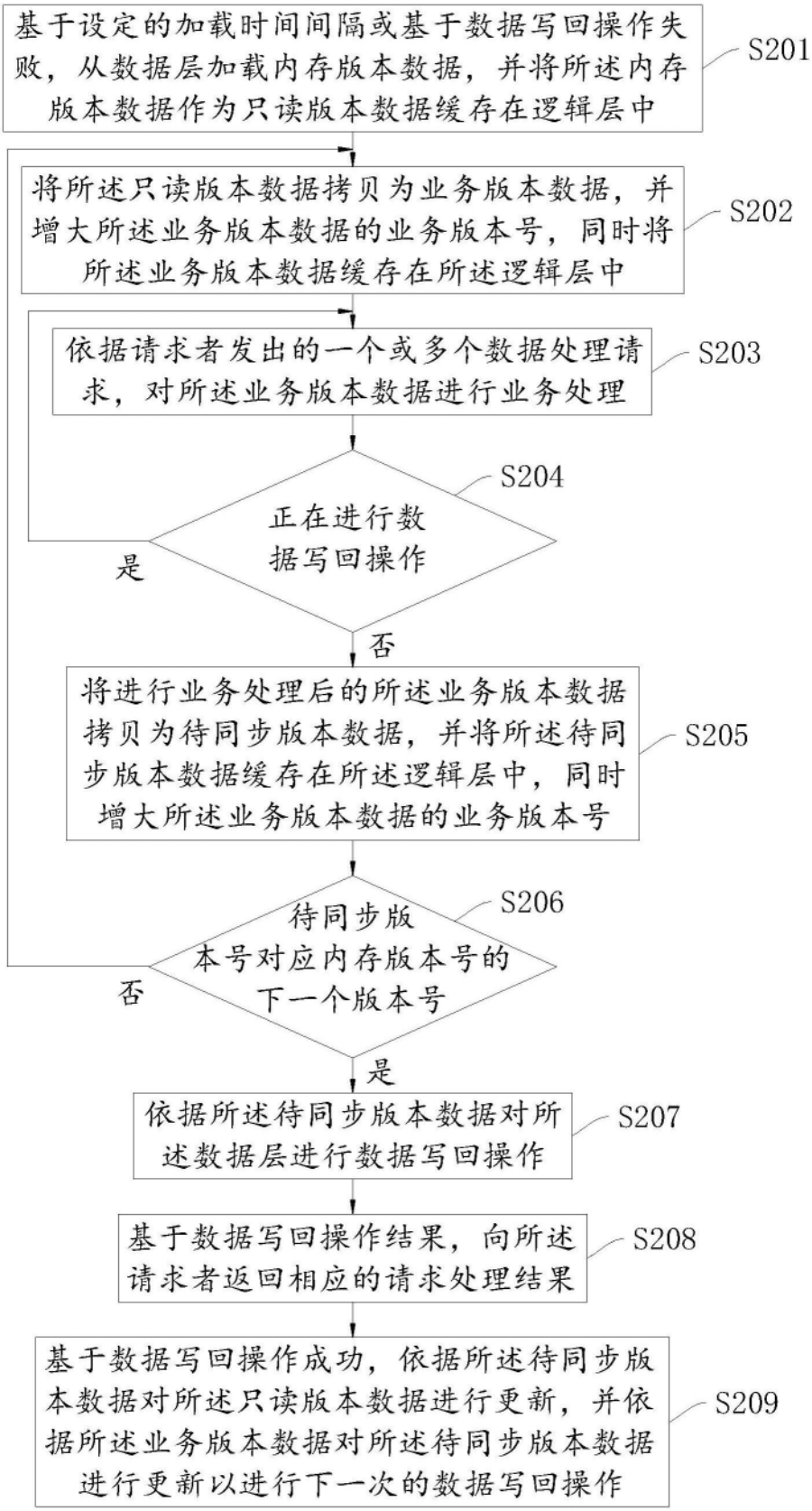

基于设定的加载时间间隔或基于数据写回操作失败，从数据层加载内存版本数据，并将所述内存版本数据作为只读版本数据缓存在逻辑层中
S201
将所述只读版本数据拷贝为业务版本数据，并增大所述业务版本数据的业务版本号，同时将所述业务版本数据缓存在所述逻辑层中
S202
依据请求者发出的一个或多个数据处理请求，对所述业务版本数据进行业务处理
S203
正在进行数据写回操作
S204
是
否
将进行业务处理后的所述业务版本数据拷贝为待同步版本数据，并将所述待同步版本数据缓存在所述逻辑层中，同时增大所述业务版本数据的业务版本号
S205
待同步版本号对应内存版本号的下一个版本号
S206
否
是
依据所述待同步版本数据对所述数据层进行数据写回操作
S207
基于数据写回操作结果，向所述请求者返回相应的请求处理结果
S208
基于数据写回操作成功，依据所述待同步版本数据对所述只读版本数据进行更新，并依据所述业务版本数据对所述待同步版本数据进行更新以进行下一次的数据写回操作
S209

图3


逻辑层
只读版本数据
V1=y（y≤x）
业务版本数据
V2=y+1
只读版本数据
V3=0
加载数据
数据层
内存版本数据 V0=x

图4

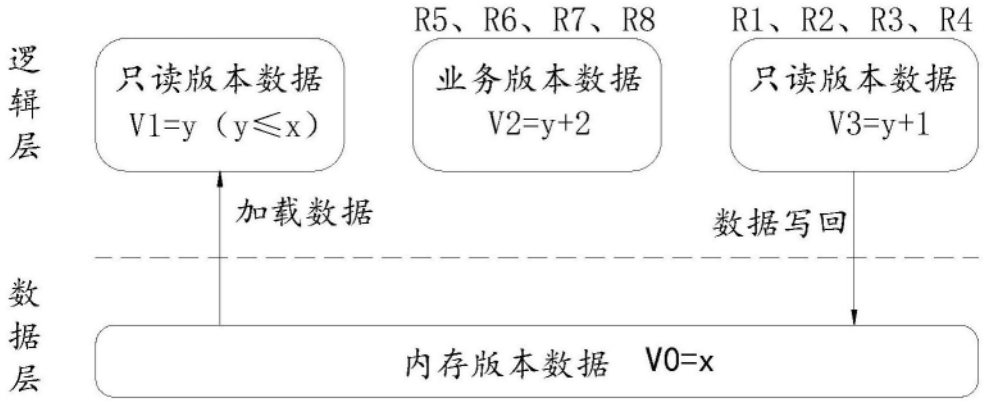

R5、R6、R7、R8
R1、R2、R3、R4
逻辑层
只读版本数据
V1=y（y≤x）
业务版本数据
V2=y+2
只读版本数据
V3=y+1
加载数据
数据写回
数据层
内存版本数据 V0=x

图5

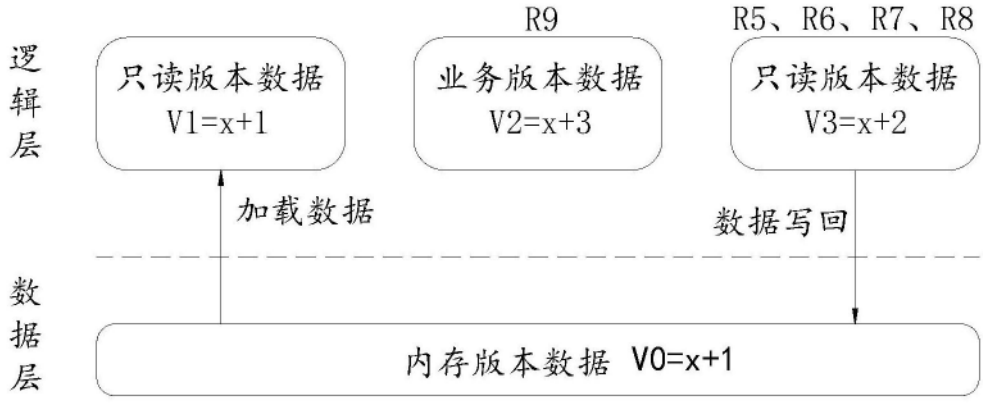

R9
R5、R6、R7、R8
逻辑层
只读版本数据
V1=x+1
业务版本数据
V2=x+3
只读版本数据
V3=x+2
加载数据
数据写回
数据层
内存版本数据 V0=x+1

图6

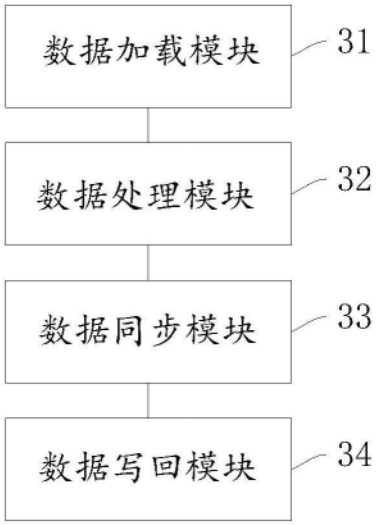

数据加载模块
31
数据处理模块
32
数据同步模块
33
数据写回模块
34

图7

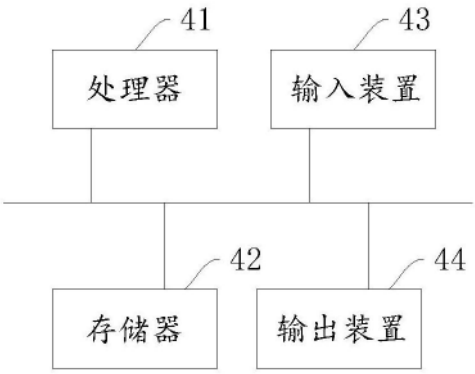

41
处理器
43
输入装置
42
存储器
44
输出装置

图8