



(12) 发明专利申请

(10) 申请公布号 CN 115510016 A

(43) 申请公布日 2022. 12. 23

(21) 申请号 202211295543.1

(22) 申请日 2022.10.21

(71) 申请人 济南浪潮数据技术有限公司

地址 250000 山东省济南市自由贸易试验区济南片区浪潮路1036号浪潮科技园S05楼S311室

(72) 发明人 张亚东 齐鹏鹤 李旭东

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

专利代理师 刘源

(51) Int. Cl.

G06F 16/182 (2019.01)

G06F 16/13 (2019.01)

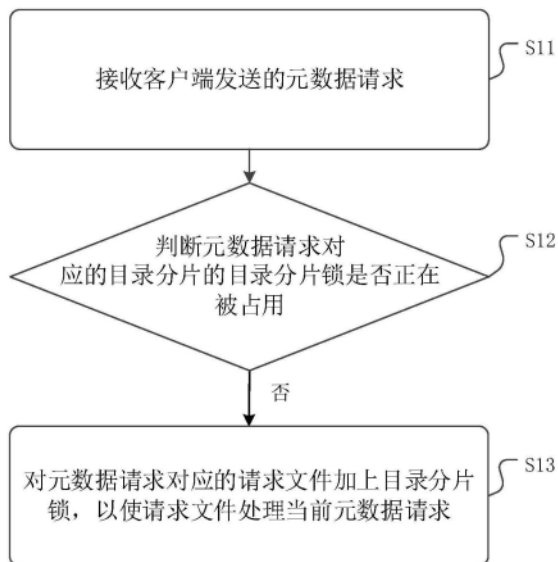
权利要求书2页 说明书8页 附图2页

(54) 发明名称

一种基于目录分片的客户端应答方法、装置及介质

(57) 摘要

本申请公开了一种基于目录分片的客户端应答方法、装置及介质,涉及分布式文件系统领域,为了提高分布式系统的客户端元数据应答处理能力,预先对分布式文件系统中的文件进行分片处理,当接收到客户端发送的元数据请求后,按照其对应的目录分片,判断该目录分片的目录分片锁是否被占用,若没有被占用,则可处理该元数据请求。仅需要判断元数据请求对应的目录分片是否被占用,不需要考虑其他的目录分片,在同一时域内,目录分片之间并行处理,不同时域的目录分片之间也互不影响,时域之间并发处理。客户端按照目录分片数进行并发处理,提高了客户端的元数据应答并发处理能力,同时保证文件的元数据的一致性与有效性,提高分布式系统的并发处理能力。



1. 一种基于目录分片的客户端并发应答方法,其特征在于,包括:
接收客户端发送的元数据请求;
判断所述元数据请求对应的目录分片的目录分片锁是否正在被占用;
若否,对所述元数据请求对应的请求文件加上目录分片锁,以使所述请求文件处理当前所述元数据请求。
2. 根据权利要求1所述的基于目录分片的客户端并发应答方法,其特征在于,若所述元数据请求对应的所述目录分片的所述目录分片锁正在被占用,则还包括:
将所述元数据请求加入等待任务队列,当所述目录分片锁解除占用后,控制所述请求文件处理当前所述元数据请求。
3. 根据权利要求2所述的基于目录分片的客户端并发应答方法,其特征在于,所述接收客户端发送的元数据请求之后,还包括:
判断所述元数据请求是否涉及时域公共变量;
若是,将涉及到的时域公共变量对应的所述目录分片加上目录分片锁。
4. 根据权利要求2所述的基于目录分片的客户端并发应答方法,其特征在于,所述接收客户端发送的元数据请求之后,还包括:
判断所述元数据请求是否涉及修改删除内容;
若是,将当前时域加上时域锁。
5. 根据权利要求3所述的基于目录分片的客户端并发应答方法,其特征在于,所述接收客户端发送的元数据请求之后,还包括:
判断所述元数据请求是否为查询请求;
若是,仅为当前所述请求文件加上所述目录分片锁。
6. 根据权利要求1至5任意一项所述的基于目录分片的客户端并发应答方法,其特征在于,所述接收客户端发送的元数据请求,之前还包括:
将目录文件进行分片,存储至所述目录分片中;
为每一个所述目录分片设计一个目录分片锁,所述目录分片锁之间相互独立。
7. 根据权利要求1所述的基于目录分片的客户端并发应答方法,其特征在于,所述判断所述元数据请求对应的目录分片的目录分片锁是否正在被占用,包括:
根据所述元数据请求对应的索引信息获取其所在的目录分片;
判断所述目录分片的所述目录分片锁是否正在被占用。
8. 一种基于目录分片的客户端并发应答装置,其特征在于,包括:
接收模块,用于接收客户端发送的元数据请求;
判断模块,用于判断所述元数据请求对应的目录分片的目录分片锁是否正在被占用;
若否,触发处理模块;
所述处理模块,用于对所述元数据请求对应的请求文件加上目录分片锁,以使所述请求文件处理当前所述元数据请求。
9. 一种基于目录分片的客户端并发应答装置,其特征在于,包括:
存储器,用于存储计算机程序;
处理器,用于执行所述计算机程序时实现如权利要求1至7任一项所述的基于目录分片的客户端并发应答方法的步骤。

10. 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,所述计算机程序被处理器执行时实现如权利要求1至7任一项所述的基于目录分片的客户端并发应答方法的步骤。

一种基于目录分片的客户端应答方法、装置及介质

技术领域

[0001] 本申请涉及分布式文件系统领域,特别是涉及一种基于目录分片的客户端应答方法、装置及介质。

背景技术

[0002] 计算机通过文件系统管理、存储数据,而信息爆炸时代中人们可以获取的数据成指数倍的增长,单纯通过增加硬盘个数来扩展计算机文件系统的存储容量的方式,在容量大小、容量增长速度、数据备份、数据安全等方面的表现都差强人意。分布式文件系统可以有效解决数据的存储和管理难题:将固定于某个地点的某个文件系统,扩展到任意多个地点/多个文件系统,众多的节点组成一个文件系统网络。每个节点可以分布在不同的地点,通过网络进行节点间的通信和数据传输。人们在使用分布式文件系统时,无需关心数据是存储在哪个节点上、或者是从哪个节点从获取的,只需要像使用本地文件系统一样管理和存储文件系统中的数据。

[0003] 为保证元数据处理的一致性与安全性,客户端应答目前是加有客户端锁进行单线程串行处理,同时增加元数据请求与应答的保序性。但是单线程串行处理使得客户端元数据应答处理并发能力低,导致消息阻塞严重,不利于分布式文件系统的高并发处理。

[0004] 由此可见,提高分布式系统的客户端元数据应答处理能力,是本领域人员亟待解决的技术问题。

发明内容

[0005] 本申请的目的是提供一种提高分布式系统的客户端元数据应答处理能力的客户端并发应答方法、装置及介质。

[0006] 为解决上述技术问题,本申请提供一种基于目录分片的客户端并发应答方法,包括:

[0007] 接收客户端发送的元数据请求;

[0008] 判断元数据请求对应的目录分片的目录分片锁是否正在被占用;

[0009] 若否,对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。

[0010] 优选地,上述的基于目录分片的客户端并发应答方法,若元数据请求对应的目录分片的目录分片锁正在被占用,则还包括:

[0011] 将元数据请求加入等待任务队列,当目录分片锁解除占用后,控制请求文件处理当前元数据请求。

[0012] 优选地,上述基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求之后,还包括:

[0013] 判断元数据请求是否涉及时域公共变量;

[0014] 若是,将涉及到的时域公共变量对应的目录分片加上目录分片锁。

[0015] 优选地,上述基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求之后,还包括:

[0016] 判断元数据请求是否涉及修改删除内容;

[0017] 若是,将当前时域加上时域锁。

[0018] 优选地,上述基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求之后,还包括:

[0019] 判断元数据请求是否为查询请求;

[0020] 若是,仅为当前请求文件加上目录分片锁。

[0021] 优选地,上述基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求,之前还包括:

[0022] 将目录文件进行分片,存储至目录分片中;

[0023] 为每一个目录分片设计一个目录分片锁,目录分片锁之间相互独立。

[0024] 优选地,上述基于目录分片的客户端并发应答方法,判断元数据请求对应的目录分片的目录分片锁是否正在被占用,包括:

[0025] 根据元数据请求对应的索引信息获取其所在的目录分片;

[0026] 判断目录分片的目录分片锁是否正在被占用。

[0027] 为解决上述技术问题,本申请还提供一种基于目录分片的客户端并发应答装置,包括:

[0028] 接收模块,用于接收客户端发送的元数据请求;

[0029] 判断模块,用于判断元数据请求对应的目录分片的目录分片锁是否正在被占用;若否,触发处理模块;

[0030] 处理模块,用于对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。

[0031] 为解决上述技术问题,本申请还提供一种基于目录分片的客户端并发应答装置,包括:

[0032] 存储器,用于存储计算机程序;

[0033] 处理器,用于执行计算机程序时实现上述的基于目录分片的客户端并发应答方法的步骤。

[0034] 为解决上述技术问题,本申请还提供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,计算机程序被处理器执行时实现上述的基于目录分片的客户端并发应答方法的步骤。

[0035] 本申请所提供的基于目录分片的客户端并发应答方法,包括接收客户端发送的元数据请求;判断元数据请求对应的目录分片的目录分片锁是否正在被占用;若否,对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。预先地,对分布式文件系统中的文件进行分片处理,当接收到客户端发送的元数据请求后,按照其对应的目录分片,判断该目录分片的目录分片锁是否被占用,若没有被占用,则可处理该元数据请求。仅需要判断元数据请求对应的目录分片的目录分片锁是否被占用,不需要考虑其他的目录分片是否正在处理其他任务,即,在同一时域内,目录分片之间并行处理,提高了客户端应答的并发性。另外,不同时域的目录分片之间也互不影响,可实现时域之间的完全并

发。客户端按照目录分片数进行并发处理,提高了客户端的元数据应答并发处理能力,同时保证文件的元数据的一致性与有效性,从而提高分布式系统的并发处理能力。

[0036] 另外,本申请还提供一种装置及介质,与上述方法对应,效果同上。

附图说明

[0037] 为了更清楚地说明本申请实施例,下面将对实施例中所需要使用的附图做简单的介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0038] 图1为本申请实施例提供的一种基于目录分片的客户端并发应答方法的流程图;

[0039] 图2为本申请实施例提供的一种基于目录分片的客户端并发应答装置的结构图;

[0040] 图3为本申请实施例提供的另一种基于目录分片的客户端并发应答装置的结构图。

具体实施方式

[0041] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下,所获得的所有其他实施例,都属于本申请保护范围。

[0042] 本申请的核心是提供一种基于目录分片的客户端并发应答方法、装置及介质。

[0043] 为了使本技术领域的人员更好地理解本申请方案,下面结合附图和具体实施方式对本申请作进一步的详细说明。

[0044] 分布式文件系统(Distributed File System,DFS)是指文件系统管理的物理存储资源不一定直接连接在本地节点上,而是通过计算机网络与节点(可简单的理解为一台计算机)相连;或是若干不同的逻辑磁盘分区或卷标组合在一起而形成的完整的有层次的文件系统。DFS为分布在网络上任意位置的资源提供一个逻辑上的树形文件系统结构,从而使用户访问分布在网络上的共享文件更加简便。单独的DFS共享文件夹的作用是相对于通过网络上的其他共享文件夹的访问点。分布式文件系统把大量数据分散到不同的节点上存储,大大减小了数据丢失的风险。分布式文件系统具有冗余性,部分节点的故障并不影响整体的正常运行,而且即使出现故障的计算机存储的数据已经损坏,也可以由其它节点将损坏的数据恢复出来。因此,安全性是分布式文件系统最主要的特征。分布式文件系统通过网络将大量零散的计算机连接在一起,形成一个巨大的计算机集群,使各主机均可以充分发挥其价值。此外,集群之外的计算机只需要经过简单的配置就可以加入到分布式文件系统中,具有极强的可扩展能力。

[0045] 分布式文件系统架构主要面向的集群系统所采用的一套策略方案。主要考虑的有读写性能,数据容灾备份,避免单点故障,以及数据备份,数据容错恢复等。分布式文件系统客户端,是指针对分布式文件系统,需要有配套的接入端来实现用户能访问分布式文件系统,目前主流做法是两个,一个是原生客户端(Native client),另一个是协议网关代理服务。

[0046] 为了保证一个方法或属性在高并发情况下的同一时间只能被同一个线程执行,在

传统单体应用单机部署的情况下,可以使用并发处理相关的功能进行互斥控制。但是,随着业务发展的需要,原单体单机部署的系统被演化成分布式集群系统后,由于分布式系统多线程、多进程并且分布在不同机器上,客户端元数据应答处理并发能力低,消息阻塞严重,不利于分布式文件系统的高并发处理。

[0047] 为解决上述问题,本实施例提供一种基于目录分片的客户端并发应答方法,如图1所示,包括:

[0048] S11:接收客户端发送的元数据请求;

[0049] S12:判断元数据请求对应的目录分片的目录分片锁是否正在被占用;

[0050] 若否,S13:对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。

[0051] 本实施例提到的元数据请求可以是访问、修改、删除等,本实施例不作具体限制,根据实际需要设置即可。

[0052] 需要说明的是,本申请基于C++、操作系统内核(linux)环境、分布式文件系统实现。

[0053] 预先地,对分布式文件系统中的文件进行分片处理,当接收到客户端发送的元数据请求后,按照其对应的目录分片,判断该目录分片的目录分片锁是否被占用,若没有被占用,则可处理该元数据请求。

[0054] 在本实施例中,仅需要判断元数据请求对应的目录分片的目录分片锁是否被占用,不需要考虑其他的目录分片是否正在处理其他任务,即,在同一时域内,目录分片之间并行处理,提高了客户端应答的并发性。另外,不同时域的目录分片之间也互不影响,可实现时域之间的完全并发。

[0055] 本实施例提到的时域指的是Session:在计算机中,尤其是在网络应用中,称为“会话控制”一般都译成时域。在计算机专业术语中,Session是指一个终端用户与交互系统进行通信的时间间隔,通常指从注册进入系统到注销退出系统之间所经过的时间。以及如果需要的话,可能还有一定的操作空间。Session对象存储特定用户会话所需的属性及配置信息。这样,当用户在应用程序的Web页之间跳转时,存储在Session对象中的变量将不会丢失,而是在整个用户会话中一直存在下去。当用户请求来自应用程序的Web页时,如果该用户还没有会话,则全球广域网(World Wide Web,web)服务器将自动创建一个Session对象。当会话过期或被放弃后,服务器将终止该会话。Session对象最常见的一个用法就是存储用户的首选项。需要注意的是,一个Session的概念需要包括特定的客户端,特定的服务器端以及不中断的操作时间。A用户和C服务器建立连接时所处的Session同B用户和C服务器建立连接时所处的Session是两个不同的Session。

[0056] 因此,提高本实施例提供的基于目录分片的客户端并发应答方法,包括接收客户端发送的元数据请求;判断元数据请求对应的目录分片的目录分片锁是否正在被占用;若否,对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。预先地,对分布式文件系统中的文件进行分片处理,当接收到客户端发送的元数据请求后,按照其对应的目录分片,判断该目录分片的目录分片锁是否被占用,若没有被占用,则可处理该元数据请求。仅需要判断元数据请求对应的目录分片的目录分片锁是否被占用,不需要考虑其他的目录分片是否正在处理其他任务,即,在同一时域内,目录分片之间并行处理,

提高了客户端应答的并发性。另外,不同时域的目录分片之间也互不影响,可实现时域之间的完全并发。客户端按照目录分片数进行并发处理,提高了客户端的元数据应答并发处理能力,同时保证文件的元数据的一致性与有效性,从而提高分布式系统的并发处理能力。

[0057] 优选地,基于目录分片的客户端并发应答方法,若元数据请求对应的目录分片的目录分片锁正在被占用,则还包括:

[0058] 将元数据请求加入等待任务队列,当目录分片锁解除占用后,控制请求文件处理当前元数据请求。

[0059] 若该元数据请求对应的目录分片的目录分片锁正在被占用,即该目录分片正在处理其他的元数据请求,则当前的元数据请求需要加入等待任务队列,直到当前目录分片的目录分片锁解除占用后,处理当前元数据请求。也就是说,在同一目录分片内,文件只能串行进行处理,也就是说分片锁相当于分片目录的单行通道,一次只能一个。保证处理任务的有序性,元数据请求与应答的保序性。

[0060] 根据上述实施例,在同一时域的不同目录分片中,可能同时涉及一些公共变量,本实施例提供一种优选方案,基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求之后,还包括:

[0061] 判断元数据请求是否涉及时域公共变量;

[0062] 若是,将涉及到的时域公共变量对应的目录分片加上目录分片锁。

[0063] 当接收到元数据请求后,判断元数据请求是否涉及时域公共变量;若是,将涉及到的时域公共变量对应的目录分片加上目录分片锁。确保当前元数据请求任务处理时,公共变量数据的准确性,元数据的一致性与有效性。

[0064] 根据上述实施例,若元数据请求涉及修改内容或删除内容,本实施例提供一种优选方案,基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求之后,还包括:

[0065] 判断元数据请求是否涉及修改删除内容;

[0066] 若是,将当前时域加上时域锁。

[0067] 当接收到元数据请求后,若当前元数据请求涉及修改内容或删除内容,则需要对当前时域加上时域锁,以确保当前元数据请求任务的处理的元数据的一致性与有效性。

[0068] 根据上述实施例,若元数据请求仅涉及查询内容,不会对元数据的内容造成修改,本实施例提供一种优选方案,基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求之后,还包括:

[0069] 判断元数据请求是否为查询请求;

[0070] 若是,仅为当前请求文件加上目录分片锁。

[0071] 当接收到元数据请求后,判断元数据请求是否为查询请求;若是,仅为当前请求文件加上目录分片锁,以确保当前元数据请求任务的处理的元数据的一致性与有效性。

[0072] 根据上述实施例,基于目录分片的客户端并发应答方法,接收客户端发送的元数据请求,之前还包括:

[0073] 将目录文件进行分片,存储至目录分片中;

[0074] 为每一个目录分片设计一个目录分片锁,目录分片锁之间相互独立。

[0075] 预先对目录文件进行分片,本实施例不限制进行目录分片的依据,可以是数量、内

容,根据实际需要设置即可。

[0076] 根据上述实施例,基于目录分片的客户端并发应答方法,判断元数据请求对应的目录分片的目录分片锁是否正在被占用,包括:

[0077] 根据元数据请求对应的索引信息获取其所在的目录分片;

[0078] 判断目录分片的目录分片锁是否正在被占用。

[0079] 当接收到元数据请求后,根据元数据请求对应的索引信息获取其所在的目录分片;判断目录分片的目录分片锁是否正在被占用。若没有被占用,则可处理该元数据请求。仅需要判断元数据请求对应的目录分片的目录分片锁是否被占用,不需要考虑其他的目录分片是否正在处理其他任务,即,在同一时域内,目录分片之间并行处理,提高了客户端应答的并发性。另外,不同时域的目录分片之间也互不影响,可实现时域之间的完全并发。客户端按照目录分片数进行并发处理,提高了客户端的元数据应答并发处理能力,同时保证文件的元数据的一致性与有效性,从而提高分布式系统的并发处理能力。

[0080] 在上述实施例中,对于基于目录分片的客户端并发应答方法进行了详细描述,本申请还提供基于目录分片的客户端并发应答装置对应的实施例。需要说明的是,本申请从两个角度对装置部分的实施例进行描述,一种是基于功能模块的角度,另一种是基于硬件的角度。

[0081] 基于功能模块的角度,图2为本申请实施例提供的一种基于目录分片的客户端并发应答装置的结构图,如图2所示,包括:

[0082] 接收模块21,用于接收客户端发送的元数据请求;

[0083] 判断模块22,用于判断元数据请求对应的目录分片的目录分片锁是否正在被占用;若否,触发处理模块23;

[0084] 处理模块23,用于对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。

[0085] 具体地,接收模块21,用于接收客户端发送的元数据请求;判断模块22,用于判断元数据请求对应的目录分片的目录分片锁是否正在被占用;若否,触发处理模块23对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。预先地,对分布式文件系统中的文件进行分片处理,当接收到客户端发送的元数据请求后,按照其对应的目录分片,判断该目录分片的目录分片锁是否被占用,若没有被占用,则可处理该元数据请求。仅需要判断元数据请求对应的目录分片的目录分片锁是否被占用,不需要考虑其他的目录分片是否正在处理其他任务,即,在同一时域内,目录分片之间并行处理,提高了客户端应答的并发性。另外,不同时域的目录分片之间也互不影响,可实现时域之间的完全并发。客户端按照目录分片数进行并发处理,提高了客户端的元数据应答并发处理能力,同时保证文件的元数据的一致性与有效性,从而提高分布式系统的并发处理能力。

[0086] 由于装置部分的实施例与方法部分的实施例相互对应,因此装置部分的实施例请参见方法部分的实施例的描述,这里暂不赘述。

[0087] 图3为本申请实施例提供的另一种基于目录分片的客户端并发应答装置的结构图,如图3所示,基于目录分片的客户端并发应答装置包括:存储器30,用于存储计算机程序;

[0088] 处理器31,用于执行计算机程序时实现如上述实施例(基于目录分片的客户端并

发应答方法)获取用户操作习惯信息的方法的步骤。

[0089] 本实施例提供的基于目录分片的客户端并发应答装置可以包括但不限于智能手机、平板电脑、笔记本电脑或台式电脑等。

[0090] 其中,处理器31可以包括一个或多个处理核心,比如4核心处理器、8核心处理器等。处理器31可以采用数字信号处理器(Digital Signal Processor,DSP)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)、可编程逻辑阵列(Programmable Logic Array,PLA)中的至少一种硬件形式来实现。处理器31也可以包括主处理器和协处理器,主处理器是用于对在唤醒状态下的数据进行处理的处理单元,也称中央处理器(Central Processing Unit,CPU);协处理器是用于对在待机状态下的数据进行处理的低功耗处理器。在一些实施例中,处理器31可以在集成有图像处理单元(Graphics Processing Unit,GPU),GPU用于负责显示屏所需要显示的内容的渲染和绘制。一些实施例中,处理器31还可以包括人工智能(Artificial Intelligence,AI)处理器,该AI处理器用于处理有关机器学习的计算操作。

[0091] 存储器30可以包括一个或多个计算机可读存储介质,该计算机可读存储介质可以是非暂态的。存储器30还可包括高速随机存取存储器,以及非易失性存储器,比如一个或多个磁盘存储设备、闪存存储设备。本实施例中,存储器30至少用于存储以下计算机程序301,其中,该计算机程序被处理器31加载并执行之后,能够实现前述任一实施例公开的基于目录分片的客户端并发应答方法的相关步骤。另外,存储器30所存储的资源还可以包括操作系统302和数据303等,存储方式可以是短暂存储或者永久存储。其中,操作系统302可以包括Windows、Unix、Linux等。数据303可以包括但不限于实现基于目录分片的客户端并发应答方法所涉及到的数据等。

[0092] 在一些实施例中,基于目录分片的客户端并发应答装置还可包括有显示屏32、输入输出接口33、通信接口34、电源35以及通信总线36。

[0093] 本领域技术人员可以理解,图3中示出的结构并不构成对基于目录分片的客户端并发应答装置的限定,可以包括比图示更多或更少的组件。

[0094] 本申请实施例提供的基于目录分片的客户端并发应答装置,包括存储器和处理器,处理器在执行存储器存储的程序时,能够实现如下方法:基于目录分片的客户端并发应答方法。包括接收客户端发送的元数据请求;判断元数据请求对应的目录分片的目录分片锁是否正在被占用;若否,对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。预先地,对分布式文件系统中的文件进行分片处理,当接收到客户端发送的元数据请求后,按照其对应的目录分片,判断该目录分片的目录分片锁是否被占用,若没有被占用,则可处理该元数据请求。仅需要判断元数据请求对应的目录分片的目录分片锁是否被占用,不需要考虑其他的目录分片是否正在处理其他任务,即,在同一时域内,目录分片之间并行处理,提高了客户端应答的并发性。另外,不同时域的目录分片之间也互不影响,可实现时域之间的完全并发。客户端按照目录分片数进行并发处理,提高了客户端的元数据应答并发处理能力,同时保证文件的元数据的一致性与有效性,从而提高分布式系统的并发处理能力。

[0095] 最后,本申请还提供一种计算机可读存储介质对应的实施例。计算机可读存储介质上存储有计算机程序,计算机程序被处理器执行时实现如上述基于目录分片的客户端并

发应答方法实施例中记载的步骤。

[0096] 可以理解的是,如果上述实施例中的方法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,执行本申请各个实施例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。

[0097] 本实施例提供的计算机可读存储介质,其上存储有计算机程序,当处理器执行该程序时,可实现以下方法:基于目录分片的客户端并发应答方法,包括接收客户端发送的元数据请求;判断元数据请求对应的目录分片的目录分片锁是否正在被占用;若否,对元数据请求对应的请求文件加上目录分片锁,以使请求文件处理当前元数据请求。预先地,对分布式文件系统中的文件进行分片处理,当接收到客户端发送的元数据请求后,按照其对应的目录分片,判断该目录分片的目录分片锁是否被占用,若没有被占用,则可处理该元数据请求。仅需要判断元数据请求对应的目录分片的目录分片锁是否被占用,不需要考虑其他的目录分片是否正在处理其他任务,即,在同一时域内,目录分片之间并行处理,提高了客户端应答的并发性。另外,不同时域的目录分片之间也互不影响,可实现时域之间的完全并发。客户端按照目录分片数进行并发处理,提高了客户端的元数据应答并发处理能力,同时保证文件的元数据的一致性与有效性,从而提高分布式系统的并发处理能力。

[0098] 以上对本申请所提供的基于目录分片的客户端应答方法、装置及介质进行了详细介绍。说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以对本申请进行若干改进和修饰,这些改进和修饰也落入本申请权利要求的保护范围内。

[0099] 还需要说明的是,在本说明书中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

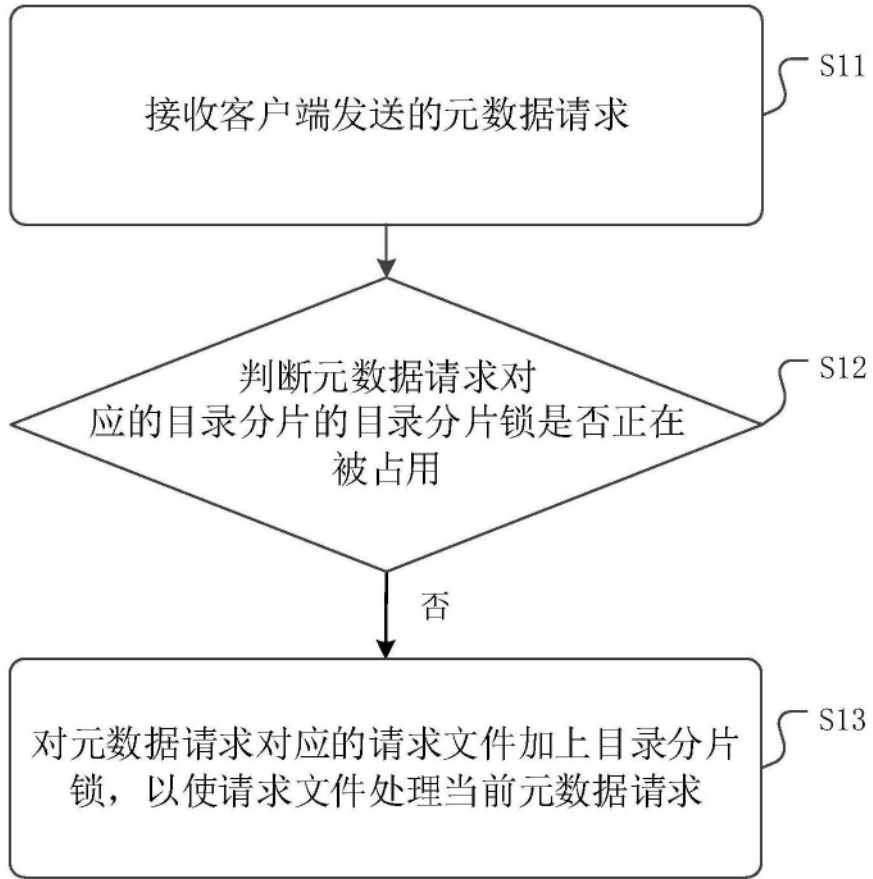


图1

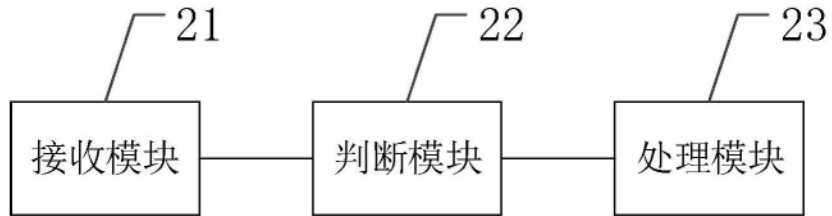


图2

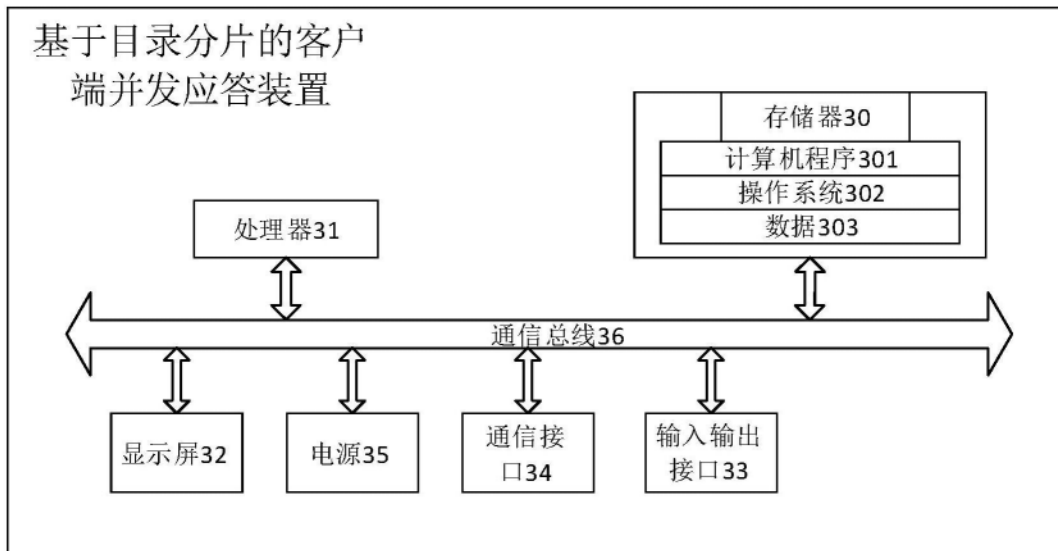


图3