



(12) 发明专利

(10) 授权公告号 CN 113806309 B

(45) 授权公告日 2024. 08. 02

(21) 申请号 202111063485.5

G06F 16/182 (2019.01)

(22) 申请日 2021.09.10

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 111858468 A, 2020.10.30

申请公布号 CN 113806309 A

CN 110941591 A, 2020.03.31

(43) 申请公布日 2021.12.17

审查员 王永贵

(73) 专利权人 济南浪潮数据技术有限公司

地址 250000 山东省济南市中国(山东)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 李伟 刘洪栋 孟祥瑞

(74) 专利代理机构 济南舜源专利事务有限公司 37205

专利代理师 孙玉营

(51) Int. Cl.

G06F 16/16 (2019.01)

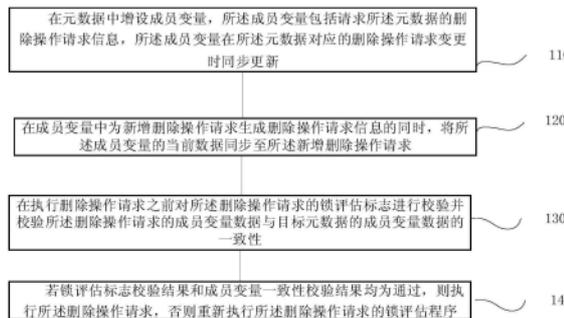
权利要求书2页 说明书9页 附图3页

(54) 发明名称

基于分布式锁的元数据删除方法、系统、终端及存储介质

(57) 摘要

本发明提供一种基于分布式锁的元数据删除方法、系统、终端及存储介质,包括:在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新;在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求;在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性;若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。本发明提高了分布式系统的稳定性。



1. 一种基于分布式锁的元数据删除方法,其特征在于,包括:

在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新;

在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求;

在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性;

若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。

2. 根据权利要求1所述的方法,其特征在于,在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新,包括:

在所述成员变量中记录请求所述元数据的删除操作请求所属节点、删除操作请求名称和删除操作请求按请求时间先后生成的序列号。

3. 根据权利要求2所述的方法,其特征在于,在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新,还包括:

在接收到新的删除操作请求时,将所述删除操作请求的名称和所属节点写入成员变量,通过在成员变量中最后生成的序列号上加1,生成新的删除操作请求的序列号;

将执行完成的删除操作请求在所述成员变量中的相应删除操作请求信息清除,若成员变量中的删除操作请求信息清空,则在接收到新的删除操作请求后重新生成删除操作请求的序列号。

4. 根据权利要求1所述的方法,其特征在于,在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求,包括:

接收到删除操作请求后,提取删除操作请求的目标元数据,并提取所述删除操作请求的信息;

将删除操作请求信息保存至所述目标元数据的成员变量,并将更新后的成员变量数据复制到所述删除操作请求中。

5. 根据权利要求1所述的方法,其特征在于,在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性,包括:

判断所述删除操作请求是否存在锁评估标志,所述锁评估标志为删除操作请求完成对目标元数据的锁评估后生成的标志:

若是,则校验所述删除操作请求的成员变量与目标元数据的成员变量的一致性;

若否,则判定所述删除操作请求未完成锁评估程序。

6. 根据权利要求5所述的方法,其特征在于,若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序,包括:

如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的

成员变量一致,则执行所述删除操作请求;

如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量不一致,则重新执行所述删除操作请求对目标元数据的锁评估程序;

如果删除操作请求不存在锁评估标志,则重新执行所述删除操作请求对目标元数据的锁评估程序。

7.一种基于分布式锁的元数据删除系统,其特征在于,包括:

信息存储单元,用于在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新;

信息同步单元,用于在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求;

请求校验单元,用于在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性;

请求处理单元,用于若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。

8.根据权利要求7所述的系统,其特征在于,所述信息存储单元用于:

在所述成员变量中记录请求所述元数据的删除操作请求所属节点、删除操作请求名称和删除操作请求按请求时间先后生成的序列号。

9.一种终端,其特征在于,包括:

处理器;

用于存储处理器的执行指令的存储器;

其中,所述处理器被配置为执行权利要求1-6任一项所述的方法。

10.一种存储有计算机程序的计算机可读存储介质,其特征在于,该程序被处理器执行时实现如权利要求1-6中任一项所述的方法。

## 基于分布式锁的元数据删除方法、系统、终端及存储介质

### 技术领域

[0001] 本发明涉及分布式存储技术领域,具体涉及一种基于分布式锁的元数据删除方法、系统、终端及存储介质。

### 背景技术

[0002] 目前的分布式文件系统中,MDS(Metadata Server,分布式文件系统中的元数据服务的简称)对元数据的所有操作(包括删除操作)都需要获取到该元数据的分布式锁并通过锁评估。在删除操作中,系统获取到待删除的元数据目录的分布式锁,并通过锁评估后,还需要在分布式锁的保护下进行一些其他操作,如果此时在这些操作中发现操作条件尚不满足,就会发起请求重入操作,等待条件满足时再进行剩余的操作;为了减少不必要的系统资源开销,每个mdr(操作请求)完成锁评估后会获得一个完成锁评估的标志,当该mdr重入删除流程时,如果发现mdr带有该标志,就跳过所评估这一阶段,进行其他的程序流程;

[0003] 但是这里存在一个风险就是如果在删除mdr完成第一次锁评估,且尚未重入时,待删除目录的锁状态如果被其他请求变更了,当删除mdr重入删除流程时,如果只是查看本身的锁评估标志,就会直接跳过锁评估过程,但是此时待删除目录的分布式锁状态可能已经被其他请求改变,导致删除操作需要保护的数据在当前锁状态下无法保证数据一致等,最终导致元数据服务异常,影响整个系统的稳定性。

### 发明内容

[0004] 针对现有技术的上述不足,本发明提供一种基于分布式锁的元数据删除方法、系统、终端及存储介质,以解决上述技术问题。

[0005] 第一方面,本发明提供一种基于分布式锁的元数据删除方法,包括:

[0006] 在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新;

[0007] 在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求;

[0008] 在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性;

[0009] 若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。

[0010] 进一步的,在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新,包括:

[0011] 在所述成员变量中记录请求所述元数据的删除操作请求所属节点、删除操作请求名称和删除操作请求按请求时间先后生成的序列号。

[0012] 进一步的,在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新,还包括:

[0013] 在接收到新的删除操作请求时,将所述删除操作请求的名称和所属节点写入成员变量,通过在成员变量中最后生成的序列号上加1,生成新的删除操作请求的序列号;

[0014] 将执行完成的删除操作请求在所述成员变量中的相应删除操作请求信息清除,若成员变量中的删除操作请求信息清空,则在接收到新的删除操作请求后重新生成删除操作请求的序列号。

[0015] 进一步的,在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求,包括:

[0016] 接收到删除操作请求后,提取删除操作请求的目标元数据,并提取所述删除操作请求的信息;

[0017] 将删除操作请求信息保存至所述目标元数据的成员变量,并将更新后的成员变量数据复制到所述删除操作请求中。

[0018] 进一步的,在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性,包括:

[0019] 判断所述删除操作请求是否存在锁评估标志,所述锁评估标志为删除操作请求完成对目标元数据的锁评估后生成的标志:

[0020] 若是,则校验所述删除操作请求的成员变量与目标元数据的成员变量的一致性;

[0021] 若否,则判定所述删除操作请求未完成锁评估程序。

[0022] 进一步的,若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序,包括:

[0023] 如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量一致,则执行所述删除操作请求;

[0024] 如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量不一致,则重新执行所述删除操作请求对目标元数据的锁评估程序;

[0025] 如果删除操作请求不存在锁评估标志,则重新执行所述删除操作请求对目标元数据的锁评估程序。

[0026] 第二方面,本发明提供一种基于分布式锁的元数据删除系统,包括:

[0027] 信息存储单元,用于在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新;

[0028] 信息同步单元,用于在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求;

[0029] 请求校验单元,用于在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性;

[0030] 请求处理单元,用于若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。

[0031] 进一步的,所述信息存储单元用于:

[0032] 在所述成员变量中记录请求所述元数据的删除操作请求所属节点、删除操作请求名称和删除操作请求按请求时间先后生成的序列号。

[0033] 进一步的,所述信息存储单元用于:

[0034] 在接收到新的删除操作请求时,将所述删除操作请求的名称和所属节点写入成员变量,通过在成员变量中最后生成的序列号上加1,生成新的删除操作请求的序列号;

[0035] 将执行完成的删除操作请求在所述成员变量中的相应删除操作请求信息清除,若成员变量中的删除操作请求信息清空,则在接收到新的删除操作请求后重新生成删除操作请求的序列号。

[0036] 进一步的,所述信息同步单元用于:

[0037] 接收到删除操作请求后,提取删除操作请求的目标元数据,并提取所述删除操作请求的信息;

[0038] 将删除操作请求信息保存至所述目标元数据的成员变量,并将更新后的成员变量数据复制到所述删除操作请求中。

[0039] 进一步的,所述请求校验单元用于:

[0040] 判断所述删除操作请求是否存在锁评估标志,所述锁评估标志为删除操作请求完成对目标元数据的锁评估后生成的标志:

[0041] 若是,则校验所述删除操作请求的成员变量与目标元数据的成员变量的一致性;

[0042] 若否,则判定所述删除操作请求未完成锁评估程序。

[0043] 进一步的,所述请求处理单元用于:

[0044] 如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量一致,则执行所述删除操作请求;

[0045] 如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量不一致,则重新执行所述删除操作请求对目标元数据的锁评估程序;

[0046] 如果删除操作请求不存在锁评估标志,则重新执行所述删除操作请求对目标元数据的锁评估程序。

[0047] 第三方面,提供一种终端,包括:

[0048] 处理器、存储器,其中,

[0049] 该存储器用于存储计算机程序,

[0050] 该处理器用于从存储器中调用并运行该计算机程序,使得终端执行上述的终端的方法。

[0051] 第四方面,提供了一种计算机存储介质,所述计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述各方面所述的方法。

[0052] 本发明的有益效果在于:

[0053] 本发明提供的基于分布式锁的元数据删除方法、系统、终端及存储介质,通过在元数据中增设成员变量,成员变量包括请求所述元数据的删除操作请求信息,并且成员变量在元数据对应的删除操作请求变更时同步更新;在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将成员变量的当前数据同步至所述新增删除操作请求;当删除操作请求重入时,其本身携带锁评估标志,再对锁评估标志校验完成之后,校验其携带的成员变量数据与目标元数据的当前成员变量数据的一致性,若在该删除操作请求重入之前有其他操作请求对目标元数据进行了锁变更,目标元数据的当前成员变量数据会发生变化,因此与该删除操作请求是不一致的,若两者不一致则需要重新对重入的删除操作请求进行锁

评估,若两者仍保持一致则可以直接执行该删除操作请求。本发明能够在目录删除流程中,避免由于MDS分布式锁状态发生变化引起的MDS的异常情况,从而提高整个分布式系统的稳定性。

[0054] 此外,本发明设计原理可靠,结构简单,具有非常广泛的应用前景。

### 附图说明

[0055] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,对于本领域普通技术人员而言,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。

[0056] 图1是本发明一个实施例的方法的示意性流程图。

[0057] 图2是本发明一个实施例的方法的另一示意性流程图。

[0058] 图3是本发明一个实施例的系统的示意性框图。

[0059] 图4为本发明实施例提供的一种终端的结构示意图。

### 具体实施方式

[0060] 为了使本技术领域的人员更好地理解本发明中的技术方案,下面将结合本发明实施例中的附图,对本发明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发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,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,都应当属于本发明保护的范围。

[0061] 下面对本发明中出现的术语进行解释。

[0062] 分布式存储是一种数据存储技术,通过网络使用企业中的每台机器上的磁盘空间,并将这些分散的存储资源构成一个虚拟的存储设备,数据分散的存储在企业的各个角落。分布式存储系统,是将数据分散存储在多台独立的设备上。传统的网络存储系统采用集中的存储服务器存放所有数据,存储服务器成为系统性能的瓶颈,也是可靠性和安全性的焦点,不能满足大规模存储应用的需要。分布式网络存储系统采用可扩展的系统结构,利用多台存储服务器分担存储负荷,利用位置服务器定位存储信息,它不但提高了系统的可靠性、可用性和存取效率,还易于扩展。

[0063] 元数据目录metadata catalog,数据元(Data Element)是电子单证最基本的、不可再分的数据单元。元数据(Metadata),又称中介数据、中继数据,为描述数据的数据(data about data),主要是描述数据属性(property)的信息,用来支持如指示存储位置、历史数据、资源查找、文件记录等功能。元数据算是一种电子式目录,为了达到编制目录的目的,必须在描述并收藏数据的内容或特色,进而达成协助数据检索的目的。都柏林核心集(Dublin Core Metadata Initiative,DCMI)是元数据的一种应用,是1995年2月由国际图书馆电脑中心(OCLC)和美国国家超级计算应用中心(National Center for Supercomputing Applications,NCSA)所联合赞助的研讨会,在邀请52位来自图书馆员、电脑专家,共同制定规格,创建一套描述网络上电子文件之特征。元数据是关于数据的组织、数据域及其关系的信息,简言之,元数据就是关于数据的数据。元数据管理。在大数据环境下,元数据的体量也非常大,元数据的存取性能是整个分布式文件系统性能的关键。常见的元数据管理可以分

为集中式和分布式元数据管理架构。集中式元数据管理架构采用单一的元数据服务器,实现简单,但是存在单点故障等问题。分布式元数据管理架构则将元数据分散在多个结点上。进而解决了元数据服务器的性能瓶颈等问题。并提高了元数据管理架构的可扩展性,但实现较为复杂,并引入了元数据一致性的问题。另外,还有一种无元数据服务器的分布式架构,通过在线算法组织数据,不需要专用的元数据服务器。但是该架构对数据一致性的保障很困难,实现较为复杂。文件目录遍历操作效率低下,并且缺乏文件系统全局监控管理功能。

[0064] 图1是本发明一个实施例的方法的示意性流程图。其中,图1执行主体可以为一种基于分布式锁的元数据删除系统。

[0065] 如图1所示,该方法包括:

[0066] 步骤110,在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新;

[0067] 步骤120,在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求;

[0068] 步骤130,在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性;

[0069] 步骤140,若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。

[0070] 为了便于对本发明的理解,下面以本发明基于分布式锁的元数据删除方法的原理,结合实施例中对基于分布式锁的元数据进行删除的过程,对本发明提供的基于分布式锁的元数据删除方法做进一步的描述。

[0071] 具体的,所述基于分布式锁的元数据删除方法包括:

[0072] S1、在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新。

[0073] 在所述成员变量中记录请求所述元数据的删除操作请求所属节点、删除操作请求名称和删除操作请求按请求时间先后生成的序列号。在接收到新的删除操作请求时,将所述删除操作请求的名称和所属节点写入成员变量,通过在成员变量中最后生成的序列号上加1,生成新的删除操作请求的序列号;将执行完成的删除操作请求在所述成员变量中的相应删除操作请求信息清除,若成员变量中的删除操作请求信息清空,则在接收到新的删除操作请求后重新生成删除操作请求的序列号。

[0074] 当分布式系统中接收到一个删除操作请求时,该删除操作请求用于请求删除目标元数据。判断删除操作请求是否已经存在成员变量数据,初次接收到的删除操作请求不存在成员变量数据,则在目标元数据的成员变量中记录所述删除操作请求的名称和所属节点,此时目标元数据可能已经有多个操作请求了,这些操作请求的信息在目标元数据的成员变量中按请求时间先后排序,例如目标元数据的已存数据为1、请求A信息;2、请求B信息。此时将新的删除操作请求表示为请求C,则更新目标元数据的成员变量为1、请求A信息;2、请求B信息;3、请求C信息。如果目标元数据的成员变量中没有其他请求信息,则直接为请求C生成1、请求C信息。

[0075] S2、在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成

员变量的当前数据同步至所述新增删除操作请求。

[0076] 接收到删除操作请求后,提取删除操作请求的目标元数据,并提取所述删除操作请求的信息;将删除操作请求信息保存至所述目标元数据的成员变量,并将更新后的成员变量数据复制到所述删除操作请求中。

[0077] 将步骤S1中更新的目标元数据的成员变量,即1、请求A信息;2、请求B信息;3、请求C信息,复制一份到请求C,此时请求C也有一份成员变量数据,即1、请求A信息;2、请求B信息;3、请求C信息,请求C存储的成员变量数据是不变的,而目标元数据中的成员变量是随操作请求的执行情况实时更新的。

[0078] 请求C完成对目标元数据的锁评估程序后得到一个锁评估标志,然后去执行其他操作。在此期间如果请求B执行完成,则目标元数据的成员变量数据更新为1、请求A信息;3、请求C信息。

[0079] S3、在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性。

[0080] 判断所述删除操作请求是否存在锁评估标志,所述锁评估标志为删除操作请求完成对目标元数据的锁评估后生成的标志:若是,则校验所述删除操作请求的成员变量与目标元数据的成员变量的一致性;若否,则判定所述删除操作请求未完成锁评估程序。

[0081] 对于重入的删除操作请求,其存在锁评估标志,在此基础上比对其写的的成员变量数据与目标元数据的当前成员变量数据的一致性,若两者不一致则说明有其他操作请求变更过目标元数据的锁状态,需要重新对重入的删除操作请求进行锁评估。

[0082] S4、若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。

[0083] 如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量一致,则执行所述删除操作请求;如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量不一致,则重新执行所述删除操作请求对目标元数据的锁评估程序;

[0084] 如果删除操作请求不存在锁评估标志,则重新执行所述删除操作请求对目标元数据的锁评估程序。同时将该删除操作请求的信息记录至目标元数据的成员变量,即执行步骤S1-S3。

[0085] 请参考图2,以下为本实施例的具体执行流程:

[0086] 1) 在mdr中增加标志A,标记是否有其他mdr操作过对应的目录元数据;

[0087] 增加标志B,标记当前mdr涉及的目录元数据锁状态是否满足要求;

[0088] 增加标志C,标记是否强制重新进行锁评估。

[0089] 2) 在删除流程中,如果发现删除操作已经完成了加锁,本次是重入操作,且操作的是目录,且mdr中inode\_mdr的副本与目录元数据成员变量inode\_mdr的当前数据不一致时,则将1)中的标志A设为true,并进入步骤3),否则进入步骤5)。

[0090] 3) 判断目录元数据的锁状态是否满足目录删除的锁状态要求,如果不满足,则将1)中的标志B设置true,进入步骤4),否则进入步骤5)。

[0091] 4) 将1)中标志C设置为true,清除目录元数据中当前的锁状态。

[0092] 5) 锁评估。如果标志C为true,进入锁评估;如果C为false,且mdr未完成加锁(不是

重入流程或者虽是重入但是加锁未完成),则进入锁评估;如果C为false,且mdr已完成锁评估,跳过锁评估程序,直接执行请求。

[0093] 6) 删除其他操作。

[0094] 如图3所示,该系统300包括:

[0095] 信息存储单元310,用于在元数据中增设成员变量,所述成员变量包括请求所述元数据的删除操作请求信息,所述成员变量在所述元数据对应的删除操作请求变更时同步更新;

[0096] 信息同步单元320,用于在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将所述成员变量的当前数据同步至所述新增删除操作请求;

[0097] 请求校验单元330,用于在执行删除操作请求之前对所述删除操作请求的锁评估标志进行校验并校验所述删除操作请求的成员变量数据与目标元数据的成员变量数据的一致性;

[0098] 请求处理单元340,用于若锁评估标志校验结果和成员变量一致性校验结果均为通过,则执行所述删除操作请求,否则重新执行所述删除操作请求的锁评估程序。

[0099] 可选地,作为本发明一个实施例,所述信息存储单元用于:

[0100] 在所述成员变量中记录请求所述元数据的删除操作请求所属节点、删除操作请求名称和删除操作请求按请求时间先后生成的序列号。

[0101] 可选地,作为本发明一个实施例,所述信息存储单元用于:

[0102] 在接收到新的删除操作请求时,将所述删除操作请求的名称和所属节点写入成员变量,通过在成员变量中最后生成的序列号上加1,生成新的删除操作请求的序列号;

[0103] 将执行完成的删除操作请求在所述成员变量中的相应删除操作请求信息清除,若成员变量中的删除操作请求信息清空,则在接收到新的删除操作请求后重新生成删除操作请求的序列号。

[0104] 可选地,作为本发明一个实施例,所述信息同步单元用于:

[0105] 接收到删除操作请求后,提取删除操作请求的目标元数据,并提取所述删除操作请求的信息;

[0106] 将删除操作请求信息保存至所述目标元数据的成员变量,并将更新后的成员变量数据复制到所述删除操作请求中。

[0107] 可选地,作为本发明一个实施例,所述请求校验单元用于:

[0108] 判断所述删除操作请求是否存在锁评估标志,所述锁评估标志为删除操作请求完成对目标元数据的锁评估后生成的标志:

[0109] 若是,则校验所述删除操作请求的成员变量与目标元数据的成员变量的一致性;

[0110] 若否,则判定所述删除操作请求未完成锁评估程序。

[0111] 可选地,作为本发明一个实施例,所述请求处理单元用于:

[0112] 如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量一致,则执行所述删除操作请求;

[0113] 如果删除操作请求存在锁评估标志,且所述删除操作请求的成员变量与目标元数据的成员变量不一致,则重新执行所述删除操作请求对目标元数据的锁评估程序;

[0114] 如果删除操作请求不存在锁评估标志,则重新执行所述删除操作请求对目标元数

据的锁评估程序。

[0115] 图4为本发明实施例提供的一种终端400的结构示意图,该终端400可以用于执行本发明实施例提供的基于分布式锁的元数据删除方法。

[0116] 其中,该终端400可以包括:处理器410、存储器420及通信单元430。这些组件通过一条或多条总线进行通信,本领域技术人员可以理解,图中示出的服务器的结构并不构成对本发明的限定,它既可以是总线形结构,也可以是星型结构,还可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0117] 其中,该存储器420可以用于存储处理器410的执行指令,存储器420可以由任何类型的易失性或非易失性存储终端或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。当存储器420中的执行指令由处理器410执行时,使得终端400能够执行以下上述方法实施例中的部分或全部步骤。

[0118] 处理器410为存储终端的控制中心,利用各种接口和线路连接整个电子终端的各个部分,通过运行或执行存储在存储器420内的软件程序和/或模块,以及调用存储在存储器内的数据,以执行电子终端的各种功能和/或处理数据。所述处理器可以由集成电路(Integrated Circuit,简称IC)组成,例如可以由单颗封装的IC所组成,也可以由连接多颗相同功能或不同功能的封装IC而组成。举例来说,处理器410可以仅包括中央处理器(Central Processing Unit,简称CPU)。在本发明实施方式中,CPU可以是单运算核心,也可以包括多运算核心。

[0119] 通信单元430,用于建立通信信道,从而使所述存储终端可以与其它终端进行通信。接收其他终端发送的用户数据或者向其他终端发送用户数据。

[0120] 本发明还提供一种计算机存储介质,其中,该计算机存储介质可存储有程序,该程序执行时可包括本发明提供的各实施例中的部分或全部步骤。所述的存储介质可为磁碟、光盘、只读存储记忆体(英文:read-only memory,简称:ROM)或随机存储记忆体(英文:random access memory,简称:RAM)等。

[0121] 因此,本发明通过在元数据中增设成员变量,成员变量包括请求所述元数据的删除操作请求信息,并且成员变量在元数据对应的删除操作请求变更时同步更新;在成员变量中为新增删除操作请求生成删除操作请求信息的同时,将成员变量的当前数据同步至所述新增删除操作请求;当删除操作请求重入时,其本身携带锁评估标志,再对锁评估标志校验完成之后,校验其携带的成员变量数据与目标元数据的当前成员变量数据的一致性,若在该删除操作请求重入之前有其他操作请求对目标元数据进行了锁变更,目标元数据的当前成员变量数据会发生变化,因此与该删除操作请求是不一致的,若两者不一致则需要重新对重入的删除操作请求进行锁评估,若两者仍保持一致则可以直接执行该删除操作请求。本发明能够在目录删除流程中,避免由于MDS分布式锁状态发生变化引起的MDS的异常情况,从而提高整个分布式系统的稳定性,本实施例所能达到的技术效果可以参见上文中的描述,此处不再赘述。

[0122] 本领域的技术人员可以清楚地了解到本发明实施例中的技术可借助软件加必需的通用硬件平台的方式来实现。基于这样的理解,本发明实施例中的技术方案本质上或者说对现有技术做出贡献的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

一个存储介质中如U盘、移动硬盘、只读存储器 (ROM, Read-Only Memory)、随机存取存储器 (RAM, Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质,包括若干指令用以使得一台计算机终端(可以是个人计算机,服务器,或者第二终端、网络终端等)执行本发明各个实施例所述方法的全部或部分步骤。

[0123] 本说明书中各个实施例之间相同相似的部分互相参见即可。尤其,对于终端实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例中的说明即可。

[0124] 在本发明所提供的几个实施例中,应该理解到,所揭露的系统和方法,可以通过其它的方式实现。例如,以上所描述的系统实施例仅仅是示意性的,例如,所述单元的划分,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,系统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。

[0125] 所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。

[0126] 另外,在本发明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。

[0127] 尽管通过参考附图并结合优选实施例的方式对本发明进行了详细描述,但本发明并不限于此。在不脱离本发明的精神和实质的前提下,本领域普通技术人员可以对本发明的实施例进行各种等效的修改或替换,而这些修改或替换都应在本发明的涵盖范围内/任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应所述以权利要求的保护范围为准。

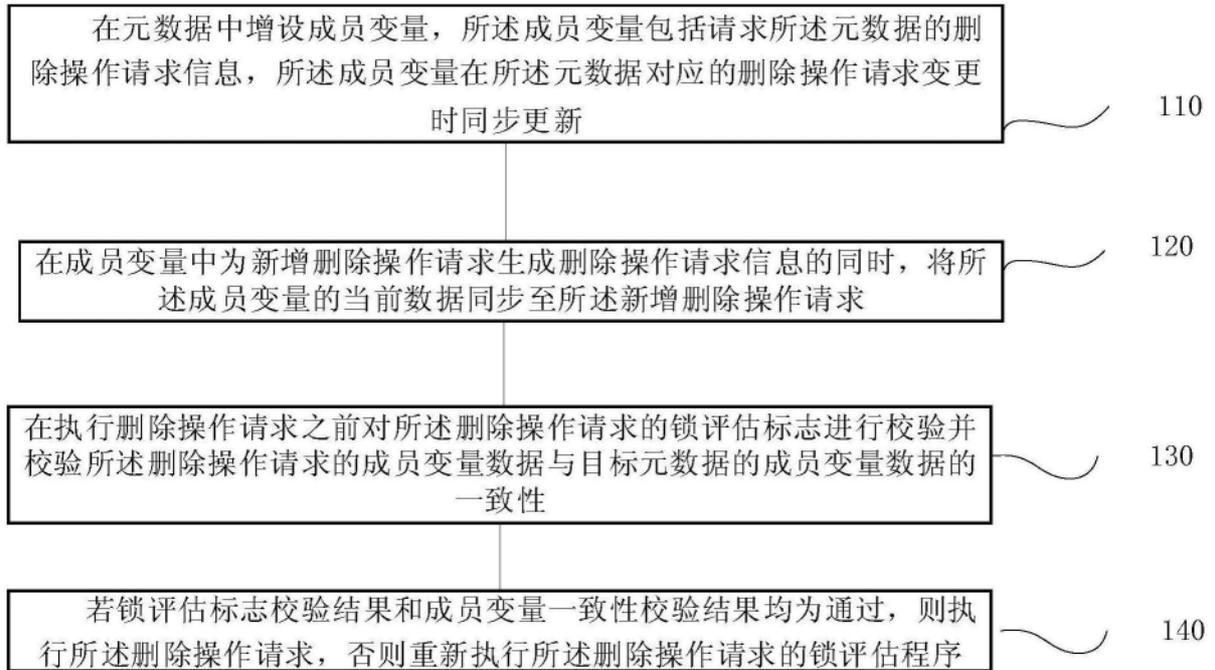


图1

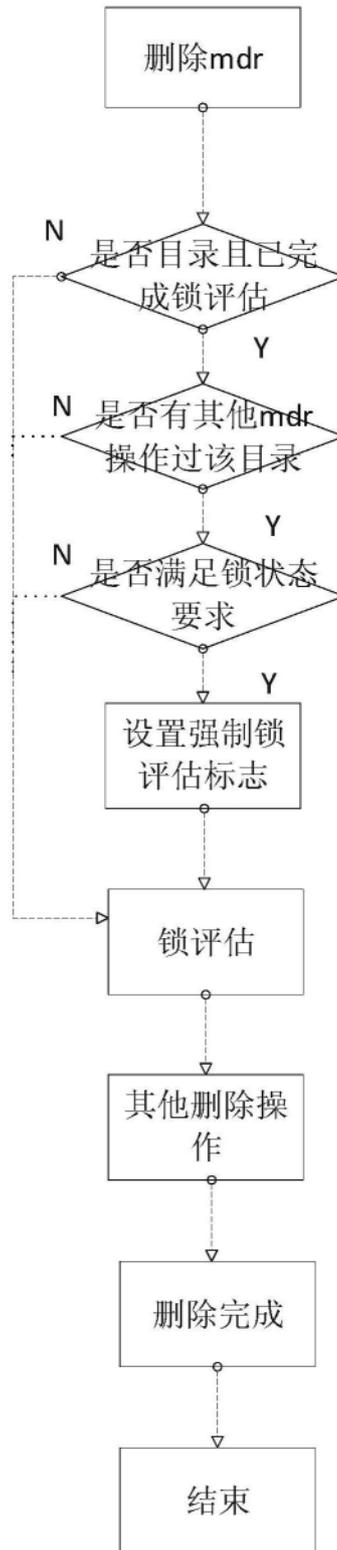


图2

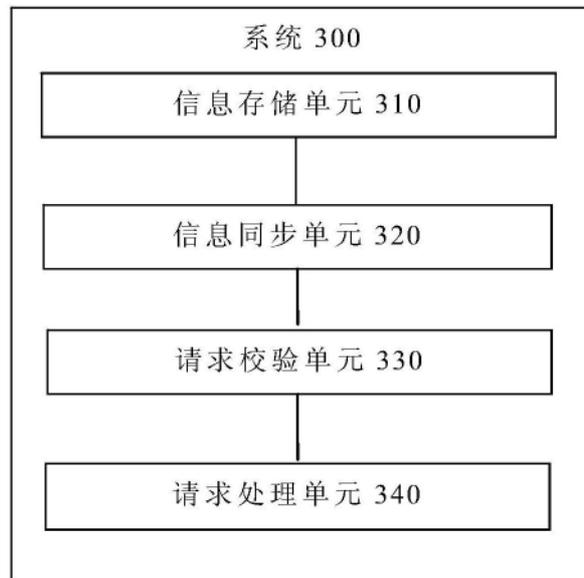


图3

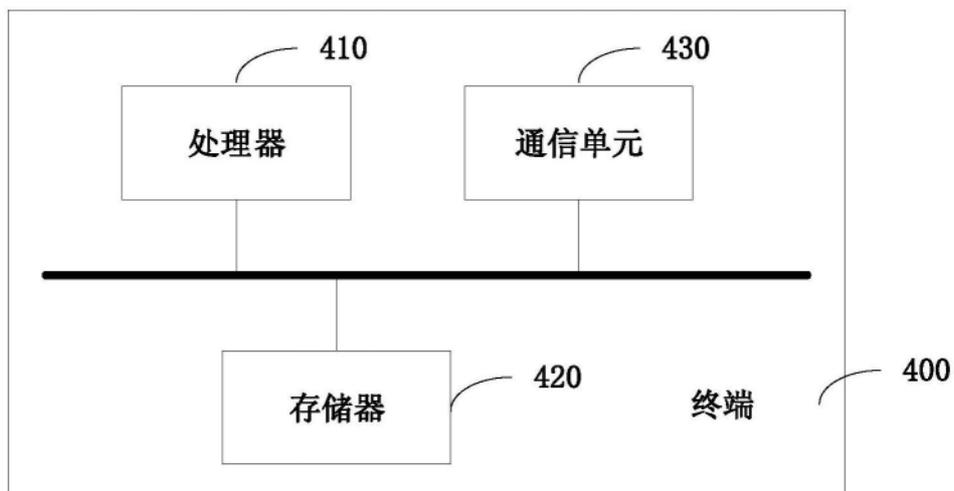


图4