



(12) 发明专利申请

(10) 申请公布号 CN 111913913 A

(43) 申请公布日 2020.11.10

(21) 申请号 202010790707.2

G06F 16/182 (2019.01)

(22) 申请日 2020.08.07

(71) 申请人 星辰天合(北京)数据科技有限公司

地址 100097 北京市海淀区蓝靛厂金源时代购物中心B区2#B座806-1

(72) 发明人 何闯 张旭明 王豪迈 胥昕

(74) 专利代理机构 北京康信知识产权代理有限公司 11240

代理人 李静茹

(51) Int. Cl.

G06F 16/13 (2019.01)

G06F 16/16 (2019.01)

G06F 16/17 (2019.01)

G06F 16/172 (2019.01)

G06F 16/18 (2019.01)

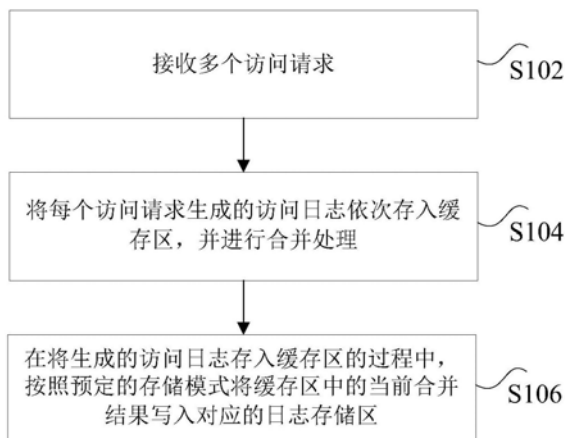
权利要求书2页 说明书8页 附图1页

(54) 发明名称

访问请求的处理方法和装置

(57) 摘要

本发明公开了一种访问请求的处理方法和装置。其中,该方法包括:接收多个访问请求,其中,访问请求用于访问存储桶;将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理;在将生成的访问日志存入缓存区的过程中,按照预定的存储模式将缓存区中的当前合并结果写入对应的日志存储区。本发明解决了相关技术中访问请求的处理方法在访问请求频繁时会消耗大量磁盘IO性能,导致性能降低的技术问题。



1. 一种访问请求的处理方法,其特征在于,包括:
接收多个访问请求,其中,所述访问请求用于访问存储桶;
将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理;
在将生成的访问日志存入所述缓存区的过程中,按照预定的存储模式将所述缓存区中的当前合并结果写入对应的日志存储区。
2. 根据权利要求1所述的方法,其特征在于,在将每个访问请求生成的访问日志依次存入缓存区之前,所述方法还包括:
基于每个访问请求的访问内容进行hash计算,生成所述访问日志,所述访问日志包括多条日志信息,其中,在所述访问内容为所述存储桶的情况下,基于所述存储桶的桶名进行hash计算;在所述访问内容为所述存储桶内存储的数据的情况下,基于所述数据的文件名进行hash计算。
3. 根据权利要求2所述的方法,其特征在于,基于每个日志对象的对象名称,将每条日志信息记录到所述缓存区的缓存图谱中。
4. 根据权利要求2所述的方法,其特征在于,将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理,包括:
基于日志信息确定所属的日志对象;
将属于同一个日志对象的日志信息合并,生成所述合并结果。
5. 根据权利要求4所述的方法,其特征在于,将所述合并结果写入到对应的日志对象所在的系统资源池,其中,
所述存储模式包括如下至少之一:按照预定的存储周期将所述缓存区中的当前合并结果写入对应的日志存储区;在所述缓存区中的缓存内容大小超过第一预定值的情况下,将所述缓存区中的当前合并结果写入对应的日志存储区。
6. 根据权利要求5所述的方法,其特征在于,如果写入所述系统资源池的合并结果的存储时间超过预定时长,则删除预定存储时间内存入的日志对象。
7. 根据权利要求5所述的方法,其特征在于,如果写入的系统资源池的使用空间占用比值超过第二预定值,则删除预定存储时间内存入的日志对象。
8. 根据权利要求1所述的方法,其特征在于,在将所述缓存区中的当前合并结果写入对应的日志存储区之前,所述方法还包括:
将日志对象的名字写入到系统资源池的管理对象的目标数据库中,其中,所述目标数据块中存储的键值为对所述存储桶的桶名和标识信息进行hash计算得到的结果,数值为所述日志对象的名字。
9. 根据权利要求1至8中任意一项所述的方法,其特征在于,在将所述合并结果写入对应的日志存储区的过程中,如果写失败,则将所述合并结果重新加载回所述缓存,并等待下次重试写入对应的日志存储区。
10. 根据权利要求1所述的方法,其特征在于,在接收多个访问请求之后,所述方法包括:校验是否记录访问日志,如果校验成功,则启动记录所述访问请求所请求的内容。
11. 一种访问请求的处理装置,其特征在于,包括:
接收模块,用于接收多个访问请求,其中,所述访问请求用于访问存储桶;
缓存模块,用于将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理;

写入模块,用于在将生成的访问日志存入所述缓存区的过程中,按照预定的存储模式将所述缓存区中的当前合并结果写入对应的日志存储区。

12.一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包括存储的程序,其中,在所述程序运行时控制所述计算机可读存储介质所在设备执行权利要求1至10中任意一项所述的访问请求的处理方法。

13.一种处理器,其特征在于,所述处理器用于运行程序,其中,所述程序运行时执行权利要求1至10中任意一项所述的访问请求的处理方法。

访问请求的处理方法和装置

技术领域

[0001] 本发明涉及数据库领域,具体而言,涉及一种访问请求的处理方法和装置。

背景技术

[0002] 在分布式文件系统ceph中,目前访问请求的处理方法是当一个请求访问存储桶后,获取相关访问信息护具,根据存储桶唯一ID和时间来生成日志对象,并将访问信息追加写到该日志对象上。同一时间段内存储桶的访问信息会存储到同一个日志对象上。随着时间变化,访问日志信息会记录到不同的日志对象上。

[0003] 但是,上述方法是同一个存储桶的每次请求访问对象网关的操作在一定时间内都会追加写到同一个日志对象上,会造成IO队列堆积,并且每次请求都要异步追加写一次造成IO次数过多。当请求频繁时会大量消耗磁盘IO性能,导致性能降低。

[0004] 针对上述的问题,目前尚未提出有效的解决方案。

发明内容

[0005] 本发明实施例提供了一种访问请求的处理方法和装置,以至少解决相关技术中访问请求的处理方法在访问请求频繁时会消耗大量磁盘IO性能,导致性能降低的技术问题。

[0006] 根据本发明实施例的一个方面,提供了一种访问请求的处理方法,包括:接收多个访问请求,其中,访问请求用于访问存储桶;将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理;在将生成的访问日志存入缓存区的过程中,按照预定的存储模式将缓存区中的当前合并结果写入对应的日志存储区。

[0007] 可选地,在将每个访问请求生成的访问日志依次存入缓存区之前,该方法还包括:基于每个访问请求的访问内容进行hash计算,生成访问日志,访问日志包括多条日志信息,其中,在访问内容为存储桶的情况下,基于存储桶的桶名进行hash计算;在访问内容为存储桶内存储的数据的情况下,基于数据的文件名进行hash计算。

[0008] 可选地,基于每个日志对象的对象名称,将每条日志信息记录到缓存区的缓存图谱中。

[0009] 可选地,将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理,包括:基于日志信息确定所属的日志对象;将属于同一个日志对象的日志信息合并,生成合并结果。

[0010] 可选地,将合并结果写入到对应的日志对象所在的系统资源池,其中,存储模式包括如下至少之一:按照预定的存储周期将缓存区中的当前合并结果写入对应的日志存储区;在缓存区中的缓存内容大小超过第一预定值的情况下,将缓存区中的当前合并结果写入对应的日志存储区。

[0011] 可选地,如果写入系统资源池的合并结果的存储时间超过预定时长,则删除预定存储时间内存入的日志对象。

[0012] 可选地,在将缓存区中的当前合并结果写入对应的日志存储区之前,该方法还包

括:将日志对象的名字写入到系统资源池的管理对象的目标数据库中,其中,目标数据块中存储的键值为对存储桶的桶名和标识信息进行hash计算得到的结果,数值为日志对象的名字。

[0013] 可选地,如果写入的系统资源池的使用空间占用比值超过第二预定值,则删除预定存储时间内存入的日志对象。

[0014] 可选地,在将合并结果写入对应的日志存储区的过程中,如果写失败,则将合并结果重新加载回缓存,并等待下次重试写入对应的日志存储区。

[0015] 可选地,在接收多个访问请求之后,该方法包括:校验是否记录访问日志,如果校验成功,则启动记录访问请求所请求的内容。

[0016] 根据本发明实施例的另一方面,还提供了一种访问请求的处理装置,包括:接收模块,用于接收多个访问请求,其中,访问请求用于访问存储桶;缓存模块,用于将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理;写入模块,用于在将生成的访问日志存入缓存区的过程中,按照预定的存储模式将缓存区中的当前合并结果写入对应的日志存储区。

[0017] 根据本发明实施例的另一方面,还提供了一种计算机可读存储介质,计算机可读存储介质包括存储的程序,其中,在程序运行时控制计算机可读存储介质所在设备执行上述的访问请求的处理方法。

[0018] 根据本发明实施例的另一方面,还提供了一种处理器,处理器用于运行程序,其中,程序运行时执行上述的访问请求的处理方法。

[0019] 在本发明实施例中,在接收到多个访问请求之后,可以将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理,进一步地,在将生成的访问日志存入缓存区的过程中,按照预定的存储模式将缓存区中的当前合并结果写入对应的日志存储区。容易注意到的是,通过将访问日志存入缓存区,并进行合并处理,从而实现优化每次访问的速度,减少日志信息写入的IO次数的目的,达到了降低磁盘IO性能消耗,提升性能的技术效果,进而解决了相关技术中访问请求的处理方法在访问请求频繁时会消耗大量磁盘IO性能,导致性能降低的技术问题。

附图说明

[0020] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0021] 图1是根据本发明实施例的一种访问请求的处理方法的流程图;

[0022] 图2是根据本发明实施例的一种可选的访问请求的处理方法的流程图;以及

[0023] 图3是根据本发明实施例的一种访问请求的处理装置的示意图。

具体实施方式

[0024] 为了使本技术领域的人员更好地理解本发明方案,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的实施例,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。

[0025] 需要说明的是,本发明的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。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本发明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。

[0026] 下面首先对本发明实施例中出现的技术名称和技术术语进行如下解释说明:

[0027] OMAP:记录key,value数据的数据库,可以是一种以键值对存储数据的数据库。

[0028] 存储桶:对象存储中的承载文件的容器。

[0029] 实施例1

[0030] 根据本发明实施例,提供了一种访问请求的处理方法,需要说明的是,在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行,并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0031] 图1是根据本发明实施例的一种访问请求的处理方法的流程图,如图1所示,该方法包括如下步骤:

[0032] 步骤S102,接收多个访问请求,其中,访问请求用于访问存储桶。

[0033] 上述步骤中的访问请求可以是请求访问对象网关的操作,也即,该访问请求会访问存储桶。

[0034] 需要说明的是,上述多个访问请求可以是同一时刻接收到的大量访问请求,也可以是短时间内接收到的大量访问请求,

[0035] 步骤S104,将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理。

[0036] 上述步骤中的缓存区可以是内存中的缓存模块,该缓存模块用于存储访问日志信息。

[0037] 在一种可选的实施例中,当有访问请求访问存储桶时,可以生成相应的访问日志信息,而且,每次写缓存内信息时,可以将属于同一个日志对象的数据进行合并,减少后续追加写到日志对象所在的系统资源池的IO次数。

[0038] 步骤S106,在将生成的访问日志存入缓存区的过程中,按照预定的存储模式将缓存区中的当前合并结果写入对应的日志存储区。

[0039] 上述步骤中的日志存储区可以是日志对象数据区域,也即,可以是日志对象所在的系统资源池。存储模式可以是预先设置的将存储区中访问日志信息追加写入到对应的日志对象数据区域的条件,在本发明上述实施例中,存储模式可以包括如下至少之一:按照预定的存储周期将缓存区中的当前合并结果写入对应的日志存储区;在缓存区中的缓存内容大小超过第一预定值的情况下,将缓存区中的当前合并结果写入对应的日志存储区。

[0040] 上述的预设周期可以是预先设置的定时器程序,例如可以是15秒,但不仅限于此。第一预定值可以是预先设置的缓存中访问日志信息的存储量,例如,可以是1000条,但不仅限于此。

[0041] 在一种可选的实施例中,可以在通过定时器程序每15秒或者访问日志信息在缓存中满足1000条的情况下,将缓存中数据追加写到对应的日志对象数据区域。

[0042] 通过本发明上述实施例,在接收到多个访问请求之后,可以将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理,进一步地,在将生成的访问日志存入缓存区的过程中,按照预定的存储模式将缓存区中的当前合并结果写入对应的日志存储区。容易注意到的是,通过将访问日志存入缓存区,并进行合并处理,从而实现优化每次访问的速度,减少日志信息写入的I/O次数的目的,达到了降低磁盘I/O性能消耗,提升性能的技术效果,进而解决了相关技术中访问请求的处理方法在访问请求频繁时会消耗大量磁盘I/O性能,导致性能降低的技术问题。

[0043] 可选地,在本发明上述实施例中,在将每个访问请求生成的访问日志依次存入缓存区之前,该方法还包括:基于每个访问请求的访问内容进行hash计算,生成访问日志,访问日志包括多条日志信息,其中,在访问内容为存储桶的情况下,基于存储桶的桶名进行hash计算;在访问内容为存储桶内存储的数据的情况下,基于数据的文件名进行hash计算。

[0044] 上述步骤中的访问日志可以是指访问日志信息集合,该集合中包含属于同一个日志对象的多条访问日志信息。

[0045] 在一种可选的实施例中,当有访问请求访问存储桶时,产生的访问日志信息可以根据访问的文件名进行hash计算,采用除留余数法,除数可以是127(表示最多由127个日志对象)。当访问请求没有访问具体的文件而只是存储桶时,可以使用存储桶的桶名进行hash计算。

[0046] 需要说明的是,每个存储桶每小时最多产生127个日志对象,如果没有访问请求则不会产生日志对象。

[0047] 可选地,在本发明上述实施例中,基于每个日志对象的对象名称,将每条日志信息记录到缓存区的缓存图谱中。

[0048] 上述步骤中的缓存图谱可以是缓存map,但不仅限于此。

[0049] 在一种可选的实施例中,可以根据hash计算的结果,并根据127个日志对象的名字分别记录到缓存map中,其中,key为日志对象的名字,value为访问日志信息集合(即上述的访问日志)。

[0050] 可选地,在本发明上述实施例中,将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理,包括:基于日志信息确定所属的日志对象;将属于同一个日志对象的日志信息合并,生成合并结果。

[0051] 在一种可选的实施例中,每次写缓存内信息时,可以将属于同一个日志对象的数据合并在一起,得到同一个日志对象的多条访问日志信息。

[0052] 可选地,在本发明上述实施例中,将合并结果写入到对应的日志对象所在的系统资源池。

[0053] 在一种可选的实施例中,每次写缓存内信息时,会将属于同一个日志对象的数据合并在一起一次性追加写到日志对象所在的系统资源池,减少I/O次数。

[0054] 可选地,在本发明上述实施例中,如果写入系统资源池的合并结果的存储时间超过预定时长,则删除预定存储时间内存入的日志对象。

[0055] 上述步骤中的预定时长可以是预先设定的日志对象能够存储的最大时长,例如,

可以是12小时,但不仅限于此。预定存储时间可以是指系统资源池中最早的存储时间,也可以是最早的一段时间,例如,预定存储时间可以是指存储了11个小时以上的时间。

[0056] 在一种可选的实施例中,如果当前累计了12个小时的日志对象在系统资源池,则可以删除最旧日志对象,也即可以直接将最早的一个或多个日志对象删除,或者将预定存储时间内的所有日志对象删除。

[0057] 可选地,在本发明上述实施例中,如果写入的系统资源池的使用空间占用比值超过第二预定值,则删除预定存储时间内存入的日志对象。

[0058] 上述步骤中的第二预定值可以是预先设置的系统资源池的最大占用比值,例如,可以是90%,但不仅限于此。

[0059] 在一种可选的实施例中,当系统资源池使用空间达到90%以后,可以删除最旧日志对象,也即可以直接将最早的一个或多个日志对象删除,或者将预定存储时间内的所有日志对象删除。

[0060] 可选地,在本发明上述实施例中,在将缓存区中的当前合并结果写入对应的日志存储区之前,该方法还包括:将日志对象的名字写入到系统资源池的管理对象的目标数据库中,其中,目标数据块中存储的键值为对存储桶的桶名和标识信息进行hash计算得到的结果,数值为日志对象的名字。

[0061] 上述步骤中的标识信息可以是存储桶的唯一ID。

[0062] 在一种可选的实施例中,可以将日志对象名字插入到value的map中,写到系统资源池的管理对象的OMAP上,其中,key为"bucket name+"-"+bucket id",用key来做hash计算,采用除留余数法,除数是31(表示最多有31个管理对象)。

[0063] 可选地,在本发明上述实施例中,在将合并结果写入对应的日志存储区的过程中,如果写失败,则将合并结果重新加载回缓存,并等待下次重试写入对应的日志存储区。

[0064] 在一种可选的实施例中,将访问日志信息集合追加写到日志对象的数据区域时,如果写失败则将访问日志信息集合重新加回缓存等待下次重试。

[0065] 可选地,在本发明上述实施例中,在接收多个访问请求之后,该方法包括:校验是否记录访问日志,如果校验成功,则启动记录访问请求所请求的内容。

[0066] 在一种可选的实施例中,在接收到访问请求之后,首先校验是否记录访问日志,如果校验成功,则记录相关访问日志信息,并将访问日志信息存储至缓存区。

[0067] 下面结合图2对本发明一种优选的实施例进行详细说明,如图2所示,该方法包括如下步骤:

[0068] 步骤S21,在接收到IO请求之后,校验是否记录访问日志。

[0069] 步骤S22,如果是,则记录相关访问日志信息。

[0070] 步骤S23,根据桶名或者对象名确定存储日志对象。

[0071] 可选的,当有请求访问存储桶时,产生的访问日志信息根据访问的文件名的进行hash计算,采用除留余数法,除数是127代表着最多有127个日志对象,当请求没有访问的具体文件只是桶时,使用桶名进行hash计算。计算的结果根据127个日志对象名字分别记录到缓存map中,key为日志对象名,value为访问日志信息集合。

[0072] 其中,每个存储桶每小时最多产生127个日志对象,如果没有请求则不会产生日志对象。

[0073] 步骤S24,访问日志信息缓存。

[0074] 可选的,通过定时器程序每15秒(可配)或者访问日志信息在缓存中满足1000条(可配)则将缓存中数据追加写到对应的日志对象数据区域,如果出现问题缓存过多超过50000条(可配)则丢掉最新的访问日志信息。

[0075] 步骤S25,加锁日志对象,将日志对象名存储到管理对象对应key的value。

[0076] 可选的,将日志对象名字插入到value的map中,写到系统资源池的管理对象OMAP上,其中key为"bucket name+"-"+bucket id",用key来做hash计算,采用除留余数法,除数是31代表着最多有31个管理对象,如果失败将访问日志信息集合加回缓存等待下次重试。

[0077] 步骤S26,访问日志信息异步存储到日志对象。

[0078] 可选的,每次写缓存内信息时,会将属于同一个日志对象的数据合并在一起一次性追加写到日志对象所在的系统资源池,减少IO次数。将访问日志信息集合追加写到日志对象的数据区域时,如果写失败则将访问日志信息集合重新加回缓存等待下次重试。

[0079] 步骤S27,判断当前日志对象记录累计情况。

[0080] 步骤S28,大于默认配置后删除。

[0081] 可选的,如果当前累计了12个小时的日志对象在系统资源池,或系统资源池使用空间达到90%,则删除最旧日志对象。

[0082] 通过上述步骤可知,本发明引入了访问日志信息内存缓存模块和一层日志对象分片映射模块,通过内存缓存日志记录信息可以优化每次访问的速度,将访问日志信息通过hash计算并合并缓存,通过定时器程序以及缓存统计计算累计到达阈值后来触发写操作已到达减少写IO次数的目的。通过日志对象分片映射可以将日志信息写到不同的日志对象分片上,之后将不同的日志对象分片元数据信息根据桶唯一ID合并到一条键值上,即达到了IO均匀分布到不同对象所在的磁盘上以提高性能又方便日志信息查找。日志对象所属的系统资源池是有多个磁盘组成的分布式存储系统,当产生127个日志对象后可以充分利用磁盘性能,增加写时并发性能。

[0083] 实施例2

[0084] 根据本发明实施例,提供了一种访问请求的处理装置,该装置可以执行上述实施例中的访问请求的处理方法,具体实现方案与优选应用场景与上述实施例相同,在此不做赘述。

[0085] 图3是根据本发明实施例的一种访问请求的处理装置的示意图,如图3所示,该装置包括:

[0086] 接收模块32,用于接收多个访问请求,其中,访问请求用于访问存储桶。

[0087] 缓存模块34,用于将每个访问请求生成的访问日志依次存入缓存区,并进行合并处理。

[0088] 写入模块36,用于在将生成的访问日志存入缓存区的过程中,按照预定的存储模式将缓存区中的当前合并结果写入对应的日志存储区。

[0089] 可选地,在本发明上述实施例中,该装置还包括:生成模块,用于在将每个访问请求生成的访问日志依次存入缓存区之前,基于每个访问请求的访问内容进行hash计算,生成访问日志,访问日志包括多条日志信息,其中,在访问内容为存储桶的情况下,基于存储桶的桶名进行hash计算;在访问内容为存储桶内存储的数据的情况下,基于数据的文件名

进行hash计算。

[0090] 可选地,在本发明上述实施例中,该装置还包括:记录模块,用于基于每个日志对象的对象名称,将每条日志信息记录到缓存区的缓存图谱中。

[0091] 可选地,在本发明上述实施例中,缓存模块包括:确定单元,用于基于日志信息确定所属的日志对象;处理单元,用于将属于同一个日志对象的日志信息合并,生成合并结果。

[0092] 可选地,在本发明上述实施例中,写入模块还用于将合并结果写入到对应的日志对象所在的系统资源池。

[0093] 可选地,在本发明上述实施例中,该装置还包括:删除模块,用于如果写入系统资源池的合并结果的存储时间超过预定时长,则删除预定存储时间内存入的日志对象。

[0094] 可选地,在本发明上述实施例中,该装置还包括:删除模块,用于如果写入的系统资源池的使用空间占用比值超过第二预定值,则删除预定存储时间内存入的日志对象。

[0095] 可选地,在本发明上述实施例中,写入模块还用于在将缓存区中的当前合并结果写入对应的日志存储区之前,将日志对象的名字写入到系统资源池的管理对象的目标数据库中,其中,目标数据块中存储的键值为对存储桶的桶名和标识信息进行hash计算得到的结果,数值为日志对象的名字。

[0096] 可选地,在本发明上述实施例中,缓存模块还用于在将合并结果写入对应的日志存储区的过程中,如果写失败,则将合并结果重新加载回缓存,并等待下次重试写入对应的日志存储区。

[0097] 可选地,在本发明上述实施例中,该装置还包括:校验模块,用于在接收多个访问请求之后,校验是否记录访问日志,如果校验成功,则启动记录访问请求所请求的内容。

[0098] 实施例3

[0099] 根据本发明实施例,提供了一种计算机可读存储介质,计算机可读存储介质包括存储的程序,其中,在程序运行时控制计算机可读存储介质所在设备执行上述实施例1中的访问请求的处理方法。

[0100] 实施例4

[0101] 根据本发明实施例,提供了一种处理器,处理器用于运行程序,其中,程序运行时执行上述实施例1中的访问请求的处理方法。

[0102] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0103] 在本发明的上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述的部分,可以参见其他实施例的相关描述。

[0104] 在本申请所提供的几个实施例中,应该理解到,所揭露的技术内容,可通过其它的方式实现。其中,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,可以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,单元或模块的间接耦合或通信连接,可以是电性或其它的形式。

[0105] 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

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。

[0106] 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。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。

[0107] 所述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,本发明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,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0108] 以上所述仅是本发明的优选实施方式,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。

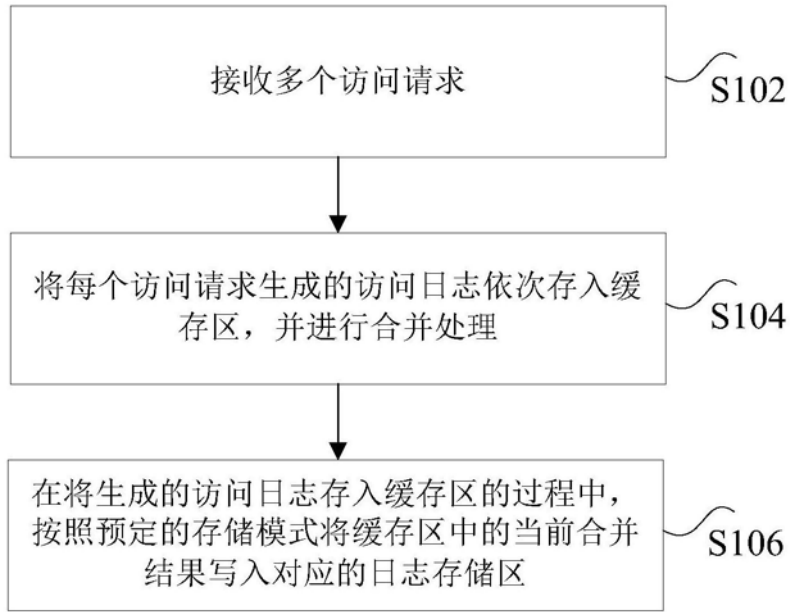


图1

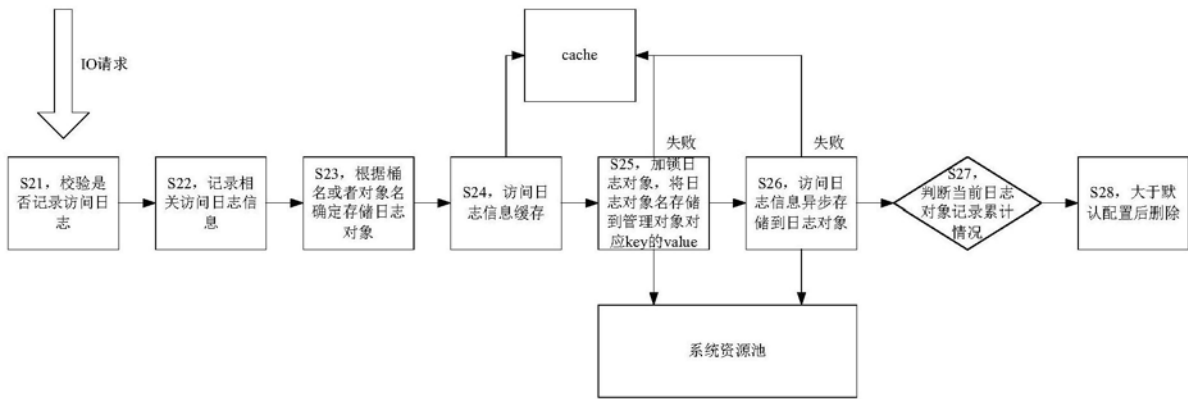


图2



图3