(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104283956 B
(45)授权公告日 2016.01.20

(21)申请号 201410525002.2
(22)申请日 2014.09.30
(73)专利权人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市福田区振兴路赛格科技园2栋东403室
(72)发明人 张浩 余晓彬
(74)专利代理机构 上海波拓知识产权代理有限公司 31264
代理人 吕静
(51)Int.Cl.
H04L 29/08(2006.01)

(56)对比文件
CN 102882943 A,2013.01.16, 全文.
CN 103268318 A,2013.08.28, 全文.
US 2014244581 A1,2014.08.28, 全文.

审查员 田珊

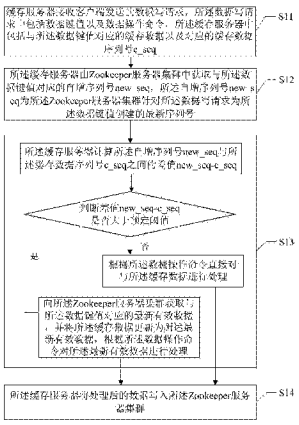
权利要求书4页 说明书10页 附图7页

(54)发明名称

强一致性分布式数据存储方法、装置及系统

(57)摘要

本发明实施例提供一种强一致性分布式数据存储方法、装置及系统，方法包括：缓存服务器接收客户端发送的数据写请求，数据写请求中包括数据键值以及数据操作命令；缓存服务器由zookeeper服务器集群中获取与数据键值对应的自增序列号；缓存服务器计算自增序列号与缓存数据序列号之间的差值，判断差值是否大于预定阈值，如果否，则根据数据操作命令直接对缓存数据进行处理，如果是，则向zookeeper服务器集群获取与数据键值对应的最新有效数据，并将缓存数据更新为最新有效数据，根据数据操作命令对最新有效数据进行处理；以及缓存服务器将处理后的数据写入zookeeper服务器集群。本发明能够在不影响Zookeeper写操作性能的情况下，提供强一致性缓存，保证数据写入的正确性。



CN 104283956 B

1. 一种强一致性分布式数据存储方法，其特征在于，包括：

缓存服务器接收客户端发送的数据写请求，所述数据写请求中包括数据键值以及数据操作命令，所述缓存服务器中包括与所述数据键值对应的缓存数据以及对应的缓存数据序列号 c_seq；

所述缓存服务器由 Zookeeper 服务器集群中获取与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；

所述缓存服务器计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理；以及

所述缓存服务器将处理后的数据写入所述 Zookeeper 服务器集群。

2. 如权利要求 1 所述的方法，其特征在于，所述向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理的步骤包括：

所述缓存服务器向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq；

所述缓存服务器计算所述自增序列号 new_seq 与所述最新数据序列号 z_seq 之间的差值 new_seq-z_seq，判断所述差值 new_seq-z_seq 是否大于所述预定阈值，

如果否，则确定所述最新数据 zk_data 为最新有效数据，将所述缓存数据更新为所述最新数据 zk_data，根据所述数据操作命令对所述最新数据 zk_data 进行处理，

如果是，则所述缓存服务器向所述 Zookeeper 服务器集群发送数据同步指令，以使所述 Zookeeper 服务器集群将与所述数据键值对应的数据同步为最新有效数据，所述缓存服务器向所述 Zookeeper 服务器集群获取同步后的最新有效数据，所述缓存服务器将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

3. 如权利要求 1 所述的方法，其特征在于，所述预定阈值为 1。

4. 如权利要求 1 或 2 所述的方法，其特征在于，所述数据操作命令为 append 写操作命令，所述对所述最新有效数据进行处理的步骤包括：对所述最新有效数据进行 append 写操作。

5. 一种强一致性分布式数据存储方法，其特征在于，包括：

客户端向缓存服务器发送数据写请求，所述数据写请求中包括数据键值以及数据操作命令；

缓存服务器根据所述数据写请求向 Zookeeper 服务器集群发送第一获取请求，用于获取与所述数据键值对应的自增序列号 new_seq，所述缓存服务器中存储有与所述数据键值对应的缓存数据以及缓存数据序列号 c_seq；

所述 Zookeeper 服务器集群根据所述第一获取请求，向所述缓存服务器返回与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；

所述缓存服务器计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则所述缓存服务器根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则所述缓存服务器向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理；以及

所述缓存服务器将处理后的数据写入所述 Zookeeper 服务器集群。

6. 如权利要求 5 所述的方法，其特征在于，所述向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理的步骤包括：

所述缓存服务器向所述 Zookeeper 服务器集群发送第二获取请求，用于获取与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq；

所述 Zookeeper 服务器集群返回与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq；

所述缓存服务器计算所述自增序列号 new_seq 与所述最新数据序列号 z_seq 之间的差值 new_seq-z_seq，判断所述差值 new_seq-z_seq 是否大于所述预定阈值，

如果否，则所述缓存服务器确定所述最新数据 zk_data 为最新有效数据，将所述缓存数据更新为所述最新数据 zk_data，根据所述数据操作命令对所述最新数据 zk_data 进行处理，

如果是，则所述缓存服务器向所述 Zookeeper 服务器集群发送数据同步指令，所述 Zookeeper 服务器集群根据所述数据同步信号将与所述数据键值对应的数据同步为最新有效数据，所述缓存服务器向所述 Zookeeper 服务器集群发送第三获取请求，用于获取同步后的最新有效数据，所述 Zookeeper 服务器集群向所述缓存服务器返回所述最新有效数据，所述缓存服务器将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

7. 如权利要求 5 所述的方法，其特征在于，所述预定阈值为 1。

8. 如权利要求 5 或 6 所述的方法，其特征在于，所述数据操作命令为 append 写操作命令，所述对所述最新有效数据进行处理的步骤，包括对所述最新有效数据进行 append 写操作。

9. 一种强一致性分布式数据存储装置，运行于缓存服务器，其特征在于，包括：

请求接收模块，用于接收客户端发送的数据写请求，所述数据写请求中包括数据键值以及数据操作命令；

数据缓存模块，用于存储与所述数据键值对应的缓存数据以及对应的缓存数据序列号 c_seq；

数据获取模块，用于由 Zookeeper 服务器集群中获取与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；

数据处理模块，用于计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则根据所述数据操作命令直接对所述缓存数据进行处理；

所述数据获取模块还用于，如果所述差值 new_seq-c_seq 大于预定阈值，向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据；

所述数据处理模块还用于将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理；以及

数据写入模块，用于将处理后的数据写入所述 Zookeeper 服务器集群。

10. 如权利要求 9 所述的装置，其特征在于，所述装置还包括数据同步模块，如果所述差值 new_seq-c_seq 大于预定阈值，

所述数据获取模块还用于，向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq；

所述数据处理模块还用于，计算所述自增序列号 new_seq 与所述最新数据序列号 z_seq 之间的差值 new_seq-z_seq，判断所述差值 new_seq-z_seq 是否大于所述预定阈值，

如果否，则确定所述最新数据 zk_data 为最新有效数据，将所述缓存数据更新为所述最新数据 zk_data，根据所述数据操作命令对所述最新数据 zk_data 进行处理，

所述数据同步模块向所述 Zookeeper 服务器集群发送数据同步指令，以使所述 Zookeeper 服务器集群将与所述数据键值对应的数据同步为最新有效数据，所述数据获取模块还用于向所述 Zookeeper 服务器集群获取同步后的最新有效数据，所述数据处理模块还用于将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

11. 一种强一致性分布式数据存储系统，其特征在于，所述系统包括客户端、缓存服务器以及 Zookeeper 服务器集群，

所述客户端用于向所述缓存服务器发送数据写请求，所述数据写请求中包括数据键值以及数据操作命令；

所述缓存服务器中存储有与所述数据键值对应的缓存数据以及缓存数据序列号 c_seq，所述缓存服务器用于，根据所述数据写请求向 Zookeeper 服务器集群发送第一获取请求，用于获取与所述数据键值对应的自增序列号 new_seq；

所述 Zookeeper 服务器集群用于，根据所述第一获取请求，向所述缓存服务器返回与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；

所述缓存服务器还用于，计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则所述缓存服务器根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则所述缓存服务器向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理，并将处理后的数据写入所述 Zookeeper 服务器集群。

12. 如权利要求 11 所述的系统，其特征在于，如果所述差值 new_seq-c_seq 大于预定阈值，

所述缓存服务器还用于向所述 Zookeeper 服务器集群发送第二获取请求，用于获取与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq；

所述 Zookeeper 服务器集群还用于返回与所述数据键值对应的最新数据 zk_data 以及

对应的最新数据序列号 z_seq ；

所述缓存服务器还用于计算所述自增序列号 new_seq 与所述最新数据序列号 z_seq 之间的差值 new_seq-z_seq，判断所述差值 new_seq-z_seq 是否大于所述预定阈值，

如果否，则所述缓存服务器确定所述最新数据 zk_data 为最新有效数据，将所述缓存数据更新为所述最新数据 zk_data，根据所述数据操作命令对所述最新数据 zk_data 进行处理，

如果是，则所述缓存服务器还用于向所述 Zookeeper 服务器集群发送数据同步指令，所述 Zookeeper 服务器集群还根据所述数据同步信号将与所述数据键值对应的数据同步为最新有效数据，所述缓存服务器还用于向所述 Zookeeper 服务器集群发送第三获取请求，用于获取同步后的最新有效数据，所述 Zookeeper 服务器集群还向所述缓存服务器返回所述最新有效数据，所述缓存服务器将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

强一致性分布式数据存储方法、装置及系统

技术领域

[0001] 本发明涉及计算机互联网技术，尤其涉及一种强一致性分布式数据存储方法、装置及系统。

背景技术

[0002] Zookeeper 是一种分布式的协调服务，为分布式应用程序提供各种服务，包括配置维护、名字服务 (Name Service)、分布式同步、组服务等。

[0003] 如图 1 所示，Zookeeper 是一个由众多服务器节点组成的集群。在 Zookeeper 中，每一个节点都被称为 Znode。Znode 维护着节点用户数据、父节点和子节点集合以及本节点状态等。这些节点又分为主节点 (leader 节点) 和从节点 (follower 节点)。Zookeeper 会保证客户端 (client) 对 Znode 的操作是顺序一致性的。

[0004] 在 Zookeeper 中，对数据的写操作，会由 follower 节点转给 leader 节点，然后由 leader 节点将请求广播给所有的 follower 节点，leader 节点收到一半以上 follower 节点的确认 (Ack) 后，就会认为写操作成功了，leader 节点会向所有 follower 节点发送 commit 信令，并通知客户端写操作成功。但是，在对数据进行读操作时，可以从任意一个节点读取，基于上述的写操作过程，可以看出，在 Zookeeper 中，并不能保证从每个 follower 节点读取的数据都是最新的有效数据，因此，Zookeeper 对于读操作只提供最终一致性服务。

[0005] 在基于 Zookeeper 的名字服务中，往往会涉及到 append 写操作，典型的 append 写操作可以由以下步骤组成 :1. 从存储里读取数据到本地，2. 在本地对数据做 append 操作，3. 将进行 append 操作后的数据从本地写回到存储中。在非强一致性分布式存储系统里，如果第 1 步中读取到数据不是最新的有效数据，则基于此结果进行 append 操作，很有可能导致写回到存储的数据是错误的，也就是说无法提供强一致性服务 (强一致性，指的是任何时刻任何请求都可以从系统中获取到最近一次成功更新的数据或者读不到有效数据)。

[0006] 对于强一致性的解决方案，业界都有一些存储系统可以提供。目前业界提供的强一致性系统，都是针对特定场景额外提供的存储系统，但是额外提供强一致性存储服务会导致系统架构很冗余。

发明内容

[0007] 有鉴于此，本发明实施例提供一种基于 Zookeeper 的强一致性分布式数据存储方法、装置及系统，可以保证写入数据的准确性。

[0008] 本发明第一实施例提出一种强一致性分布式数据存储方法，包括 :缓存服务器接收客户端发送的数据写请求，所述数据写请求中包括数据键值以及数据操作命令，所述缓存服务器中包括与所述数据键值对应的缓存数据以及对应的缓存数据序列号 c_seq ;所述缓存服务器由 Zookeeper 服务器集群中获取与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号 ;所述缓存服务器计算所述自增序列号 new_seq 与所述缓存数据

序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理；以及所述缓存服务器将处理后的数据写入所述 Zookeeper 服务器集群。

[0009] 本发明第二实施例还提出一种强一致性分布式数据存储方法，包括：客户端向缓存服务器发送数据写请求，所述数据写请求中包括数据键值以及数据操作命令；缓存服务器根据所述数据写请求向 Zookeeper 服务器集群发送第一获取请求，用于获取与所述数据键值对应的自增序列号 new_seq，所述缓存服务器中存储有与所述数据键值对应的缓存数据以及缓存数据序列号 c_seq；所述 Zookeeper 服务器集群根据所述第一获取请求，向所述缓存服务器返回与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；所述缓存服务器计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则所述缓存服务器根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则所述缓存服务器向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理；以及所述缓存服务器将处理后的数据写入所述 Zookeeper 服务器集群。

[0010] 本发明第三实施例提出一种强一致性分布式数据存储装置，运行于缓存服务器，所述装置包括：请求接收模块，用于接收客户端发送的数据写请求，所述数据写请求中包括数据键值以及数据操作命令；数据缓存模块，用于存储与所述数据键值对应的缓存数据以及对应的缓存数据序列号 c_seq；数据获取模块，用于由 Zookeeper 服务器集群中获取与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；数据处理模块，用于计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则根据所述数据操作命令直接对所述缓存数据进行处理；所述数据获取模块还用于，如果所述差值 new_seq-c_seq 大于预定阈值，向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据；所述数据处理模块还用于将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理；以及数据写入模块，用于将处理后的数据写入所述 Zookeeper 服务器集群。

[0011] 本发明第四实施例提出一种强一致性分布式数据存储系统，所述系统可以包括客户端、缓存服务器以及 Zookeeper 服务器集群，所述客户端用于向所述缓存服务器发送数据写请求，所述数据写请求中包括数据键值以及数据操作命令；所述缓存服务器中存储有与所述数据键值对应的缓存数据以及缓存数据序列号 c_seq，所述缓存服务器用于，根据所述数据写请求向 Zookeeper 服务器集群发送第一获取请求，用于获取与所述数据键值对应的自增序列号 new_seq；所述 Zookeeper 服务器集群用于，根据所述第一获取请求，向所述缓存服务器返回与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；所述

缓存服务器还用于，计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则所述缓存服务器根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则所述缓存服务器向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理，并将处理后的数据写入所述 Zookeeper 服务器集群。强一致性分布式数据存储装置强一致性分布式数据存储装置。

[0012] 相对于现有技术，本发明各实施例提出的强一致性分布式数据存储方法、装置及系统，利用 Zookeeper 中与数据键值对应的序列号来判断缓存服务器中所缓存的数据是否为最新的数据，如果是，则可以直接对其进行处理，如果不是，则向 Zookeeper 服务器集群读取、更新数据，然后对更新后的数据进行处理后再写入 Zookeeper 服务器集群，借此，在保持 Zookeeper 写操作性能不变的情况下，利用序列号，提供强一致性缓存，保证数据写入的正确性。

[0013] 为让本发明的上述和其他目的、特征和优点能更明显易懂，下文特举较佳实施例，并配合所附图式，作详细说明如下。

附图说明

[0014] 图 1 为现有的 Zookeeper 的应用环境示意图。

[0015] 图 2 为本发明实施例提供的强一致性分布式数据存储方法的应用环境图。

[0016] 图 3 为本发明第一实施例提供的强一致性分布式数据存储方法的流程图。

[0017] 图 4 为图 3 中部分流程的具体流程图。

[0018] 图 5 为本发明第二实施例提供的强一致性分布式数据存储方法的流程图。

[0019] 图 6 为图 5 中部分流程的具体流程图。

[0020] 图 7 为本发明第二实施例中一个具体示例的时序图。

[0021] 图 8 为本发明第三实施例提供的强一致性分布式数据存储装置的结构框图。

[0022] 图 9 为本发明第四实施例提供的强一致性分布式数据存储装置的结构框图。

具体实施方式

[0023] 为更进一步阐述本发明为实现预定发明目的所采取的技术手段及功效，以下结合附图及较佳实施例，对依据本发明的具体实施方式、结构、特征及其功效，详细说明如后。

[0024] 图 2 所示为本发明实施例提供的强一致性分布式数据存储方法的应用环境图，请参照图 2，本发明提供的系统 10 可以包括客户端 101、缓存服务器 102、Zookeeper 服务器集群 103。

[0025] Zookeeper 服务器集群 103 在本系统 10 里为服务器配置信息提供存储服务。本系统 10 在 Zookeeper 服务器集群 103 的前端引入缓存服务器 (cache svr)102。缓存服务器 102 会监听客户端 101 向服务器端发出的请求，并保存服务器端的回应，例如数据、结果等。

[0026] 客户端 101 写入数据时，缓存服务器 102 会接收客户端的数据写请求，每次进行数据写入前，缓存服务器 102 都会与 Zookeeper 服务器集群 103 进行数据交互，从 Zookeeper 服务器集群 103 读取、更新数据，结合客户端 101 的请求，更新本地缓存数据、处理数据，

再将处理后的数据写回到Zookeeper服务器集群103。客户端101与缓存服务器102以及Zookeeper服务器集群103之间的网络通信是基于TCP协议，TCP保证了Client/Server之间传输包的顺序，另外，Zookeeper服务器集群103执行客户端请求也是严格按照FIFO(first in first out)顺序的。

[0027] Zookeeper服务器集群103中，为了保证数据写入的顺序性，可以在每次写入数据前，针对每个数据键值(key-value)，生成对应的序列号(seq No.)，该数据键值对应的数据每写入一次，序列号就会自增一次，生成自增序列号，由于生成序列号是写操作，写操作必须经过Zookeeper的leader节点，因此自增序列号必然是最新的。于本发明的实施例中，在数据还未写入时，先生成自增序列号new_seq，即先自增，生成的这个自增序列号new_seq对应于待写入的数据，这个自增序列号new_seq之前的序列号对应于之前已写入的最新数据。

[0028] 本发明的主要思路是利用Zookeeper中与数据键值对应的序列号来判断缓存服务器102中所缓存的数据是否为最新的数据，如果是，则可以直接对其进行处理，如果不是，则向Zookeeper服务器集群103读取、更新数据，然后对更新后的数据进行处理后再写入Zookeeper服务器集群103，借此，在保持Zookeeper写操作性能不变的情况下，利用序列号，提供强一致性缓存，保证数据写入的正确性。另外，本方案直接在Zookeeper服务器集群的前端引入缓存服务器，不需要额外提供存储系统，系统架构简单。

[0029] 第一实施例

[0030] 请参见图3，为本发明第一实施例提供的强一致性分布式数据存储方法的流程图；本实施例为缓存服务器所执行的强一致性分布式数据存储方法，包括以下步骤：

[0031] 步骤S11，缓存服务器接收客户端发送的数据写请求，所述数据写请求中包括数据键值以及数据操作命令，所述缓存服务器中包括与所述数据键值对应的缓存数据以及对应的缓存数据序列号c_seq。

[0032] 步骤S12，所述缓存服务器由Zookeeper服务器集群中获取与所述数据键值对应的自增序列号new_seq，所述自增序列号new_seq为所述Zookeeper服务器集群针对所述数据写请求为所述数据键值创建的最新序列号。

[0033] 步骤S13，所述缓存服务器计算所述自增序列号new_seq与所述缓存数据序列号c_seq之间的差值new_seq-c_seq，判断所述差值new_seq-c_seq是否大于预定阈值，如果否，则根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则向所述Zookeeper服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

[0034] 请参照图4，进一步的，所述向所述Zookeeper服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理的步骤可以包括：

[0035] 步骤S131，所述缓存服务器向所述Zookeeper服务器集群获取与所述数据键值对应的最新数据zk_data以及对应的最新数据序列号z_seq；

[0036] 步骤S132，所述缓存服务器计算所述自增序列号new_seq与所述最新数据序列号z_seq之间的差值new_seq-z_seq；

[0037] 步骤S133，判断所述差值new_seq-z_seq是否大于所述预定阈值，

[0038] 步骤 S134，如果否，则确定所述最新数据 zk_data 为最新有效数据，将所述缓存数据更新为所述最新数据 zk_data，根据所述数据操作命令对所述最新数据 zk_data 进行处理；

[0039] 步骤 S135，如果是，则所述缓存服务器向所述 Zookeeper 服务器集群发送数据同步指令，以使所述 Zookeeper 服务器集群将与所述数据键值对应的数据同步为最新有效数据，所述缓存服务器向所述 Zookeeper 服务器集群获取同步后的最新有效数据，所述缓存服务器将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

[0040] 步骤 S14，所述缓存服务器将处理后的数据写入所述 Zookeeper 服务器集群。

[0041] 进一步的，Zookeeper 服务器集群还可以向缓存服务器返回写入结果，缓存服务器可以向客户端返回操作结果。

[0042] 本实施例提供的强一致性分布式数据存储方法，利用 Zookeeper 中与数据键值对应的序列号来判断缓存服务器中所缓存的数据是否为最新的数据，如果是，则可以直接对其进行处理，如果不是，则向 Zookeeper 服务器集群读取、更新数据，然后对更新后的数据进行处理后再写入 Zookeeper 服务器集群，借此，在保持 Zookeeper 写操作性能不变的情况下，利用序列号，提供强一致性缓存，保证数据写入的正确性。

[0043] 第二实施例

[0044] 图 5 为本发明第二实施例提供的强一致性分布式数据存储方法的流程图；本实施例为系统所执行的强一致性分布式数据存储方法，包括以下步骤：

[0045] 步骤 S21，客户端向缓存服务器发送数据写请求，所述数据写请求中包括数据键值以及数据操作命令。

[0046] 步骤 S22，缓存服务器根据所述数据写请求向 Zookeeper 服务器集群发送第一获取请求，用于获取与所述数据键值对应的自增序列号 new_seq，所述缓存服务器中存储有与所述数据键值对应的缓存数据以及缓存数据序列号 c_seq。

[0047] 步骤 S23，所述 Zookeeper 服务器集群根据所述第一获取请求，向所述缓存服务器返回与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号。

[0048] 步骤 S24，所述缓存服务器计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则所述缓存服务器根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则所述缓存服务器向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

[0049] 请参照图 6，进一步的，所述缓存服务器向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理的步骤可以包括：

[0050] 步骤 S241，所述缓存服务器向所述 Zookeeper 服务器集群发送第二获取请求，用于获取与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq；

[0051] 步骤 S242，所述 Zookeeper 服务器集群返回与所述数据键值对应的最新数据 zk_

data 以及对应的最新数据序列号 z_seq ;

[0052] 步骤 S243,所述缓存服务器计算所述自增序列号 new_seq 与所述最新数据序列号 z_seq 之间的差值 new_seq-z_seq ;

[0053] 步骤 S244,判断所述差值 new_seq-z_seq 是否大于所述预定阈值,

[0054] 步骤 S245,如果否,则所述缓存服务器确定所述最新数据 zk_data 为最新有效数据,将所述缓存数据更新为所述最新数据 zk_data,根据所述数据操作命令对所述最新数据 zk_data 进行处理,

[0055] 步骤 S246,如果是,则所述缓存服务器向所述 Zookeeper 服务器集群发送数据同步指令,所述 Zookeeper 服务器集群根据所述数据同步信号将与所述数据键值对应的数据同步为最新有效数据,所述缓存服务器向所述 Zookeeper 服务器集群发送第三获取请求,用于获取同步后的最新有效数据,所述 Zookeeper 服务器集群向所述缓存服务器返回所述最新有效数据,所述缓存服务器将所述缓存数据更新为所述最新有效数据,根据所述数据操作命令对所述最新有效数据进行处理。

[0056] 步骤 S25,所述缓存服务器将处理后的数据写入所述 Zookeeper 服务器集群。

[0057] 下面将结合图 7,以客户端向缓存服务器发送数据 Append 写操作请求为例,对本发明进行更详细的说明。

[0058] 请参照图 7,当客户端向 Cache 服务器发送 append 写请求 (append write request) 时,缓存服务器会向 Zookeeper 服务器集群发送第一获取请求,第一获取请求中携带有 append 写请求中的数据键值 key,第一获取请求向 Zookeeper 服务器集群获取与该数据键值 key 对应的自增序列号 new_seq。当 Cache 服务器接收到 Zookeeper 服务器返回的自增序列号 new_seq,会计算自增序列号 new_seq 与本地所存储的序列号 c_seq 之间的差值 new_seq-c_seq,由于自增序列号 new_seq 对于每次操作都会在自增一次,因此,因此,new_seq-c_seq 的结果可能是 1 或大于 1,根据 new_seq-c_seq 的结果,可能会出现以下几种场景。

[0059] 场景一 :如果 new_seq-c_seq ＝＝ 1,说明 cache 服务器中所存储的对应于该数据键值 key 的数据 c_data 已经是最新有效数据,此时,cache 服务器只要将本地序列号 c_seq 更新为 new_seq,对本地缓存的数据 c_data 进行 append 操作,并将本地数据 c_data 更新为 append 操作后的数据,然后将 append 操作后得到的数据写回 Zookeeper 服务器集群即可。

[0060] 场景二 :如果 new_seq-c_seq>1,说明 cache 服务器中所存储的对应于该数据键值 key 的数据 c_data 并不是最新有效数据,此时, cache 服务器会向 Zookeeper 发送第二获取请求,获取与该数据键值 key 对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq。由于在 Zookeeper 中,并不能保证从每个节点读取的数据都是最新的有效数据,因此,当 cache 服务器接收到 zk_data 及 z_seq 时,还要进一步判断 zk_data 是否是最新的有效数据。也就是要判断 new_seq-z_seq 是否等于 1。根据 new_seq-z_seq 的结果,可能会出现以下几种场景。

[0061] 场景三 :new_seq-c_seq>1,且 new_seq-z_seq ＝＝ 1,则说明从 Zookeeper 服务器集群读到的是最新有效数据,可以直接使用 zk_data 进行数据更新,将本地序列号 c_seq 更新为 new_seq,对 Zookeeper 服务器集群返回的数据 zk_data 进行 append 操作,并将本地数据 c_data 更新为对 zk_data 进行 append 操作后的数据,然后将 append 操作后得到的数据

写回 Zookeeper 服务器集群。

[0062] 场景四：如果 new_seq-c_seq>1，且 new_seq-z_seq>1，则说明从 Zookeeper 服务器集群读到的并不是最新有效数据，此时需要使用 sync 接口让 Zookeeper 服务器集群中的 follower 节点从集群里更新到最新有效数据，再向 Zookeeper 服务器集群发送第三获取请求，获取同步后的最新有效数据 zk_data' 以及对应的序列号 z_seq'。sync 接口是 zookeeper 提供的 java 版本的接口。Cache 服务器获取到最新的数据 zk_data' 后，为了代码级别的可靠性，还是会判断 new_seq-z_seq' 的差值是否为 1，一般此时一定是 1 的。如果不为 1，Cache 服务器会记录异常并上报，同时返回客户端写入失败，且更新 z_seq、c_seq 为 new_seq 的值。如果 new_seq-z_seq' 的差值为 1，Cache 服务器对 Zookeeper 服务器集群返回的数据 zk_data' 进行 append 操作，并将本地数据 c_data 更新为对 zk_data' 进行 append 操作后的数据，然后将 append 操作后得到的数据写回 Zookeeper 服务器集群。

[0063] 由于从 Zookeeper 服务器集群读取到不一致数据并不是常态，只有少量场景需要强制 sync 数据，因此本发明实施例提出的技术方案在性能上与普通的基于 Zookeeper 做 append 写操作的性能相当，但是可以提供强一致性服务，保证关键数据的写入正确性。

[0064] 本发明实施例还提供了一种计算机存储介质，所述计算机存储介质存储有程序，该程序执行包括图 2 至图 7 任一实施例所示方法的全部或部分步骤。

[0065] 第三实施例

[0066] 请参见图 8，为本发明第三实施例提供的强一致性分布式数据存储装置的结构示意图；本实施例提供的装置可运行于缓存服务器，所述强一致性分布式数据存储装置 30 可以包括：请求接收模块 31、数据缓存模块 32、数据获取模块 33、数据处理模块 34、数据写入模块 35、数据同步模块 36。

[0067] 其中，请求接收模块 31，用于接收客户端发送的数据写请求，所述数据写请求中包括数据键值以及数据操作命令；

[0068] 数据缓存模块 32，用于存储与所述数据键值对应的缓存数据以及对应的缓存数据序列号 c_seq；

[0069] 数据获取模块 33，用于由 Zookeeper 服务器集群中获取与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群针对所述数据写请求为所述数据键值创建的最新序列号；

[0070] 数据处理模块 34，用于计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则根据所述数据操作命令直接对所述缓存数据进行处理；

[0071] 所述数据获取模块 33 还用于，如果所述差值 new_seq-c_seq 大于预定阈值，向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新有效数据；

[0072] 所述数据处理模块 34 还用于将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理；以及

[0073] 数据写入模块 35，用于将处理后的数据写入所述 Zookeeper 服务器集群。

[0074] 如果所述差值 new_seq-c_seq 大于预定阈值，所述数据获取模块 33 还用于向所述 Zookeeper 服务器集群获取与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq；所述数据处理模块 34 还用于计算所述自增序列号 new_seq 与所述最新数据

序列号 z_seq 之间的差值 new_seq-z_seq，判断所述差值 new_seq-z_seq 是否大于所述预定阈值。

[0075] 如果否，则确定所述最新数据 zk_data 为最新有效数据，将所述缓存数据更新为所述最新数据 zk_data，根据所述数据操作命令对所述最新数据 zk_data 进行处理，数据同步模块 36 向所述 Zookeeper 服务器集群发送数据同步指令，以使所述 Zookeeper 服务器集群将与所述数据键值对应的数据同步为最新有效数据，所述数据获取模块 33 还用于向所述 Zookeeper 服务器集群获取同步后的最新有效数据，所述数据处理模块 34 还用于将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

[0076] 优选的，所述预定阈值为 1。

[0077] 所述数据操作命令例如可以为 append 写操作命令，如果数据操作命令为 append 写操作命令，所述数据处理模块 34 还用于对所述最新有效数据进行 append 写操作。

[0078] 以上各模块可以是由软件代码实现，此时，上述的各模块可存储于缓存服务器的存储器内。以上各模块同样可以由硬件例如集成电路芯片实现。

[0079] 本实施例对强一致性分布式数据存储装置 30 的各功能模块实现各自功能的具体过程，请参见上述图 2 至图 7 所示实施例中描述的具体内容，此处不再赘述。

[0080] 本实施例提供的强一致性分布式数据存储装置，利用 Zookeeper 中与数据键值对应的序列号来判断本地所缓存的数据是否为最新的数据，如果是，则可以直接对其进行处理，如果不是，则向 Zookeeper 服务器集群读取、更新数据，然后对更新后的数据进行处理后再写入 Zookeeper 服务器集群，借此，在保持 Zookeeper 写操作性能不变的情况下，利用序列号，提供强一致性缓存，保证数据写入的正确性。

[0081] 第四实施例

[0082] 请参见图 9，为本发明第四实施例提供的强一致性分布式数据存储系统的结构示意图；本实施例提供的系统 40 可以包括：客户端 41、缓存服务器 42 以及 Zookeeper 服务器集群 43。其中，

[0083] 所述客户端 41 用于向所述缓存服务器 42 发送数据写请求，所述数据写请求中包括数据键值以及数据操作命令；

[0084] 所述缓存服务器 42 中存储有与所述数据键值对应的缓存数据以及缓存数据序列号 c_seq，所述缓存服务器用于，根据所述数据写请求向 Zookeeper 服务器集群 43 发送第一获取请求，用于获取与所述数据键值对应的自增序列号 new_seq；

[0085] 所述 Zookeeper 服务器集群 43 用于，根据所述第一获取请求，向所述缓存服务器 42 返回与所述数据键值对应的自增序列号 new_seq，所述自增序列号 new_seq 为所述 Zookeeper 服务器集群 43 针对所述数据写请求为所述数据键值创建的最新序列号；

[0086] 所述缓存服务器 42 还用于，计算所述自增序列号 new_seq 与所述缓存数据序列号 c_seq 之间的差值 new_seq-c_seq，判断所述差值 new_seq-c_seq 是否大于预定阈值，如果否，则所述缓存服务器 42 根据所述数据操作命令直接对所述缓存数据进行处理，如果是，则所述缓存服务器 42 向所述 Zookeeper 服务器集群 43 获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理，并将处理后的数据写入所述 Zookeeper 服务器集群 43。

[0087] 进一步的，如果所述差值 new_seq-c_seq 大于预定阈值，所述缓存服务器 42 还用于向所述 Zookeeper 服务器集群 43 发送第二获取请求，用于获取与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq ；所述 Zookeeper 服务器集群 42 还用于返回与所述数据键值对应的最新数据 zk_data 以及对应的最新数据序列号 z_seq ；所述缓存服务器 42 还用于计算所述自增序列号 new_seq 与所述最新数据序列号 z_seq 之间的差值 new_seq-z_seq，判断所述差值 new_seq-z_seq 是否大于所述预定阈值，如果否，则所述缓存服务器 42 确定所述最新数据 zk_data 为最新有效数据，将所述缓存数据更新为所述最新数据 zk_data，根据所述数据操作命令对所述最新数据 zk_data 进行处理，如果是，则所述缓存服务器 42 还用于向所述 Zookeeper 服务器集群 43 发送数据同步指令，所述 Zookeeper 服务器集群 42 还根据所述数据同步信号将与所述数据键值对应的数据同步为最新有效数据，所述缓存服务器 42 还用于向所述 Zookeeper 服务器集群 43 发送第三获取请求，用于获取同步后的最新有效数据，所述 Zookeeper 服务器集群 43 还向所述缓存服务器 42 返回所述最新有效数据，所述缓存服务器 42 将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理。

[0088] 优选的，所述预定阈值为 1。

[0089] 所述数据操作命令可以为 append 写操作命令，如果所述数据操作命令为 append 写操作命令，所述缓存服务器 42 还用于对所述最新有效数据进行 append 写操作。

[0090] 本实施例对强一致性分布式数据存储装置 40 的各功能模块实现各自功能的具体过程，请参见上述图 2 至图 7 所示实施例中描述的具体内容，此处不再赘述。

[0091] 本实施例提供的强一致性分布式数据存储系统，利用 Zookeeper 中与数据键值对应的序列号来判断缓存服务器中所缓存的数据是否为最新的数据，如果是，则可以直接对其进行处理，如果不是，则向 Zookeeper 服务器集群读取、更新数据，然后对更新后的数据进行处理后再写入 Zookeeper 服务器集群，借此，在保持 Zookeeper 写操作性能不变的情况下，利用序列号，提供强一致性缓存，保证数据写入的正确性。另外，本方案直接在 Zookeeper 服务器集群的前端引入缓存服务器，不需要额外提供存储系统，系统架构简单。

[0092] 需要说明的是，本发明实施例的系统中各设备的功能可根据上述方法实施例中的方法具体实现，其具体实现过程可以参照上述方法实施例的相关描述，在此不赘述。

[0093] 需要说明的是，本说明书中的各个实施例均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的部分互相参见即可。对于装置类实施例而言，由于其与方法实施例基本相似，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[0094] 需要说明的是，在本文中，术语“包括”、“包含”或者其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、物品或者装置不仅包括那些要素，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、物品或者装置所固有的要素。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、物品或者装置中还存在另外的相同要素。

[0095] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，上述提到的存储介质可以是只读存储器，磁盘或光盘等。

[0096] 以上所述，仅是本发明的较佳实施例而已，并非对本发明作任何形式上的限制，虽然本发明已以较佳实施例揭露如上，然而并非用以限定本发明，任何熟悉本专业的技术人员，在不脱离本发明技术方案范围内，当可利用上述揭示的技术内容做出些许更动或修饰为等同变化的等效实施例，但凡是未脱离本发明技术方案内容，依据本发明的技术实质对以上实施例所作的任何简单修改、等同变化与修饰，均仍属于本发明技术方案的范围内。

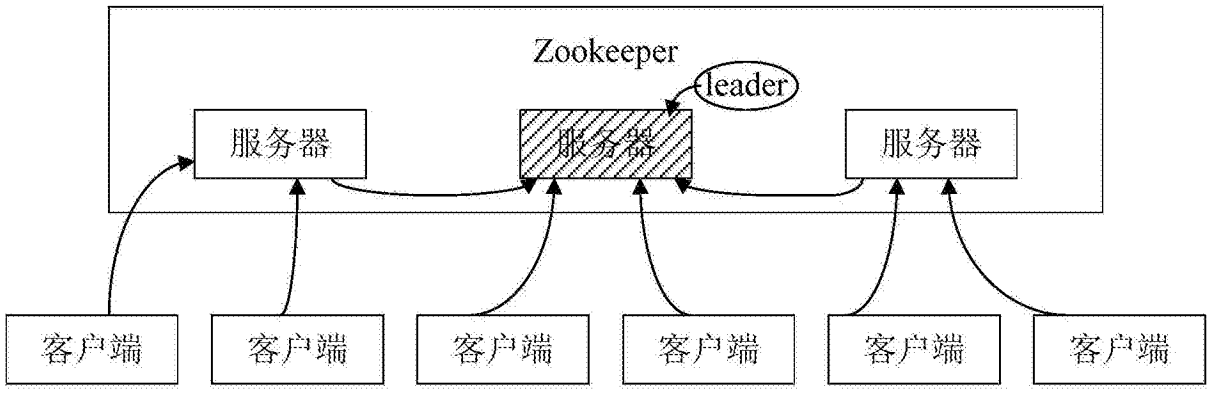

Zookeeper
leader
服务器
服务器
服务器
客户端
客户端
客户端
客户端
客户端
客户端

图 1

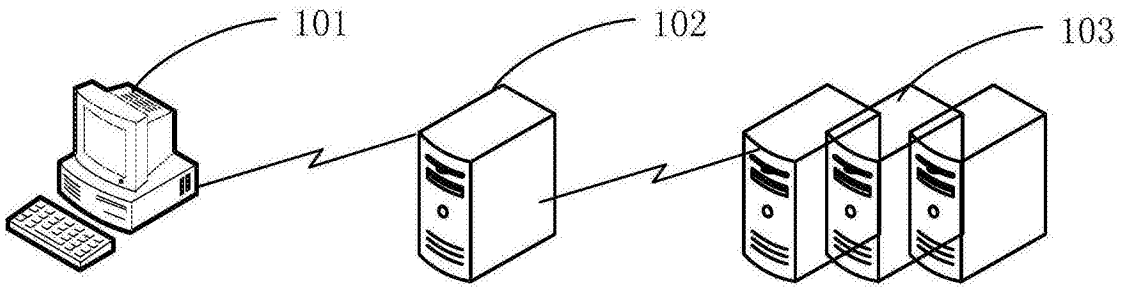

10
101
102
103

图 2

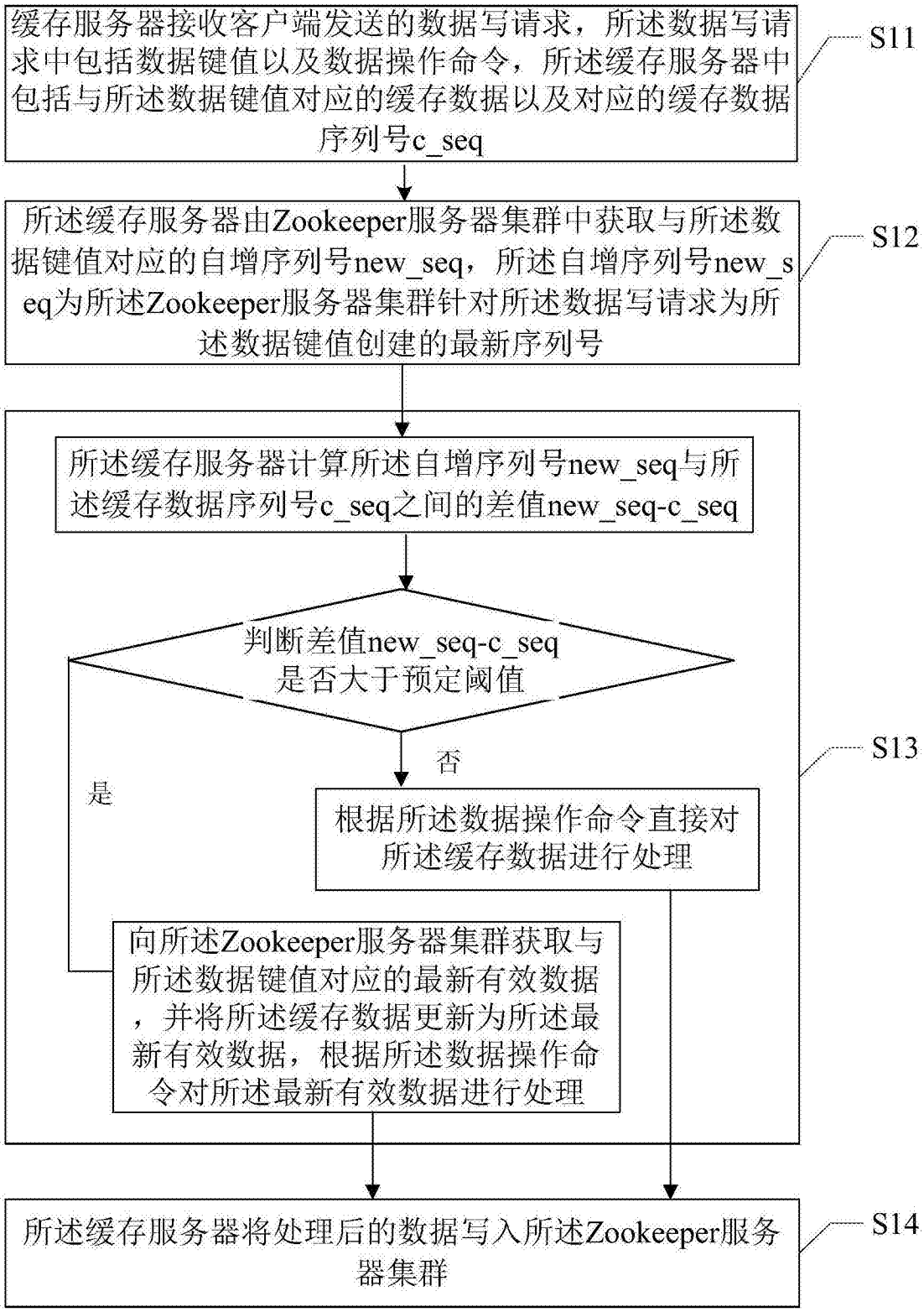

缓存服务器接收客户端发送的数据写请求，所述数据写请求中包括数据键值以及数据操作命令，所述缓存服务器中包括与所述数据键值对应的缓存数据以及对应的缓存数据序列号c_seq
S11
所述缓存服务器由Zookeeper服务器集群中获取与所述数据键值对应的自增序列号new_seq，所述自增序列号new_seq为所述Zookeeper服务器集群针对所述数据写请求为所述数据键值创建的最新序列号
S12
所述缓存服务器计算所述自增序列号new_seq与所述缓存数据序列号c_seq之间的差值new_seq-c_seq
判断差值new_seq-c_seq是否大于预定阈值
否
是
根据所述数据操作命令直接对所述缓存数据进行处理
向所述Zookeeper服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理
S13
所述缓存服务器将处理后的数据写入所述Zookeeper服务器集群
S14

图 3

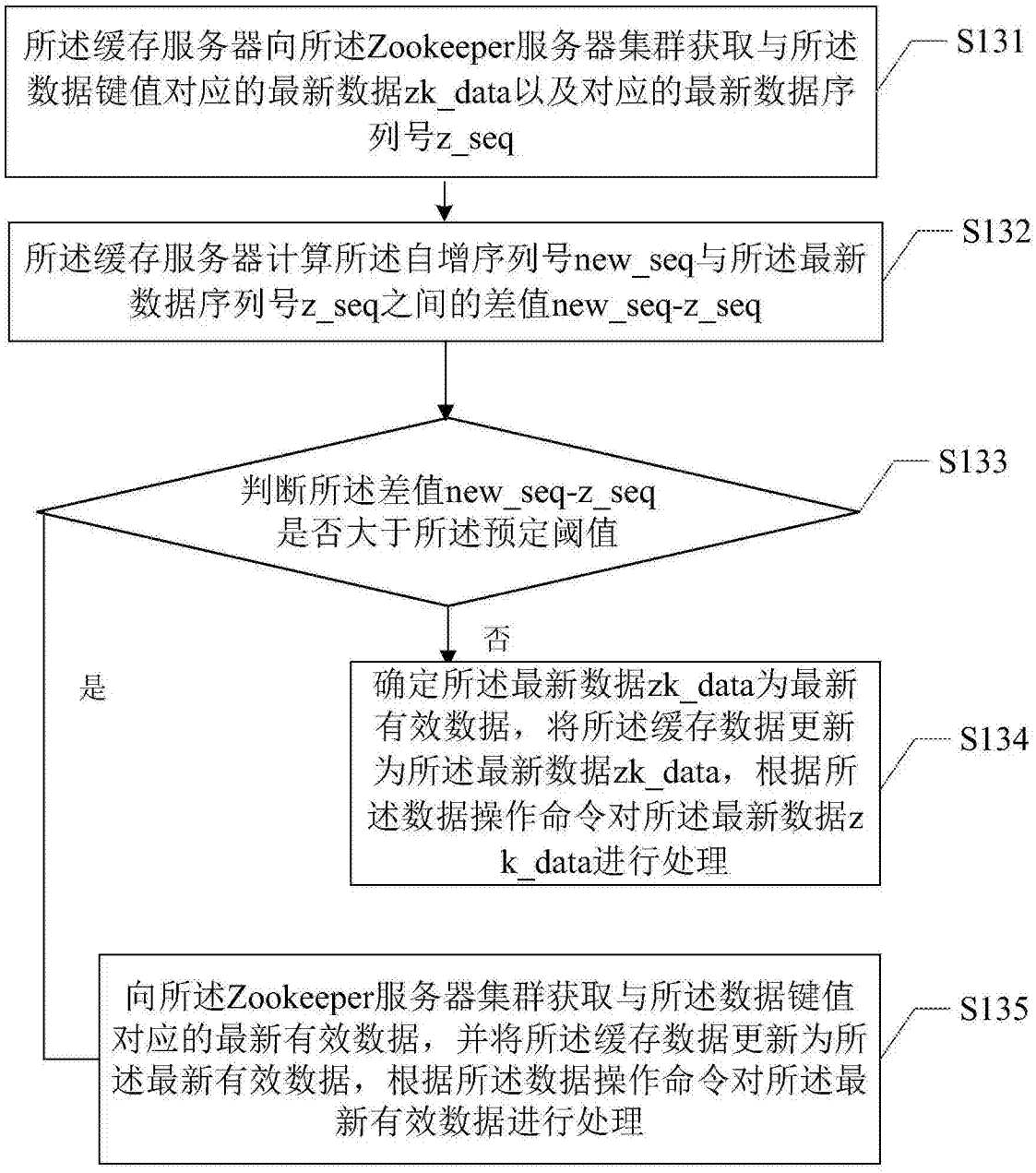

所述缓存服务器向所述Zookeeper服务器集群获取与所述数据键值对应的最新数据zk_data以及对应的最新数据序列号z_seq
S131
所述缓存服务器计算所述自增序列号new_seq与所述最新数据序列号z_seq之间的差值new_seq-z_seq
S132
判断所述差值new_seq-z_seq是否大于所述预定阈值
S133
否
是
确定所述最新数据zk_data为最新有效数据，将所述缓存数据更新为所述最新数据zk_data，根据所述数据操作命令对所述最新数据zk_data进行处理
S134
向所述Zookeeper服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理
S135

图 4

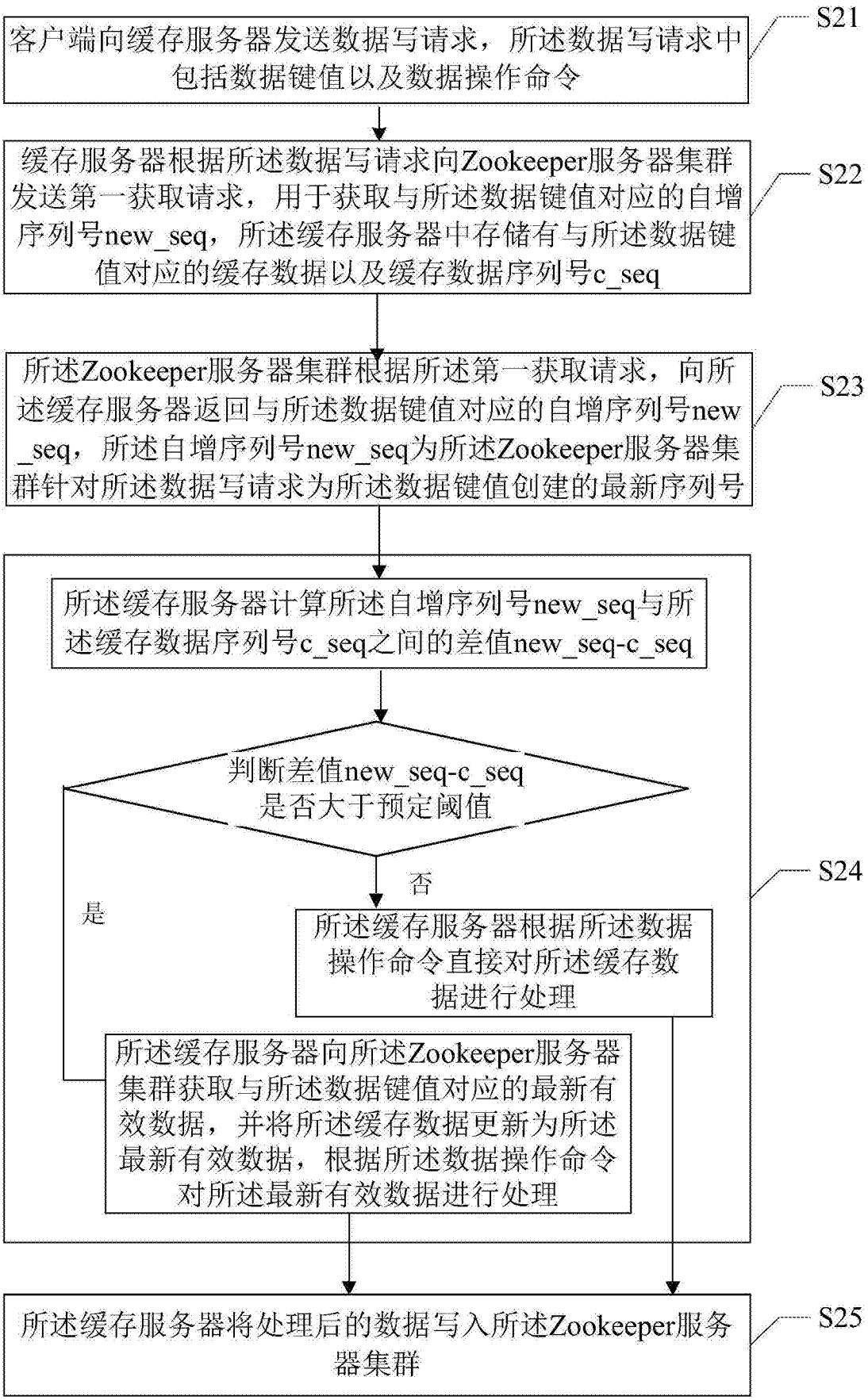

客户端向缓存服务器发送数据写请求，所述数据写请求中包括数据键值以及数据操作命令
S21
缓存服务器根据所述数据写请求向Zookeeper服务器集群发送第一获取请求，用于获取与所述数据键值对应的自增序列号new_seq，所述缓存服务器中存储有与所述数据键值对应的缓存数据以及缓存数据序列号c_seq
S22
所述Zookeeper服务器集群根据所述第一获取请求，向所述缓存服务器返回与所述数据键值对应的自增序列号new_seq，所述自增序列号new_seq为所述Zookeeper服务器集群针对所述数据写请求为所述数据键值创建的最新序列号
S23
所述缓存服务器计算所述自增序列号new_seq与所述缓存数据序列号c_seq之间的差值new_seq-c_seq
判断差值new_seq-c_seq是否大于预定阈值
否
是
所述缓存服务器根据所述数据操作命令直接对所述缓存数据进行处理
所述缓存服务器向所述Zookeeper服务器集群获取与所述数据键值对应的最新有效数据，并将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理
S24
所述缓存服务器将处理后的数据写入所述Zookeeper服务器集群
S25

图 5

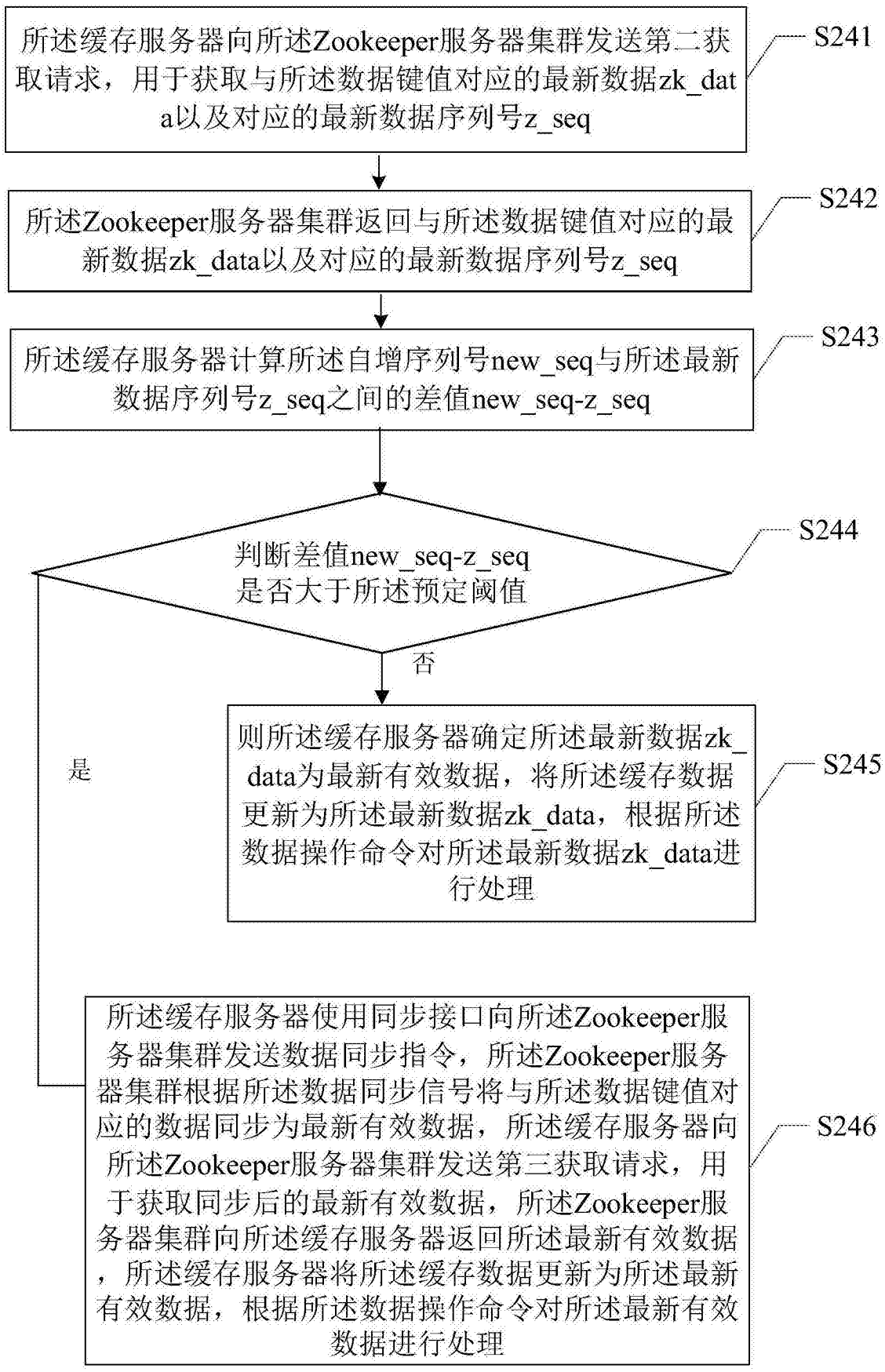

所述缓存服务器向所述Zookeeper服务器集群发送第二获取请求，用于获取与所述数据键值对应的最新数据zk_data以及对应的最新数据序列号z_seq
S241
所述Zookeeper服务器集群返回与所述数据键值对应的最新数据zk_data以及对应的最新数据序列号z_seq
S242
所述缓存服务器计算所述自增序列号new_seq与所述最新数据序列号z_seq之间的差值new_seq-z_seq
S243
判断差值new_seq-z_seq是否大于所述预定阈值
S244
否
是
则所述缓存服务器确定所述最新数据zk_data为最新有效数据，将所述缓存数据更新为所述最新数据zk_data，根据所述数据操作命令对所述最新数据zk_data进行处理
S245
所述缓存服务器使用同步接口向所述Zookeeper服务器集群发送数据同步指令，所述Zookeeper服务器集群根据所述数据同步信号将与所述数据键值对应的数据同步为最新有效数据，所述缓存服务器向所述Zookeeper服务器集群发送第三获取请求，用于获取同步后的最新有效数据，所述Zookeeper服务器集群向所述缓存服务器返回所述最新有效数据，所述缓存服务器将所述缓存数据更新为所述最新有效数据，根据所述数据操作命令对所述最新有效数据进行处理
S246

图 6

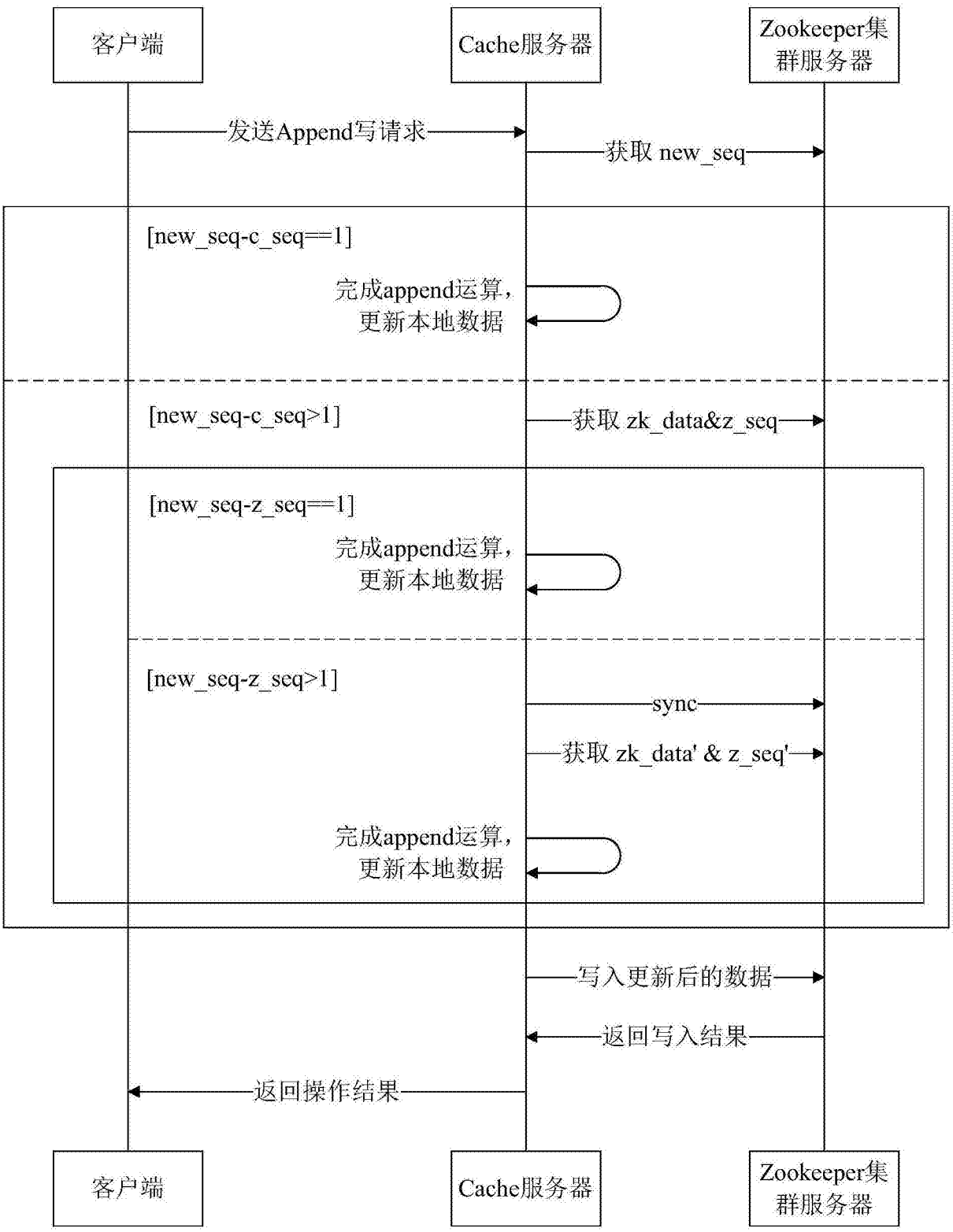

客户端
Cache服务器
Zookeeper集群服务器
发送Append写请求
获取 new_seq
[new_seq-c_seq==1]
完成append运算，更新本地数据
[new_seq-c_seq>1]
获取 zk_data&z_seq
[new_seq-z_seq==1]
完成append运算，更新本地数据
[new_seq-z_seq>1]
sync
获取 zk_data' & z_seq'
完成append运算，更新本地数据
写入更新后的数据
返回写入结果
返回操作结果
客户端
Cache服务器
Zookeeper集群服务器

图 7

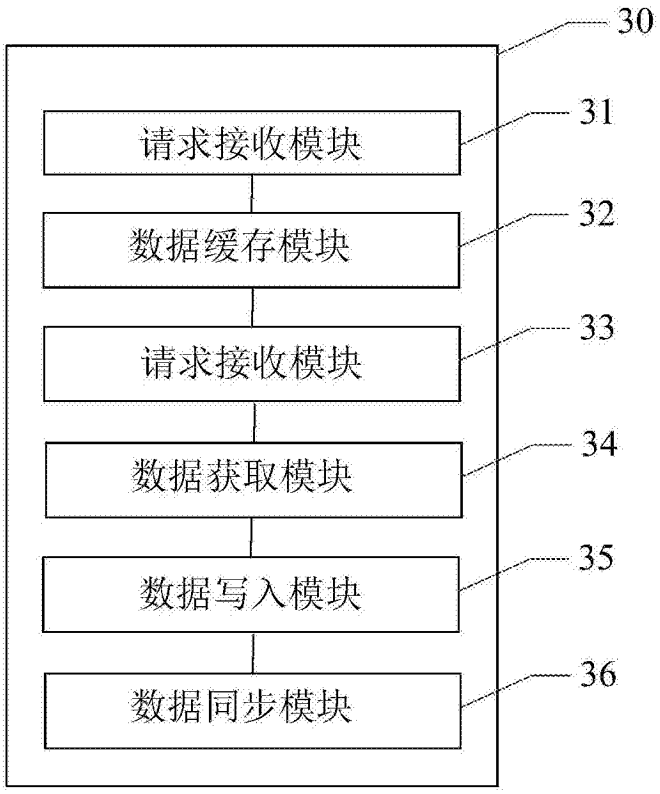

30
请求接收模块
31
数据缓存模块
32
请求接收模块
33
数据获取模块
34
数据写入模块
35
数据同步模块
36

图 8

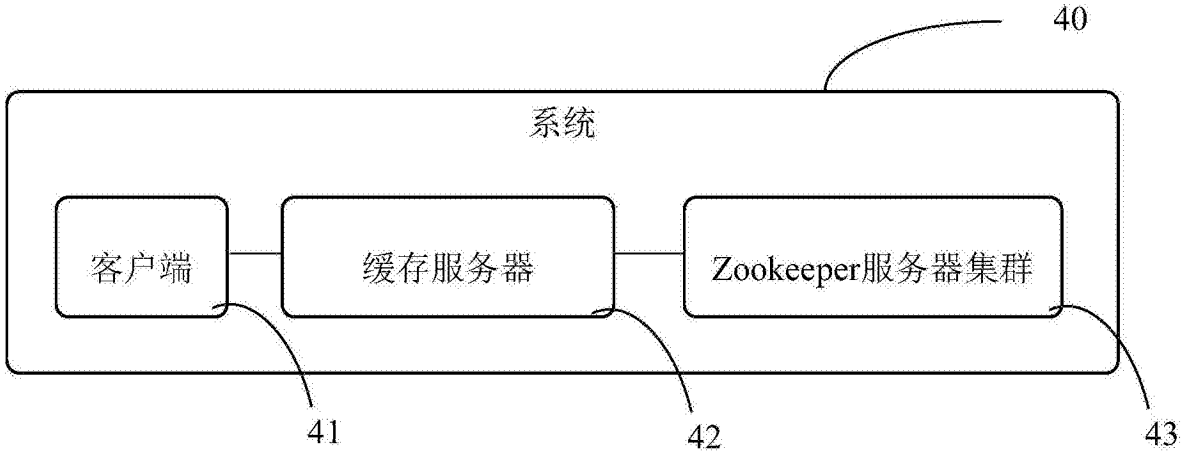

40
系统
客户端
缓存服务器
Zookeeper服务器集群
41
42
43

图 9