(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 113535746 B
(45)授权公告日 2021.11.23

(21)申请号 202111065583.2

(22)申请日 2021.09.13

(65)同一申请的已公布的文献号
申请公布号 CN 113535746 A

(43)申请公布日 2021.10.22

(73)专利权人 环球数科集团有限公司
地址 518063 广东省深圳市南山区粤海街道高新南九道10号深圳湾科技生态园10栋B座17层01-03号

(72)发明人 张卫平 丁烨 张浩宇

(74)专利代理机构 北京清控智云知识产权代理事务所(特殊普通合伙) 11919
代理人 马肃

(51)Int.Cl.
G06F 16/23(2019.01)
G06F 16/22(2019.01)
G06F 16/2455(2019.01)
G06F 16/27(2019.01)
G06F 16/28(2019.01)
G06F 11/34(2006.01)
G06F 11/30(2006.01)
G06F 9/50(2006.01)
G06F 21/64(2013.01)
G06F 21/60(2013.01)

(56)对比文件
CN 107784055 A,2018.03.09
CN 107784055 A,2018.03.09
CN 112925796 A,2021.06.08
CN 105512167 A,2016.04.20
CN 108647361 A,2018.10.12
CN 105488043 A,2016.04.13
CN 110233802 A,2019.09.13
CN 110502916 A,2019.11.26
EP 1515216 A2,2005.03.16
屠要峰等.一种分布式缓存系统的关键技术及应用.《计算机科学》.2018,(第05期),

审查员 陈丽娜

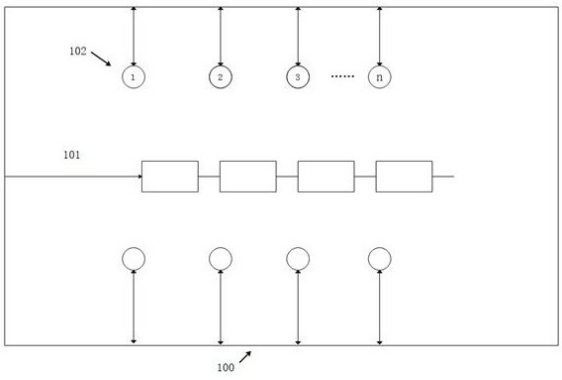
权利要求书2页 说明书9页 附图4页

(54)发明名称

一种非关系型数据通过随机序列读写后控制一致性的方法

(57)摘要

本发明提供了一种非关系型数据通过随机序列读写后控制一致性的方法;所述非关系型数据的数据库由一个分布式系统维护;分布式系统内部同时维护一条用于保存非关系型数据库键索引目录的索引公链;所述索引主链的每一个区块中写入非关系型数据库内所有键值对<key-value>所在的地址块信息,并以键key只为索引特征;当所述非关系型数据库的写操作记录达到指定阈值后,所述索引主链要求全链节点验证所述键索引目录并打包写入区块内,并在指定时间阈值内创建新的区块;路由站点、主站点、从站点根据所述索引主链的最后一个区块的所述索引记录,验证非关系型数据库中键key的存储地址位置,并以所述索引主链的记录信息作为最终合法的数据块记录依据。



CN 113535746 B

1.一种非关系型数据通过随机序列读写后控制一致性的方法,其特征在于,所述方法包括设置路由站点;所述路由站点由分布式系统上的至少一个节点担任;所述路由站点用于响应客户端对于数据库的读/写请求,并用于统计当前分布式系统上各节点的负载能力以及事务处理进度;所述控制一致性的方法还包括设置至少n个主站点,所述主站点用于响应写入操作,并且采用监听模块,在写入数据同时监听分布式系统上是否有对当前待写入数据块的写入请求;所述控制一致性的方法还包括设置至少m个从站点;所述从站点只具备只读权限,并且所述从站点只允许在所述主站点响应读取要求后,方可进行数据块的读取;

其中,分布式系统内部维护一条用于保存非关系型数据库键索引目录的索引主链;所述索引主链以联盟链形式建立,并拒绝分布式系统外的任何节点参与所述索引主链的任何操作;所述索引主链的每一个区块中包括写入非关系型数据库内所有键值对<key-value>所在的地址块信息;当所述非关系型数据库的写操作记录达到指定阈值后,所述索引主链要求全链节点完成前一区块的写入打包操作,并在指定时间阈值内创建新的区块;所述路由站点、所述主站点、所述从站点验证所述索引主链的最后一个区块的索引记录;所述索引记录包含非关系型数据库键key的存储数据块的地址位置,并以所述索引主链的记录信息作为最终合法的数据块记录依据。

2.根据权利要求1所述一种非关系型数据通过随机序列读写后控制一致性的方法,其特征在于,所述路由站点由所述索引主链的全链对链上的候选节点进行推选确定;所述路由站点的推选周期性进行;在推选所述路由站点前,由候选节点完成至少一次多并发任务的响应测试,对候选节点进行并发性能表现排序并选出至少一个候选站点成为路由站点。

3.根据权利要求2所述一种非关系型数据通过随机序列读写后控制一致性的方法,其特征在于,所述路由站点通过程序接口响应由外部提出对所述非关系型数据库的读/写请求,并对读/写请求进行区分;对其中的读请求分派到所述从站点进行处理;对其中的写请求强制写入日志记录,并赋予每一条写请求一个请求时间戳。

4.根据权利要求3所述一种非关系型数据通过随机序列读写后控制一致性的方法,其特征在于,所述主站点由所述索引主链的全链对链上的候选节点进行认证确定;所述主站点的推选周期性进行;在推选所述主站点前,由候选节点完成至少一次写入测试;所述写入测试至少包括对候选节点的连续写入、随机写入、写入延迟的性能考察;所述索引主链参考多个所述候选节点的写入性能并根据非关系型数据库的写入需求选取至少n个候选节点作为主站点。

5.根据权利要求4所述一种非关系型数据通过随机序列读写后控制一致性的方法,其特征在于,所述主站点通过多级缓存写入的方法,对所述非关系型数据的键值对<key-value>进行写入操作并建立位置对应的数据块地址;所述主站点对所述数据块地址进行写入状态标识,使所述数据块地址的写入状态至少包括等待写入状态、正在写入状态或者拒绝写入状态。

6.根据权利要求5所述一种非关系型数据通过随机序列读写后控制一致性的方法,其特征在于,所述从站点响应由所述路由站点分派的读取作;所述从站点对所述数据块地址进行读操作前,先读取所述数据块地址的写入状态,并只读取所述写入状态为等待写入状态或者拒绝写入状态的所述数据块地址内的键值对<key-value>。

7.根据权利要求6所述一种非关系型数据通过随机序列读写后控制一致性的方法,其

特征在于,多个所述主站点监听已完成的写操作的数量;当所述写操作数量超过阈值后,多个所述主站点对所述日志记录进行遍历并对所有发生过写操作的数据块地址上记录的所述键值对<key-value>中的键key进行筛重操作;对存在完全相同键值对<key-value>的不同地址块,由多个所述主站点作共识证明确认其中一个地址块作为合法地址块,并建立键值对-地址居的关联;对存在相同键key,但对应不同值value的两个以上的键值对<key-value>,多个所述主站点通过共识机制,验证所述日志记录中对键key的最后一条记录的合法性,并且选取最后一条合法的所述写入操作,并以该条最后对键值对<key-value>写入操作的地址块作为唯一合法地址块,并建立键值对-地址块的关联。

8.根据权利要求7所述一种非关系型数据通过随机序列读写后控制一致性的方法,其特征在于,在所有所述主站点通过确认所述键值对<key-value>与地址块具有唯一对应关系后,生成所述键索引目录;所述键索引目录广播到所述索引主链,由全链节点进行验证,并在验证完成后,由所述路由站点将所述键索引目录进行打包并写入所述索引主链的最后一个区块;所述路由站点同时对所述索引主链的最后一个区块进行哈希加密运算,从而获得固定长度的哈希值;所述索引主链生成一个新区块,并由所述路由站点将上一区块的哈希值写入新区块的区块头内。

9.一种分布式系统,其特征在于,包括:存储器、处理器及存储在所述存储器上并可在所述处理器上运行的一种非关系型数据通过随机序列读写后控制一致性的方法的程序;所述一种非关系型数据通过随机序列读写后控制一致性的方法被所述处理器执行时,实现如权利要求8所述的一种非关系型数据通过随机序列读写后控制一致性的方法的步骤。

一种非关系型数据通过随机序列读写后控制一致性的方法

技术领域

[0001] 本发明涉及非关系型数据库技术领域。具体而言，涉及一种非关系型数据通过随机序列读写后控制一致性的方法。

背景技术

[0002] 当今随着各种智能系统以及大数据系统的广泛应用，对数据库以及数据库相关上下链条的应用需求呈现爆发式增长。数据库为大量数据的存储提供了具有强检索性以及海量存储性的技术方案，然而随着技术的进步，数据库亦因应不同需求，开始面向不同领域作出尝试和探索。当前数据库分为关系型数据库和非关系型数据库。关系型数据库是指采用了关系模型来组织数据的数据库，例如常见地采用二维表格模型作为数据的记录型式，并通过诸如“行”“列”的作为对数据的检索方式；而一个关系型数据库就是由二维表及其之间的联系所组成的一个数据组织。非关系型数据库是指非关系型的、分布式的、且一般不保证遵循原子性、一致性、隔离性、持久性四大原则的数据存储系统。非关系型数据库以键值对<key-value>的型式存储，且结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对<key-value>，不局限于固定的结构。在非关系型数据库中，数据请求仅需要根据键key取出相应的值value就可以完成查询。非关系型数据库特别适用于如社交应用程序、音视频网站，阅读类型网站等高并发性数据请求，但是对数据一致性要求不高的应用中。并且由于非关系型数据库具有高度的可扩展灵活性，在需要系统升级、增加功能时，往往意味着数据结构巨大变动，这一点关系型数据库难以应付，而非关系型数据库则可以良好地作出适应性调整。

[0003] 然而在某些特定领域，非关系型数据在经过大量随机序列的读写操作后，尤其是大量的并发性写操作提出修改非关系型数据的值value时，极容易导致在分布式系统中存储的非关系型数据库出现一致性异常错误。这种一致性错误可能在该领域中容易造成可被裂变的数据漏洞，并且随着时间的推迟以及读写操作的指数式增加，该部分错误将难以作出一致性的真伪判断，从而影响了数据的安全和可靠性。

[0004] 查阅相关地已公开技术方案，公开号为US2016350352 (A1) 提出一种维护存储于数据库集群中多个节点的易失性存储器中的数据库对象一致性的技术；该技术使用与多个数据块相关联的共享锁，通过在写入操作中锁定该部分数据块并隔离，并在锁认数据修改的有关性后重要释放进行最终的有效性修改，从而在写入操作中保留或取消之前的写入内容；公开号为KR20110070667(A)的技术方案，提供一种用于控制基于时间的元数据缓存一致性的系统及其方法，通过维护存储在 Web 应用程序服务器的元数据缓存中的元数据与存储在数据库中的元数据之间的一致性，基于时间来维护缓存一致性；公开号为US6718347(B1)的技术方案提出通过比对两台或以上的计算机中的两套或以上的存储系统的数据一致性，从而判断数据库中正确的数据区块。以上技术方案目前都基于常规的关系型数据以及常规的静态存储方式，并未对非关系型数据库作出更多的讨论。

发明内容

[0005] 本发明的目的在于,提供一种非关系型数据通过随机序列读写后控制一致性的方法;所述控制方法通过区块链的去中心化、不可篡改以及强一致性的特点,对所述非关系型数据的数据库进行一致性的保护,并且利用非关系型数据的存储特点,只针对其中的键key与数据块建立映射关系,从而节省了在验证一致性过程中的系统算力和时间的消耗,提高了分布式系统在实现所述控制方法时的运行效率。

[0006] 本发明采用如下技术方案:

[0007] 一种非关系型数据通过随机序列读写后控制一致性的方法,所述控制方法包括设置路由站点;所述路由站点由分布式系统上的至少一个节点担任;所述路由站点用于响应客户端对于数据库的读/写请求,并用于统计当前分布式系统上各节点的负载能力以及事务处理进度;所述控制方法还包括设置至少n个主站点,所述主站点用于响应写入操作,并且采用监听模块,在写入数据同时监听分布式系统上是否有对当前待写入数据块的写入请求;所述控制一致性的方法还包括设置至少m个从站点;所述从站点只具备只读权限,并且所述从站点只允许在所述主站点响应读取要求后,方可进行数据块的读取;

[0008] 其中,分布式系统内部维护一条用于保存非关系型数据库键索引目录的索引主链;所述索引主链以联盟链形式建立,并拒绝分布式系统外的任何节点参与所述索引主链的任何操作;所述索引主链的每一个区块中包括写入非关系型数据库内所有键值对<key-value>所在的地址块信息;当所述非关系型数据库的写操作记录达到指定阈值后,所述索引主链要求全链节点完成前一区块的写入打包操作,并在指定时间阈值内创建新的区块;所述路由站点、所述主站点、所述从站点验证所述索引主链的最后一个区块的索引记录;所述索引记录包含非关系型数据库中key的存储数据块的地址位置,并以所述索引主链的记录信息作为最终合法的数据块记录依据;

[0009] 所述路由站点由所述索引主链的全链对链上的候选节点进行推选确定;所述路由站点的推选周期性进行;在推选所述路由站点前,由候选节点完成至少一次多并发任务的响应测试,对候选节点进行并发性能表现排序并选出至少一个候选站点成为路由站点;

[0010] 所述路由站点通过程序接口响应由外部提出对所述非关系型数据库的读/写请求,并对读/写请求进行区分;对其中的读请求分派到所述从站点进行处理;对其中的写请求强制写入日志记录,并赋予每一条写请求一个请求时间戳;

[0011] 所述主站点由所述索引主链的全链对链上的候选节点进行认证确定;所述主站点的推选周期性进行;在推选所述主站点前,由候选节点完成至少一次写入测试;所述写入测试至少包括对候选节点的连续写入、随机写入、写入延迟的性能考察;所述索引主链参考多个所述候选节点的写入性能并根据非关系型数据库的写入需求选取至少n个候选节点作为主站点;

[0012] 所述主站点通过多级缓存写入的方法,对所述非关系型数据的键值对<key-value>进行写入操作并建立位置对应的数据块地址;所述主站点对所述数据块地址进行写入状态标识,使所述数据块地址的写入状态至少包括等待写入状态、正在写入状态或者拒绝写入状态;

[0013] 所述从站点响应由所述路由站点分派的读取作;所述从站点对所述数据块地址进行读操作前,先读取所述数据块地址的写入状态,并只读取所述写入状态为等待写入状态

或者拒绝写入状态的所述数据块地址内的键值对<key-value>；

[0014] 多个所述主站点监听已完成的写操作的数量；当所述写操作数量超过阈值后，多个所述主站点对所述日志记录进行遍历并对所有发生过写操作的数据块地址上记录的所述键值对<key-value>中的key进行筛重操作；对存在完全相同键值对<key-value>的不同地址块，由多个所述主站点作共识证明确认其中一个地址块作为合法地址块，并建立键值对-地址居的关联；对存在相同键key，但对应不同值value的两个以上的键值对<key-value>，多个所述主站点通过共识机制，验证所述日志记录中对键key的最后一条记录的合法性，并且选取最后一条合法的所述写入操作，并以该条最后对键值对<key-value>写入操作的地址块作为唯一合法地址块，并建立键值对-地址块的关联；

[0015] 在所有所述主站点通过确认所述键值对<key-value>与地址块具有唯一对应关系后，生成所述键索引目录；所述键索引目录广播到所述索引主链，由全链节点进行验证，并在验证完成后，由所述路由站点将所述键索引目录进行打包并写入所述索引主链的最后一个区块；所述路由站点同时对所述索引主链的最后一个区块进行哈希加密运算，从而获得固定长度的哈希值；所述索引主链生成一个新区块，并由所述路由站点将上一区块的哈希值写入新区块的区块头内。

[0016] 本发明所取得的有益效果是：

[0017] 1．本控制方法综合利用了区块链与非关系型数据各自的特点，区别以往数据库为了保护数据的强一致性从而对运行数据库的系统提出极高的运算性能要求以保证，实现了成本-性能的平衡；

[0018] 2．本控制方法利用分布式上多个运算节点各自具有的性能特点，进行读/写任务的适当分配，最大化利用分布式的总算力并且保证了非关系型数据库对于高并发特性的响应速度；

[0019] 3．本控制方法建立的联盟链形式对多余的分布式系统中的节点进行隔离，避免了无关节点进行非法操作的可能性，也减少对原有分布式系统的改动，能适应分布式系统的布置形式的平滑过渡。

[0020] 4．本控制方法适用于基于各类非关系型数据库的编程系统、语言或算法，具有良好的通用性效果。

附图说明

[0021] 从以下结合附图的描述可以进一步理解本发明。图中的部件不一定按比例绘制，而是将重点放在示出实施例的原理上。在不同的视图中，相同的附图标记指定对应的部分。

[0022] 图1为本发明在分布式系统内维护的所述索引主链示意图；

[0023] 图2为本发明的分布式系统的构成示意图；

[0024] 图3为本发明所述的数据块构成示意图；

[0025] 图4为本发明所述索引目录示意图；

[0026] 图5为本发明所述具备多级缓存的节点示意图；

[0027] 附图标号说明：100-分布式系统；101-索引主链；102-分布式系统节点。

具体实施方式

[0028] 为了使得本发明的目的技术方案及优点更加清楚明白,以下结合其实施例,对本发明进行进一步详细说明;应当理解,此处所描述的具体实施例仅用于解释本发明 ,并不用于限定本发明。对于本领域技术人员而言,在查阅以下详细描述之后,本实施例的其它系统.方法和/或特征将变得显而易见。旨在所有此类附加的系统、方法、特征和优点都包括在本说明书内.包括在本发明的范围内,并且受所附权利要求书的保护。在以下详细描述描述了所公开的实施例的另外的特征,并且这些特征根据以下将详细描述将是显而易见的。

[0029] 本发明实施例的附图中相同或相似的标号对应相同或相似的部件;在本发明的描述中,需要理解的是,若有术语“上”、“下”、“左”、“右”等指示的方位或位置关系为基于附图所示的方位或位置关系,仅是为了便于描述本发明和简化描述,而不是指示或暗示所指的装置或组件必须具有特定的方位.以特定的方位构造和操作,因此附图中描述位置关系的用语仅用于示例性说明,不能理解为对本专利的限制,对于本领域的普通技术人员而言,可以根据具体情况理解上述术语的具体含义。

[0030] 实施例一:

[0031] 如附图1,一种非关系型数据通过随机序列读写后控制一致性的方法;所述控制方法包括设置路由站点;所述路由站点由分布式系统上的至少一个节点担任;所述路由站点用于响应客户端对于数据库的读/写请求,并用于统计当前分布式系统上各节点的负载能力以及事务处理进度;所述控制方法还包括设置至少n个主站点,所述主站点用于响应写入操作,并且采用监听模块,在写入数据同时监听分布式系统上是否有对当前待写入数据块的写入请求;所述控制一致性的方法还包括设置至少m个从站点;所述从站点只有只读权限,并且所述从站点只允许在所述主站点响应读取要求后,方可以进行数据块的读取;

[0032] 其中,分布式系统内部维护一条用于保存非关系型数据库键索引目录的索引主链;所述索引主链以联盟链形式建立,并拒绝分布式系统外的任何节点参与所述索引主链的任何操作;所述索引主链的每一个区块中包括写入非关系型数据库内所有键值对<key-value>所在的地址块的地址信息;当所述非关系型数据库的写操作记录达到指定阈值后,所述索引主链要求全链节点完成前一区块的写入打包操作,并在指定时间阈值内创建新的区块;所述路由站点、所述主站点、所述从站点根据所述索引主链的最后一个区块的索引记录,验证非关系型数据库中键key的存储地址块位置,并以所述索引主链的记录信息作为最终合法的数据块记录依据;

[0033] 所述路由站点由所述索引主链的全链对链上的候选节点进行推选确定;所述路由站点的推选以固定时间周期性进行;在推选所述路由站点前,由候选节点完成至少一次多并发任务的响应测试,对候选节点进行并发性能表现排序并选出至少一个候选站点成为路由站点;

[0034] 所述路由站点通过程序接口响应由外部提出对所述非关系型数据库的读/写请求,并对读/写请求进行区分;对其中的读请求分派到所述从站点进行处理;对其中的写请求生成写入记录,并将所述写入记录记入运行日志,并赋予每一条写请求一个请求时间戳;

[0035] 所述主站点由所述索引主链的全链对链上的候选节点进行认证确定;所述主站点的推选周期性进行;在推选所述主站点前,由候选节点完成一次写入测试;所述写入测试至少包括对候选节点的连续写入、随机写入、写入延迟的性能考察;所述索引主链参考多个所

述候选节点的写入性能并根据非关系型数据库的写入需求选取至少n个候选节点作为主站点；

[0036] 所述主站点通过多级缓存写入的方法，对所述非关系型数据的键值对<key-value>进行写入操作并建立位置对应的数据块地址；所述主站点对所述数据块地址进行写入状态标识，使所述数据块地址的写入状态至少包括等待写入状态、正在写入状态或者拒绝写入状态；

[0037] 所述从站点响应由所述路由站点分派的读取作；所述从站点对所述数据块地址进行读操作前，先读取所述数据块地址的写入状态，并只读取所述写入状态为等待写入状态或者拒绝写入状态的所述数据块地址内的键值对<key-value>；

[0038] 多个所述主站点监听已完成的写操作的数量；当所述写操作数量超过阈值后，多个所述主站点对所述日志记录进行遍历并对所有发生过写操作的数据块地址上记录的所述键值对<key-value>中的key进行筛重操作；对存在完全相同键值对<key-value>的不同地址块，由多个所述主站点作共识证明确认其中一个地址块作为合法地址块，并建立键值对-地址居的关联；对存在相同键key，但对应不同值value的两个以上的键值对<key-value>，多个所述主站点通过共识机制，验证所述日志记录中对键key的最后一条记录的合法性，并且选取最后一条合法的所述写入操作，并以该条最后对键值对<key-value>写入操作的地址块作为唯一合法地址块，并建立键值对-地址居的关联；

[0039] 在所有所述主站点通过确认所述键值对<key-value>与地址块具有唯一对应关系后，生成所述键索引目录；所述键索引目录广播到所述索引主链，由全链节点进行验证，并在验证完成后，由所述路由站点将所述键索引目录进行打包并写入所述索引主链的最后一个区块；所述路由站点同时对所述索引主链的最后一个区块进行哈希加密运算，从而获得固定长度的哈希值；所述索引主链生成一个新区块，并由所述路由站点将上一区块的哈希值写入新区块的区块头内；

[0040] 与具有严格表结构的关系型数据库相对的，在非关系型数据库中，数据以大量的键值对<key-value>型式存储；这些键值对<key-value>没有对存储空间位置进行严格的要求，相互之间没有耦合性，在对非关系型数据的索引过程是直接通过键key的查找，确定键key对应的值value，从而再作进一步的操作；以上特点对于海量的并发型数据请求具有相应的优势；关系型数据库在查找数据中需要大量解析行、列位置的过程，在非关系型数据库上被省略，从而提高了数据库的运作速度；

[0041] 进一步的，对于读/写操作来说，机械硬盘的读/写速度受制于物理机械结构和工艺的原因，一直难以突破，成为了数据库读取的严重瓶颈；近年随着当前半导体存储的高速发展，具有高读写速度、低读写延迟特性的固态硬盘，逐渐取代机械硬盘成为数据库存储空间的其中重要组成；然而机械硬盘的断电可存储、单位存储空间成本低的特性，又令机械硬盘仍然在数据库中具有重要的地位，目前还是大量依赖于机械硬盘作为数据库的底层存储空间；

[0042] 进一步的，对于分布式系统，分布式节点存在各具差异的计算机系统；某些节点中运行着大量机械硬盘，能够作为分布系运储中的主力存储服务器用于部署数据库；而一些节点只配置少量机械硬盘，甚至没有使用机械硬盘，但配置了一定容量的固态硬盘，其读能力突出并具有较强的数据缓存能力；而还有一些节点则配备了大容量的随机存取存储器

RAM,并且配置了多核心的中央处理器CPU,对于海量的并发式随机读/写操作具有巨大优势;

[0043] 本实施例以附图2的型式,建立分布式系统下的非关系型数据库整体架构;在业务应用层中,通过外部应用程序作为客户端,与用户进行互动操作,并获得用户的需求;这些应用程序包括基于浏览器的Web应用程序,桌面级应用程序,移动设备应用程序等;其中应用程序中包括了大量数据读/写需求,需要向后台数据库提出数据请求;这些读/写需求通过应用程序接口的定向聚焦,与分布式系统的代理层耦合;

[0044] 在分布式系统的代理层中,所述的路由站点需要处理大量的并发式读/写响应;该部分的读/写请求数量对于基础级别应用层面达到数万次/秒,甚至在大型应用中,达到数千万次/秒;因此,所述路由站点优先地需要配置多核心处理器,高速的随机存取存储器RAM;进一步的,对于分布式系统而言,每个节点可能同时运行着多个服务,因此其负载能力会在一段时间出现波动情况;因此,所述控制方法定时对担当所述路由站点的节点进行周期性的测试,或者进行周期性的负载监控和测试,推选出其中适当充当所述路由站点的至少一个节点;

[0045] 其中,所述路由站点至少需要实现以下几项功能:1. 响应由业务应用层发出的需求;2. 将需求分类,例如需求是涉及读/写,或者是只读需求;3. 监控分布式系统内各个参与节点的当前负载性能,并对所述主站点、所述从站点进行工作量负载均衡,保证应用层面的服务需求能够最大化地被分布式系统消化;

[0046] 进一步的,对于配置了大容量随机存取存储器RAM,以及系统内具备磁盘阵列甚至固态磁盘磁盘阵列作为二级或者更多级缓存的节点服务器,则可以优选地具有更优秀的写入能力,并且部署多级缓存的写入机制,例如按照Nginx本地缓存、分布式缓存、Tomcat堆缓存的方式,将对键值对<key-value>的写入请求先缓存到具备高速写入能力的存储元件内,并且明确当前地址块的地址;在后续写入队列减小时,再写入所述存储服务器内;

[0047] 其中,上述的数据块示意图如附图3;在机械硬盘、固态硬盘或者随机存取存储器RAM中存储数据,存储空间都将被划分出多个数据块block,每个数据块作为记录数据的载体单元,具有一个唯一的地址并记录在数据块的数块头(block header);其数据块的其他部分,则用于存储数据,包括未写入任何数据的零区域,以及已被写入数据的区域;每个数据块的容量可从4KB到32KB或者更大,其内部可以存储多于一条的所述键值对<key-value>;

[0048] 进一步的,所述存储服务器以机械式硬盘部署数据库为主;以大容量存储作为关注点,并且作为将非关系型数据库向关系型数据库转化的存储节点,以使数据库具备长久保存或者冷备份功能;所述存储服务器与所述主站点进行周期性的异步同步,以实现数据在数据库内的最终一致性;

[0049] 以上对于写入数据的缓存设计,都基于各节点对于建立节点所需服务器的成本考量为主,因此其随机情况强,优选地所述路由节点需要对分布式系统各节点的进行有效的负载监控和考察;

[0050] 而针对随机序列并发式的数据写入,所述路由节点虽然会对写请求事件赋予时间戳,然而偶然情况由于并发时刻太接近,或者并发量太大,存在对同一个键key进行写入时,由分布式上多于一个所述主站点对多于一个缓存位置创建缓存空间,并执行了对同一个键

key的不同的值value的写入;由此,对于同一个键key,则可能存在于对应的两个值value,保存在两个数据块内,并生成了两条键值对<key-value>的记录索引;

[0051] 进一步的,通过时间戳的验证,根据分布式系统内订立一验证规则,明确最终保存键值对<key-value>的数据块的地址;这里所指的数据块存储地址,可能包括在分布式系统的某节点的内存、缓存、或者硬盘上;通过搜索目标键key存在的数据块位置,确认目标键key唯一的合法的的键值对<key-value>的数据块,并将其余包含目标键key的数据块中的键值对<key-value>清除,保证了键值对<key-value>最终的合法数据块,从而保证了键值对<key-value>的最终一致性;

[0052] 进一步的,通过区块链的运作,保证最终的验证结果都由链上所有具备资格的分布式系统内的节点所验证通过;区块链对于数据的不可篡改性以及强一致性的特点,适合用于对验证后的数据作为信任背书;

[0053] 进一步的,本实施例中,只对键值对<key-value>的键key值,以及地址块的地址值进行绑定和打包操作;记录这两项信息所需要的储存空间,大大小于储存完整键值对<key-value>的储存空间,不仅有利于对区块信息进行打包时消耗的系统性能,也大大提高了全链验证的效率;

[0054] 进一步的,在大部分情况下,对数据库的读/写操作比例一般是读请求占大多数;所述路由站点优选地根据非关系型数据库的实际需求情况,调整选出的所述主站点数量比例,平衡读/写操作的负载程度;

[0055] 进一步的,通过所述主站点对数据块进行三个状态的写入状态进行标识,在数据块内的键值对<key-value>存在可疑异常或者正在清除进行时,拒绝对数据块的写入或读取操作,将异常情况进行一定程度的阻断,等待分布式系统内再次完成一致性验证后,再重新释放异常数据块的读/写功能,保证了非关系型数据库在有效的阻截机制

[0056] 下进行随机序列的读/写操作。

[0057] 实施例二:

[0058] 本实施例应当理解为至少包含前述任意一个实施例的全部特征,并在其基础上进一步改进;

[0059] 由于非关系型数据库的数据并非严格按照关系模型来组织数据的数据库的,并且某一类型键值对<key-value>会被大量请求读/写,技术上被称为高热度的键值对<key-value>;因此可针对该类型高热度的键值对<key-value>,所述控制方法给予更高的关注程度;而一些被低频请求读/写的键值对<key-value>,同样由于处理低热度的状态,其一致性能够较强地保持,所述控制方法亦针对性地作为关注度的调整;

[0060] 所述分布式系统对一定固定时间段内,例如一日或12小时内,所有执行过读/写的键key进行统计记录,获得一个键key在此统计周期内的执行次数排行列表;对所述排行列表内,执行次数按数量排序在前20%键key作为高热度键值对<key-value>;优选地将所述高热度的键key的存储位置相对固定于一个或几个数据块中,并将存储有所述高热度键值对<key-value>的数据块常规化地建立于高速缓存内,包括高速的随机存取存储器RAM或者高速磁盘阵列内,使对所述高热度键值对<key-value>的读/写操作都能够在高速缓存内进行;

[0061] 进一步的,对所述高热度键值对<key-value>的每次写入记录除了赋予所述时间

戳，还在每次更新时，对所述高热度键值对<key-value>建立版本号，并将所述时间戳与所述版本号一一对应；在对所述高热度键值对<key-value>进行一致性审查时，则除了可以根据键值对<key-value>的最后一条写入记录，判定其最终合法值，而且可以根据所述时间戳与所述版本号的关系，对所述键值对<key-value>进行版本回溯，找出从某版本号开始时所述键值对<key-value>被一致性验证后的值作为起点，从新对所述键值对<key-value>进行合法性写入操作，并再次作出一致性验证，最大限度重建所述键值对<key-value>在上一次一致性验证后的所有写操作；

[0062] 进一步的，所述版本号信息亦随所述键索引目录进行全链验证后，写入所述索引主链的区块内；

[0063] 进一步的，在一个或以上的监察周期后，其中没有再出现在高热度排序中的所述键值对<key-value>则列为过期键值对<key-value>，清除其高热度属性，并将其回归到普通键值对<key-value>的处理流程。

[0064] 实施例三：

[0065] 本实施例应当理解为至少包含前述任意一个实施例的全部特征，并在其基础上进一步改进：

[0066] 在对所述高热度键值对<key-value>安排存储于具有读/写速度性能更强的高速缓存位置后，其读/写速度会适应于每日上数十万次的读/写操作，但若存储了所述高热度键值对<key-value>的节点，或者节点内的存储部件，例如内存、硬盘发生异常故障，出现宕机或者数据库崩溃，则在短时间内会造成数据的堵塞或者数据灾难；因此本实施例针对所述高热度键值对<key-value>在高速缓存内的存储进行优化；

[0067] 对于存储了所述高热度键值对<key-value>的数据块，由所述主站点安排次一级读/写性能缓存进行镜像式写入，该部分负责镜像写入的数据块列为镜像数据块；所述镜像数据块必须存储在与原数据块不同的物理节点上，并且所述镜像数据块通过所述分布式系统确认其合法性，并且所述镜像数据块与原数据块的映射关系由所述路由节点进行记录；

[0068] 所述镜像数据块根据实际的性能级别，优选地对原数据块进行强一致性的同步写入，即在每一个单独的时刻，所述镜像数据块与原数据块都具有同样的键值对<key-value>存储在内部；或者，所述所述镜像数据块与原数据块进行异步写入，在每隔一段时间，例如10分钟或者20分钟后，将原数据块的数据全部进行镜像写入到所述镜像数据块上；

[0069] 进一步的，在原数据块发生异常，导致异常的读/写操作响应时，由所述主站点将原数据块地址的写入状态设置为拒绝写入状态；

[0070] 进一步的，所述主站点向所述路由站点查询原数据块是否具有所述镜像数据块，若存在所述镜像数据块，则所述主站点反馈到所述路由站点，将所有指向原数据块地址的读/写操作，全部指向所述镜像数据块，并对所述镜像数据块内的键值对<key-value>进行下一级备份处理，例如调入关系型数据库内作可断电型的数据备份；

[0071] 通过上述技术方案，对随机序列的非关系型数据库的读/写操作的一致性作出周期性的验证，并由大量实施读/写节点参与验证的过程并为验证后的一致性作出背书。

[0072] 在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述或记载的部分，可以参见其它实施例的相关描述。

[0073] 虽然上面已经参考各种实施例描述了本发明，但是应当理解，在不脱离本发明的

范围的情况下，可以进行许多改变和修改。也就是说上面讨论的方法，系统和设备是示例。各种配置可以适当地省略，替换或添加各种过程或组件。例如，在替代配置中，可以以与所描述的顺序不同的顺序执行方法，和/或可以添加，省略和/或组合各种部件。而且，关于某些配置描述的特征可以以各种其他配置组合，如可以以类似的方式组合配置的不同方面和元素。此外，随着技术发展其中的元素可以更新，即许多元素是示例，并不限制本公开或权利要求的范围。

[0074] 在说明书中给出了具体细节以提供对包括实现的示例性配置的透彻理解。然而，可以在没有这些具体细节的情况下实践配置例如，已经示出了众所周知的电路，过程，算法，结构和技术而没有不必要的细节，以避免模糊配置。该描述仅提供示例配置，并且不限制权利要求的范围，适用性或配置。相反，前面对配置的描述将为本领域技术人员提供用于实现所描述的技术的使能描述。在不脱离本公开的精神或范围的情况下，可以对元件的功能和布置进行各种改变。

[0075] 综上，其旨在上述详细描述被认为是例示性的而非限制性的，并且应当理解，以上这些实施例应理解为仅用于说明本发明而不用于限制本发明的保护范围。在阅读了本发明的记载的内容之后，技术人员可以对本发明作各种改动或修改，这些等效变化和修饰同样落入本发明权利要求所限定的范围。

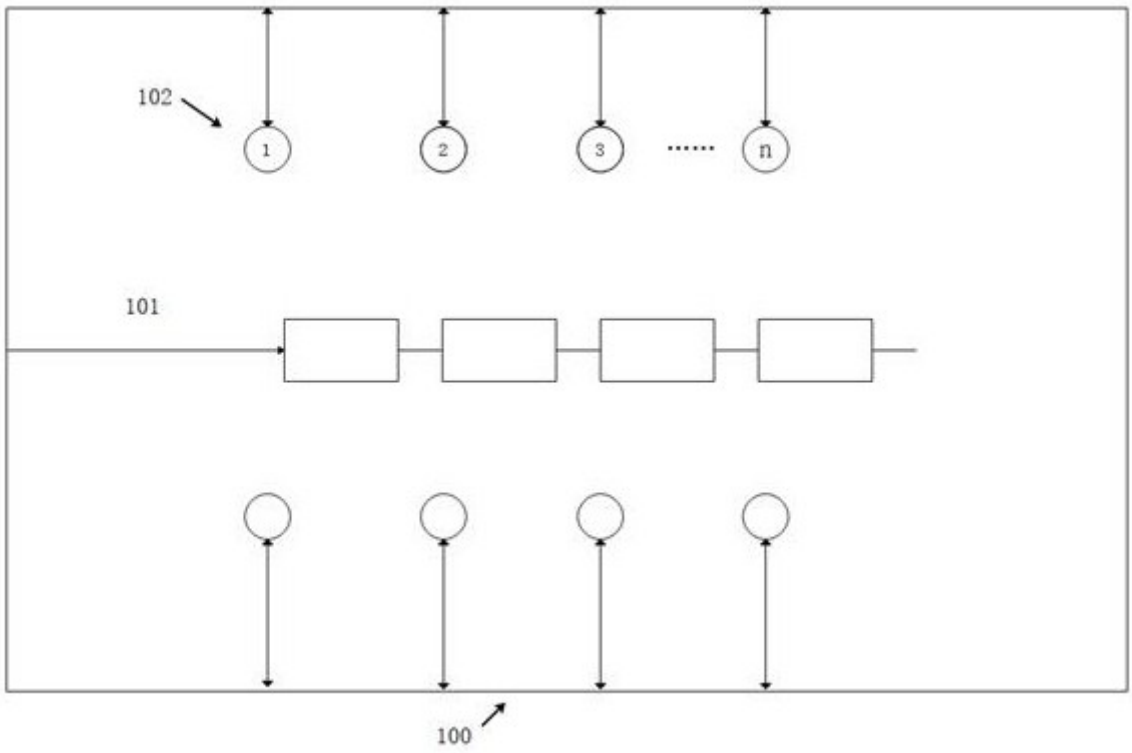

102
1
2
3
n
101
100

图1

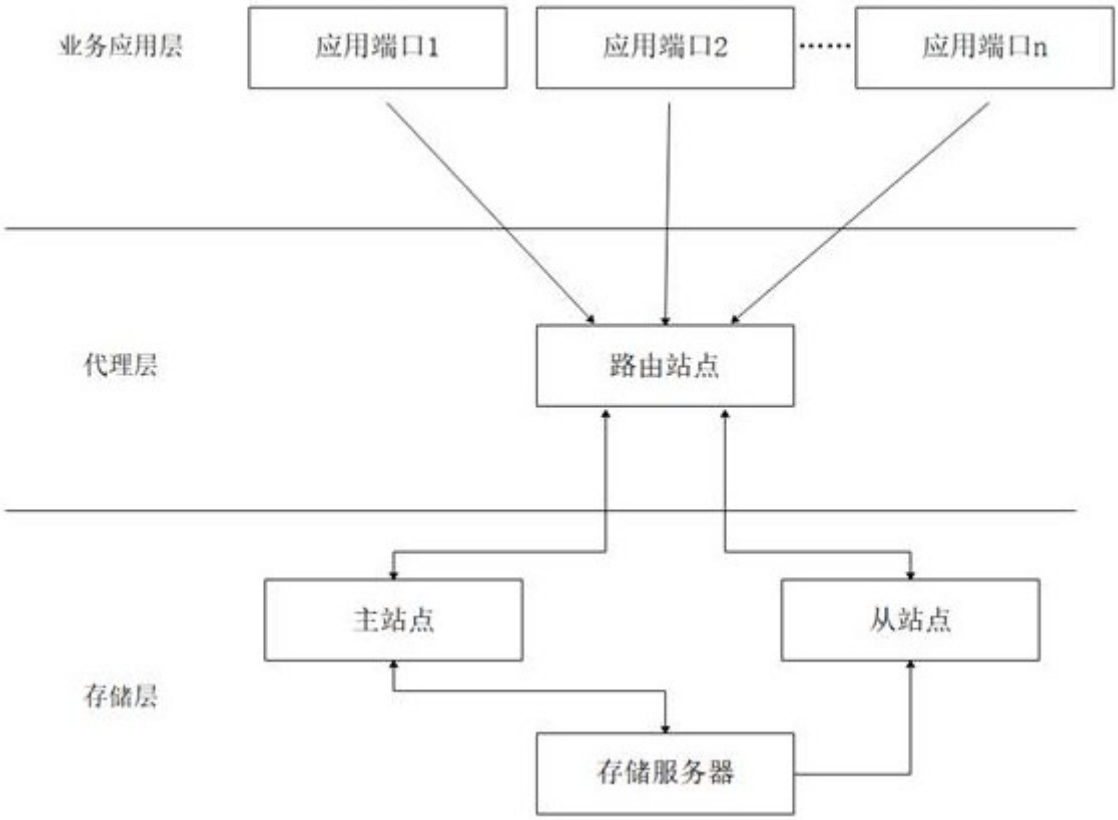

业务应用层
应用端口1
应用端口2
应用端口n
代理层
路由站点
主站点
从站点
存储层
存储服务器

图2

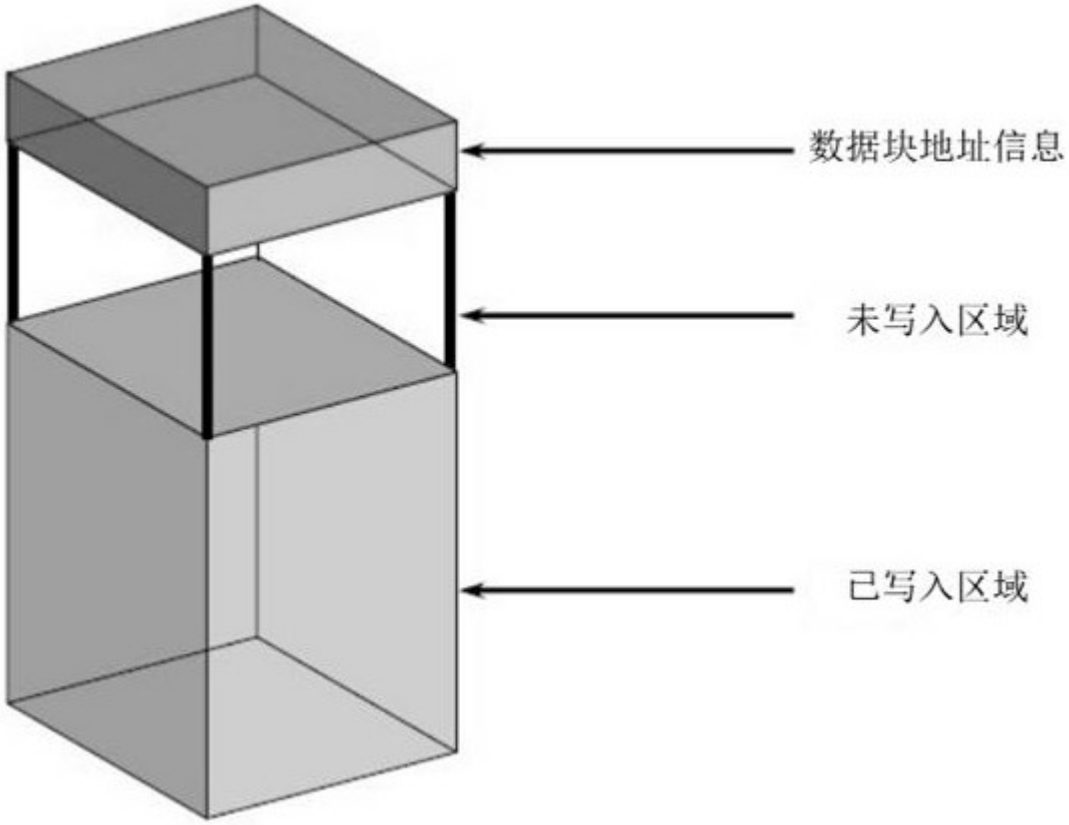


图3

图4

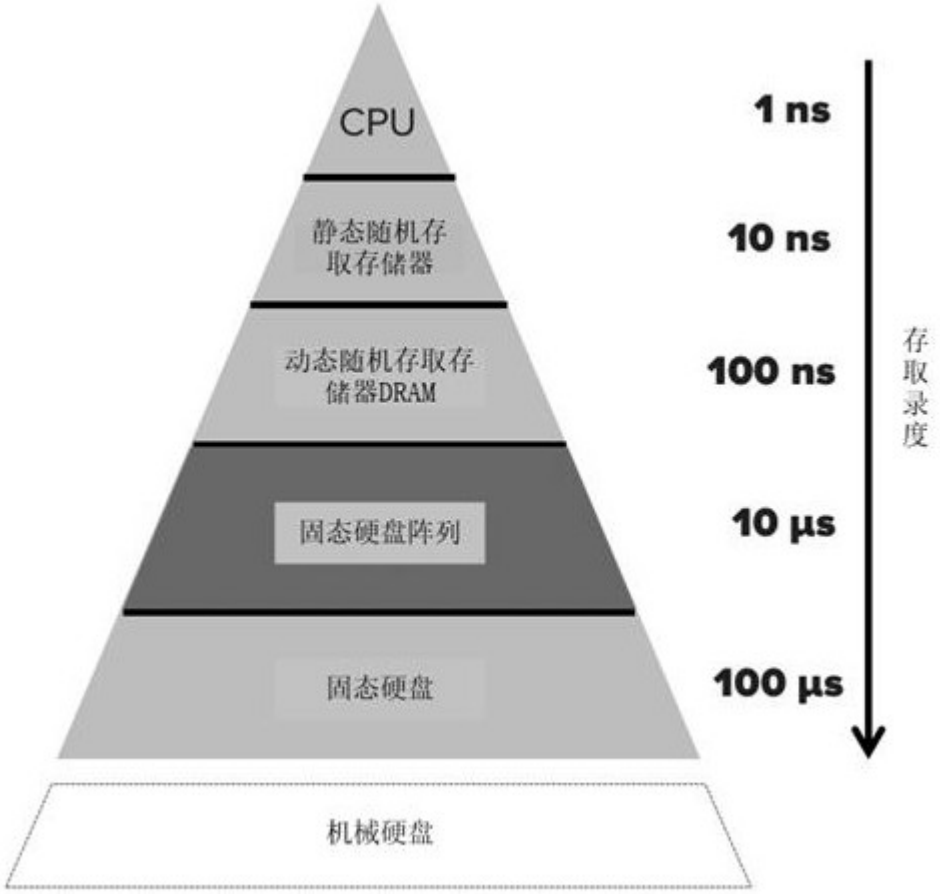

CPU
静态随机存取存储器
动态随机存取存储器DRAM
固态硬盘阵列
固态硬盘
机械硬盘
1 ns
10 ns
100 ns
10 μs
100 μs
存取录度

图5