



(12) 发明专利申请

(10) 申请公布号 CN 117591316 A

(43) 申请公布日 2024. 02. 23

(21) 申请号 202311694799.4

(22) 申请日 2023.12.08

(71) 申请人 中电云计算技术有限公司

地址 430056 湖北省武汉市经济技术开发区
人工智能科技园N栋研发楼3层
N3013号

(72) 发明人 安佳星

(74) 专利代理机构 武汉智权专利代理事务所

(特殊普通合伙) 42225

专利代理师 洪晨曦

(51) Int. Cl.

G06F 9/54 (2006.01)

G06F 9/50 (2006.01)

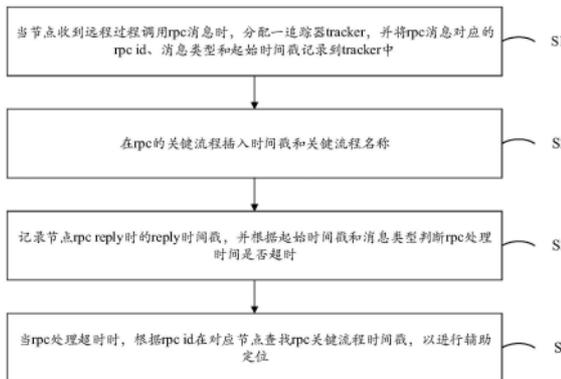
权利要求书2页 说明书7页 附图3页

(54) 发明名称

追踪存储I/O执行情况的方法、装置、设备及存储介质

(57) 摘要

一种追踪存储I/O执行情况的方法、装置、设备及存储介质,涉及数据存储技术领域,其中,追踪存储I/O执行情况的方法包括:当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;在rpc的关键流程插入时间戳和关键流程名称;记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;当rpc处理超时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。本申请能有效的监控分布式存储中的I/O处理情况,保留问题定位信息,增加了分布式存储系统的可定位性。



1. 一种追踪存储IO执行情况的方法,其特征在于,所述追踪存储IO执行情况的方法包括:

当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;

在rpc的关键流程插入时间戳和关键流程名称;

记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;

当rpc处理超时时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

2. 如权利要求1所述的追踪存储IO执行情况的方法,其特征在于:所述记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型计算rpc处理时间是否超时,包括:

判断reply时间戳和起始时间戳的差值是否超出基于消息类型设定的时间阈值;

若超出设定的时间阈值,则判定rpc处理时间超时。

3. 如权利要求2所述的追踪存储IO执行情况的方法,其特征在于:

根据不同的消息类型设置不同的时间阈值。

4. 如权利要求1所述的追踪存储IO执行情况的方法,其特征在于:

分配tracker时,在tracker pool中取出tracker,并将tracker加入tracker inflight 队列。

5. 如权利要求4所述的追踪存储IO执行情况的方法,其特征在于:

若tracker pool为空,则通过alloc函数申请tracker,并在释放时将申请的tracker放入tracker pool中。

6. 如权利要求4所述的追踪存储IO执行情况的方法,其特征在于:

若rpc陷入死循环或主动切出不继续处理,利用看门狗每隔一段时间检测tracker inflight队列rpc的起始时间戳,若超阈值未处理,则将信息输出到日志并进行故障恢复报警。

7. 如权利要求1所述的追踪存储IO执行情况的方法,其特征在于:

当rpc处理超时时,将记录的rpc关键流程时间戳输出到日志并报警。

8. 一种追踪存储IO执行情况的装置,其特征在于,所述追踪存储IO执行情况的装置包括:

分配模块,其用于当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;

插入模块,其用于在rpc的关键流程插入时间戳和关键流程名称;

判断模块,其用于记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;

定位模块,其用于当rpc处理超时时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

9. 一种追踪存储IO执行情况的设备,其特征在于,所述追踪存储IO执行情况的设备包括处理器、存储器、以及存储在所述存储器上并可被所述处理器执行的追踪存储IO执行情况的程序,其中所述追踪存储IO执行情况的程序被所述处理器执行时,实现如权利要求1至7中任一项所述的追踪存储IO执行情况的方法的步骤。

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追踪存储IO执行情况的程序,其中所述追踪存储IO执行情况的程序被处理器执行时,实现如权利要求1至7中任一项所述的追踪存储IO执行情况的方法的步骤。

追踪存储IO执行情况的方法、装置、设备及存储介质

技术领域

[0001] 本申请涉及数据存储技术领域,具体涉及一种追踪存储IO执行情况的方法、装置、设备及存储介质。

背景技术

[0002] 分布式存储系统随着服务器数量的增加,故障率也随之上升,为了保证服务器出现故障的情况下系统仍然可用,一般的做法时把一个数据分成多份储存在不同的服务器中,根据需求使用不同的冗余策略来保证数据的可靠性和容错性。

[0003] 以三副本写为例,将数据复制三份副本存在三个不同的节点上,具有一个主副本,两个从副本。客户端Client发送RPC (Remote Procedure Call, 远程过程调用) 请求 (IO) 到服务器的主副本节点上,再由主副本经网络分发到另外两个从副本中。

[0004] 为了辅助定位op异常、超时等bug,则需要对IO (Input/Output, 输入/输出) 路径上的时延进行监控,现有的IO时延监控方法大多是监控某个流程的平均时延,存在不能精确到具体IO、不能跨界点监控等缺陷;若存在平均时延符合预期但某个IO执行时间超时的情况,则不能准确的定位其超时原因。

发明内容

[0005] 本申请提供一种追踪存储IO执行情况的方法、装置、设备及存储介质,其能有效的监控分布式存储中的IO处理情况,保留问题定位信息,增加了分布式存储系统的可定位性。

[0006] 第一方面,本申请实施例提供一种追踪存储IO执行情况的方法,所述追踪存储IO执行情况的方法包括:

[0007] 当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;

[0008] 在rpc的关键流程插入时间戳和关键流程名称;

[0009] 记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;

[0010] 当rpc处理超时时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

[0011] 结合第一方面,在一种实施方式中,所述记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型计算rpc处理时间是否超时,包括:

[0012] 判断reply时间戳和起始时间戳的差值是否超出基于消息类型设定的时间阈值;

[0013] 若超出设定的时间阈值,则判定rpc处理时间超时。

[0014] 结合第一方面,在一种实施方式中,根据不同的消息类型设置不同的时间阈值。

[0015] 结合第一方面,在一种实施方式中,分配tracker时,在tracker pool中取出tracker,并将tracker加入inflight队列。

[0016] 结合第一方面,在一种实施方式中,若tracker pool为空,则通过alloc函数申请

tracker,并在释放时将申请的tracker放入tracker pool中。

[0017] 结合第一方面,在一种实施方式中,若rpc陷入死循环或主动切出不继续处理,利用看门狗每隔一段时间检测inflight队列rpc的起始时间戳,若超阈值未处理,则将信息输出到日志并进行故障恢复报警。

[0018] 结合第一方面,在一种实施方式中,当rpc处理超时,将记录的rpc关键流程时间戳输出到日志并报警。

[0019] 第二方面,本申请实施例提供了一种追踪存储IO执行情况的装置,所述追踪存储IO执行情况的装置包括:

[0020] 分配模块,其用于当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;

[0021] 插入模块,其用于在rpc的关键流程插入时间戳和关键流程名称;

[0022] 判断模块,其用于记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;

[0023] 定位模块,其用于当rpc处理超时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

[0024] 第三方面,本申请实施例提供了一种追踪存储IO执行情况的设备,所述追踪存储IO执行情况的设备包括处理器、存储器、以及存储在所述存储器上并可被所述处理器执行的追踪存储IO执行情况的程序,其中所述追踪存储IO执行情况的程序被所述处理器执行时,实现如上述任一种所述的追踪存储IO执行情况的的方法的步骤。

[0025] 第四方面,本申请实施例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追踪存储IO执行情况的程序,其中所述追踪存储IO执行情况的程序被处理器执行时,实现如上述任一种所述的追踪存储IO执行情况的的方法的步骤。

[0026] 本申请实施例提供的技术方案带来的有益效果至少包括:

[0027] 本申请中的追踪存储IO执行情况的的方法,其当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;在rpc的关键流程插入时间戳和关键流程名称;记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;当rpc处理超时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

[0028] 本申请中的方案在每个线程上维护一个tracker的内存池,用来存放追踪器的实例,可以减少内存申请释放,减少cpu和内存的消耗。以rpc id为唯一标识符,不同线程使用不同的tracker跟踪,实现免锁;另外,以rpc id为唯一标识符,还可以跨节点追踪时延。

附图说明

[0029] 图1为本申请追踪存储IO执行情况的的方法一实施例的流程示意图;

[0030] 图2为本申请tracker分配的流程;

[0031] 图3为本申请节点rpc处理流程图;

[0032] 图4为本申请追踪存储IO执行情况的装置一实施例的功能模块示意图;

[0033] 图5为本申请实施例方案中涉及的追踪存储IO执行情况的设备的硬件结构示意图。

具体实施方式

[0034] 为了使本技术领域的人员更好地理解本申请方案,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,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0035] 本申请的说明书和权利要求书及上述附图中的术语“包括”和“具有”以及它们任何变形,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、产品或设备没有限定于已列出的步骤或单元,而是可选地还包括没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其他步骤或单元。术语“第一”、“第二”和“第三”等描述,是用于区分不同的对象等,其不代表先后顺序,也不限定“第一”、“第二”和“第三”是不同的类型。

[0036] 在本申请实施例的描述中,“示例性的”、“例如”或者“举例来说”等用于表示作例子、例证或说明。本申请实施例中被描述为“示例性的”、“例如”或者“举例来说”的任何实施例或设计方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言,使用“示例性的”、“例如”或者“举例来说”等词旨在以具体方式呈现相关概念。

[0037] 在本申请实施例的描述中,除非另有说明,“/”表示或的意思,例如,A/B可以表示A或B;文本中的“和/或”仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况,另外,在本申请实施例的描述中,“多个”是指两个或多于两个。

[0038] 在本申请实施例描述的一些流程中,包含了按照特定顺序出现的多个操作或步骤,但是应该理解,这些操作或步骤可以不按照其在本申请实施例中的出现的顺序来执行或并行执行,操作的序号仅用于区分各个不同的操作,序号本身不代表任何的执行顺序。另外,这些流程可以包括更多或更少的操作,并且这些操作或步骤可以按顺序执行或并行执行,并且这些操作或步骤可以进行组合。

[0039] 为使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请实施方式作进一步地详细描述。

[0040] 第一方面,本申请实施例提供一种追踪存储IO执行情况的方法。

[0041] 一实施例中,参照图1,图1为本申请追踪存储IO执行情况的方法一实施例的流程示意图。如图1所示,追踪存储IO执行情况的方法包括:

[0042] S1、当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型(rpc type)和起始时间戳记录到tracker中。

[0043] 参见图2所示,在本实施例中,分配tracker时,采用的方式是在tracker pool中取出tracker,并将tracker加入inflight队列。若tracker pool为空,则通过alloc函数申请tracker,在释放时将申请的tracker放入tracker pool中,从而可以减少内存申请释放次数。

[0044] 参见图3所示,本实施例中的节点包括主节点和从节点,主节点接收Client写主op发送的rpc1,会分配tracker1记录rpc1 id至tracker1。随后,主节点分发从副本写op,发送rpc2,此时从节点接收从副本写rpc2,分配tracker2记录rpc2 id至tracker2。值得说明的是,主节点的tracker1也会记录rpc2 id,以保证和从节点的对接。

[0045] 可以理解的是,对于多个从节点的情形,比如三副本写的情形,其具有两个从节点,两个从节点的处理方式类似,参照图3和上述描述处理即可,本发明在此不做赘述。

[0046] S2、在rpc的关键流程插入时间戳和关键流程名称。

[0047] 本实施例中所指的关键流程,指的是rpc处理过程中常常会经历的流程,其也是本申请的重点追踪对象,关键流程比较多(细分有几十个),以图3中三个主要流程举例,在分布式存储收到一个写op后:

[0048] 1、需要准备op上下文信息(context,里面保存op所在对象,所在地址及偏移,主从分片分布等信息);

[0049] 2、需要将数据写入物理硬盘中;

[0050] 3、需要更新对象的元数据信息,包括此op对应硬盘上的位置,事务等。

[0051] 每个关键流程前后会存在时间戳,若op超时,根据这些时间戳就能计算出是哪个流程卡住。

[0052] 此外,时间戳可以使用整型压缩算法减少内存,关键流程名称在节点初始化时使用index表示,使用压缩字典算法字段减少内存。

[0053] S3、记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时。

[0054] 具体而言,步骤S3包括:

[0055] S31、判断reply时间戳和起始时间戳的差值是否超出基于消息类型设定的时间阈值;

[0056] S33、若超出设定的时间阈值,则判定rpc处理时间超时。

[0057] 以图3为例进行说明,主节点对应的是时间戳1、2、3、9、10和11,从节点对应的是时间戳4、5、6、7和8。主节点和从节点分别设置时间阈值timeout1和timeout2。

[0058] 对于从节点,若 $\text{stamp8} - \text{stamp4} > \text{timeout2}$,则判定rpc处理时间超时。

[0059] 对于主节点,若 $\text{stamp11} - \text{stamp1} > \text{timeout1}$,则判定rpc处理时间超时。

[0060] 可以理解的是,tracker1中会记录从副本的rpc2的id,通过此id可以在从副本所在节点找到超时信息,若没有则说明写从副本流程没超时,问题出在主op其他流程。

[0061] 值得说明的是,一些实施例中,不同类型的消息可定制化自己的超时时间,也即可以根据不同的消息类型设置不同的时间阈值。若rpc type为update,消息需要多个节点处理后再返回。

[0062] S4、当rpc处理超时时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

[0063] 在本实施例中,当rpc处理超时时,会将记录的rpc关键流程时间戳输出到日志并报警。

[0064] 比如,在进行追踪时,若产生超时报警,定位超时原因是从副本处理时间过长,可以通过从副本唯一标识符rpc id,到其所在节点查找从副本tracker打印信息,则可以追踪跨节点处理的消息的流程,辅助问题定位。

[0065] 此外,若存在bug或系统故障等原因,rpc在某一阶段陷入死循环或主动切出不继续处理,可以利用看门狗每隔一段时间检测tracker inflight队列rpc的起始时间戳,若超阈值未处理,可将其信息输出到日志并进行故障恢复报警等处理。

[0066] 具体而言,在开启看门狗后,会遍历tracker inflight队列,然后通过tracker起始时间戳判断是否超时,如果未超时,则看门狗休眠,如果超时,则dump tracker定位信息,上报告警。

[0067] 综上所述,本申请中的追踪存储IO执行情况的方法,其当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;在rpc的关键流程插入时间戳和关键流程名称;记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;当rpc处理超时时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

[0068] 本申请中的方案在每个线程上维护一个tracker的内存池,用来存放追踪器的实例,可以减少内存申请释放,减少cpu和内存的消耗。以rpc id为唯一标识符,不同线程使用不同的tracker跟踪,实现免锁;另外,以rpc id为唯一标识符,还可以跨节点追踪时延。从而能够将每个IO执行路径上关键节点的时延记录在内存上,在IO异常时自动dump出来,用来辅助定位。此外,时间戳,关键点信息,点位名称及含义使用压缩算法压缩字段,还可以减少内存消耗。

[0069] 第二方面,本申请实施例还提供一种追踪存储IO执行情况的装置。

[0070] 一实施例中,参照图4,图4为本申请追踪存储IO执行情况的装置一实施例的功能模块示意图。如图4所示,追踪存储IO执行情况的装置包括:

[0071] 分配模块,其用于当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;

[0072] 插入模块,其用于在rpc的关键流程插入时间戳和关键流程名称;

[0073] 判断模块,其用于记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;

[0074] 定位模块,其用于当rpc处理超时时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

[0075] 进一步地,一实施例中,所述分配模块记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型计算rpc处理时间是否超时,包括:

[0076] 判断reply时间戳和起始时间戳的差值是否超出基于消息类型设定的时间阈值;

[0077] 若超出设定的时间阈值,则判定rpc处理时间超时。

[0078] 进一步地,一实施例中,所述分配模块根据不同的消息类型设置不同的时间阈值。

[0079] 进一步地,一实施例中,所述分配模块分配tracker时,在tracker pool中取出tracker,并将tracker加入inflight队列。

[0080] 进一步地,一实施例中,所述分配模块还用于:

[0081] 若tracker pool为空,则通过alloc函数申请tracker,并在释放时将申请的tracker放入tracker pool中。

[0082] 进一步地,一实施例中,所述定位模块还用于:

[0083] 若rpc陷入死循环或主动切出不继续处理,利用看门狗每隔一段时间检测inflight队列rpc的起始时间戳,若超阈值未处理,则将信息输出到日志并进行故障恢复报警。

[0084] 进一步地,一实施例中,所述定位模块还用于:

[0085] 当rpc处理超时,将记录的rpc关键流程时间戳输出到日志并报警。

[0086] 其中,上述追踪存储IO执行情况的装置中各个模块的功能实现与上述追踪存储IO执行情况的方法实施例中各步骤相对应,其功能和实现过程在此处不再一一赘述。

[0087] 综上所述,本申请中的追踪存储IO执行情况的装置,其包括分配模块、插入模块、判断模块和定位模块。分配模块用于当节点收到远程过程调用rpc消息时,分配一追踪器tracker,并将rpc消息对应的rpc id、消息类型和起始时间戳记录到tracker中;插入模块用于在rpc的关键流程插入时间戳和关键流程名称;判断模块用于记录节点rpc reply时的reply时间戳,并根据起始时间戳和消息类型判断rpc处理时间是否超时;定位模块用于当rpc处理超时时,根据rpc id在对应节点查找rpc关键流程时间戳,以进行辅助定位。

[0088] 本申请中的方案在每个线程上维护一个tracker的内存池,用来存放追踪器的实例,可以减少内存申请释放,减少cpu和内存的消耗。以rpc id为唯一标识符,不同线程使用不同的tracker跟踪,实现免锁;另外,以rpc id为唯一标识符,还可以跨节点追踪时延。从而能够将每个IO执行路径上关键节点的时延记录在内存上,在IO异常时自动dump出来,用来辅助定位。此外,时间戳,关键点信息,点位名称及含义使用压缩算法压缩字段,还可以减少内存消耗。

[0089] 第三方面,本申请实施例提供一种追踪存储IO执行情况的设备,追踪存储IO执行情况的设备可以是个人计算机(personal computer,PC)、笔记本电脑、服务器等具有数据处理功能的设备。

[0090] 参照图5,图5为本申请实施例方案中涉及的追踪存储IO执行情况的设备的硬件结构示意图。本申请实施例中,追踪存储IO执行情况的设备可以包括处理器、存储器、通信接口以及通信总线。

[0091] 其中,通信总线可以是任何类型的,用于实现处理器、存储器以及通信接口互连。

[0092] 通信接口包括输入/输出(input/output,I/O)接口、物理接口和逻辑接口等用于实现追踪存储IO执行情况的设备内部的器件互连的接口,以及用于实现追踪存储IO执行情况的设备与其他设备(例如其他计算设备或用户设备)互连的接口。物理接口可以是以太网接口、光纤接口、ATM接口等;用户设备可以是显示屏(Display)、键盘(Keyboard)等。

[0093] 存储器可以是各种类型的存储介质,例如随机存取存储器(random access memory,RAM)、只读存储器(read-only memory,ROM)、非易失性RAM(non-volatile RAM,NVRAM)、闪存、光存储器、硬盘、可编程ROM(programmable ROM,PROM)、可擦除PROM(erasable PROM,EPROM)、电可擦除PROM(electrically erasable PROM,EEPROM)等。

[0094] 处理器可以是通用处理器,通用处理器可以调用存储器中存储的追踪存储IO执行情况的程序,并执行本申请实施例提供的追踪存储IO执行情况的方法。例如,通用处理器可以是中央处理器(central processing unit,CPU)。其中,追踪存储IO执行情况的程序被调用时所执行的方法可参照本申请追踪存储IO执行情况的方法的各个实施例,此处不再赘述。

[0095] 本领域技术人员可以理解,图5中示出的硬件结构并不构成对本申请的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0096] 第四方面,本申请实施例还提供一种计算机可读存储介质。

[0097] 本申请计算机可读存储介质上存储有追踪存储IO执行情况的程序,其中所述追踪

存储I0执行情况的程序被处理器执行时,实现如上述的追踪存储I0执行情况的方法的步骤。

[0098] 其中,追踪存储I0执行情况的程序被执行时所实现的方法可参照本申请追踪存储I0执行情况的方法的各个实施例,此处不再赘述。

[0099] 需要说明的是,上述本申请实施例序号仅仅为了描述,不代表实施例的优劣。

[0100] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。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如上所述的一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备执行本申请各个实施例所述的方法。

[0101] 以上仅为本申请的优选实施例,并非因此限制本申请的专利范围,凡是利用本申请说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本申请的专利保护范围内。

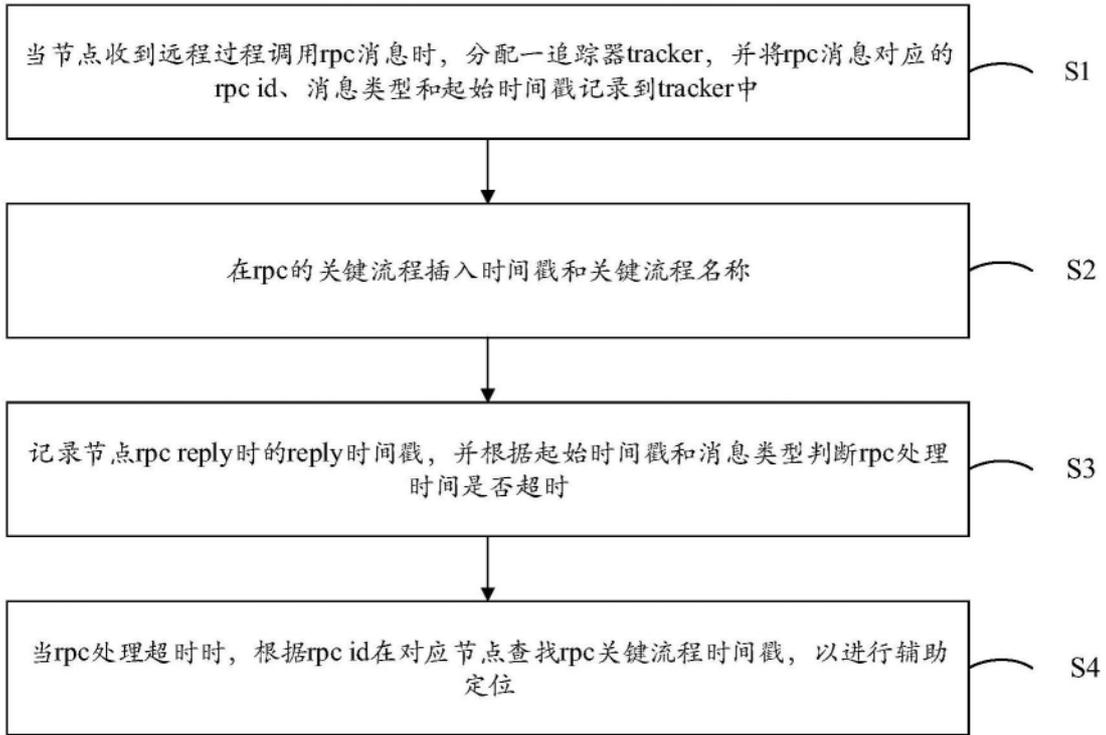


图1

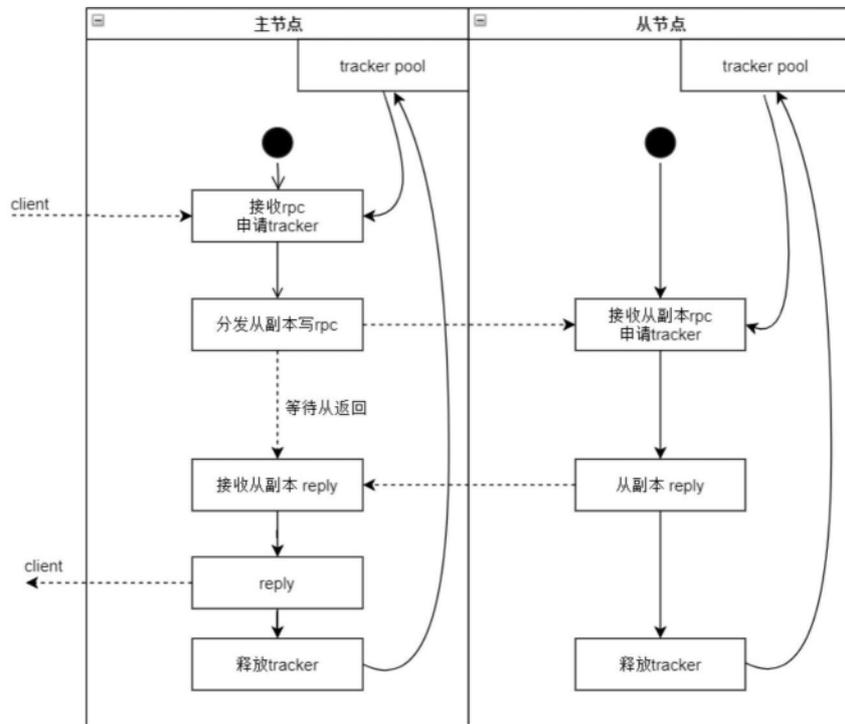


图2

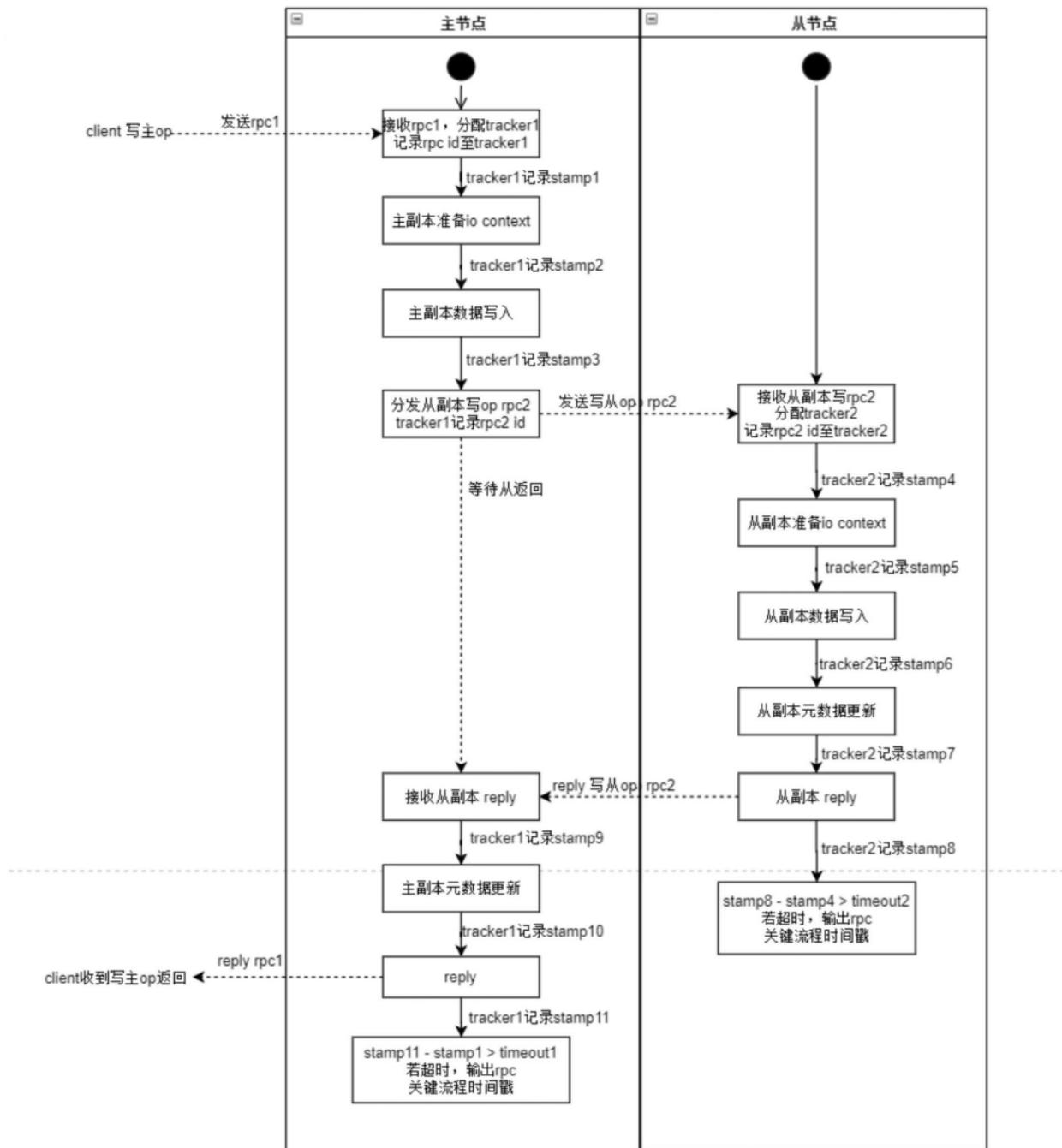


图3

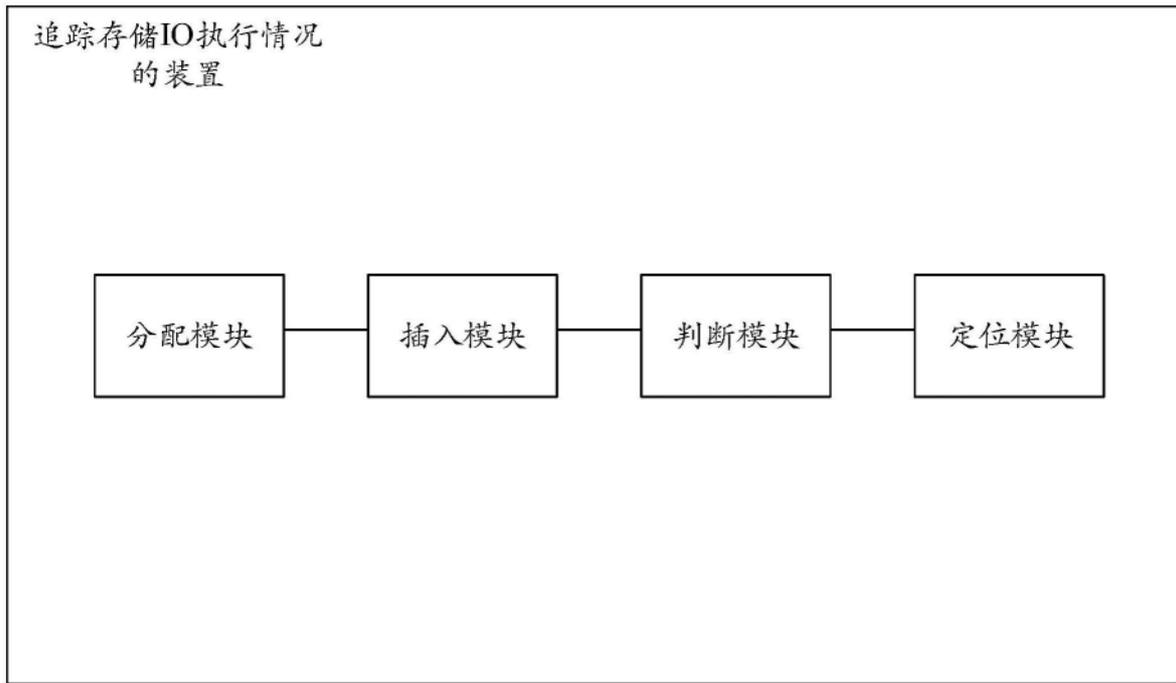


图4

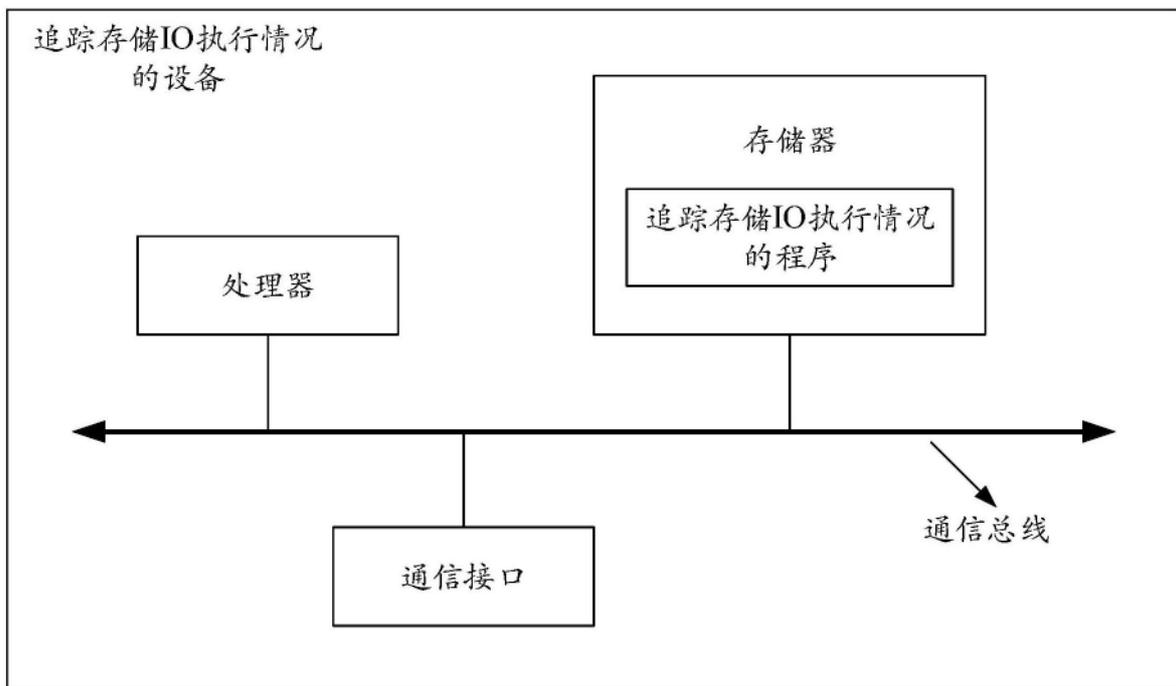


图5