



(12) 发明专利

(10) 授权公告号 CN 113312349 B

(45) 授权公告日 2024. 10. 11

(21) 申请号 202110439886.X

G06F 16/2455 (2019. 01)

(22) 申请日 2021. 04. 23

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 108846024 A, 2018. 11. 20

申请公布号 CN 113312349 A

CN 108108252 A, 2018. 06. 01

(43) 申请公布日 2021. 08. 27

审查员 乔晋

(73) 专利权人 北京淇瑀信息科技有限公司

地址 100012 北京市朝阳区双营路11号院3

号楼2层4单元207

(72) 发明人 叶峰 陈朝

(74) 专利代理机构 北京清诚知识产权代理有限公司

11691

专利代理师 喻颖

(51) Int. Cl.

G06F 16/22 (2019. 01)

G06F 16/23 (2019. 01)

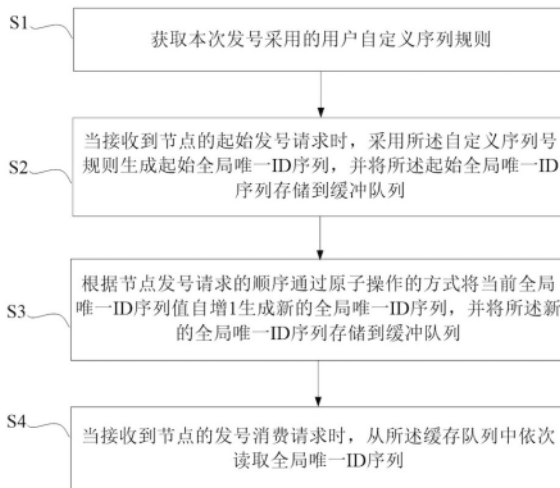
权利要求书2页 说明书8页 附图4页

(54) 发明名称

一种基于原子操作的分布式发号方法、装置及电子设备

(57) 摘要

本发明公开了一种基于原子操作的分布式发号方法、装置及电子设备,所述方法包括:获取本次发号采用的用户自定义序列规则;当接收到节点的起始发号请求时,采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;当接收到节点的发号消费请求时,从所述缓存队列中依次读取全局唯一ID序列。本发明通过嵌入式的ID组件在分布式系统中完成全局唯一流水号的发号,同时本发明支持自定义格式的业务流水号场景,并能有效减小流水号的存储空间。



1. 一种基于原子操作的分布式发号方法,其特征在于,所述方法包括:
通过嵌入式方式在分布式系统的各个节点中生成全局唯一ID组件,每个节点对应一个ID组件用以生成该节点请求的全局唯一ID序列;
ID组件获取本次发业务流水号采用的用户自定义序列规则;
当接收到节点的起始发号请求时, ID组件采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;
ID组件根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;
当ID组件接收到节点的发号消费请求时,从所述缓冲队列中依次读取全局唯一ID序列。
2. 根据权利要求1所述的方法,其特征在于,当所述缓冲队列达到存储阈值时,清空缓冲队列,并根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列。
3. 根据权利要求2所述的方法,其特征在于,所述获取本次发号采用的用户自定义序列规则包括:
接收用户输入的关键词;
根据所述关键词匹配用户自定义序列库,显示匹配到的至少一个用户自定义序列规则;
根据用户对所述至少一个用户自定义序列规则的选取操作确定本次发号采用的用户自定义序列规则。
4. 根据权利要求3所述的方法,其特征在于,所述全局唯一ID序列的取值范围采用分片原理生成。
5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:
将从缓存队列读取的全局唯一ID序列存入数据库。
6. 根据权利要求1所述的方法,其特征在于,所述自定义序列号规则的结构包括序列号名称、开始索引、序列最小值、序列最大值、递增步长、前一个索引、当前索引和序列号格式。
7. 一种基于原子操作的分布式发号装置,其特征在于,所述装置包括:
生成模块,用于通过嵌入式方式在分布式系统的各个节点中生成全局唯一ID组件,每个节点对应一个ID组件用以生成该节点请求的全局唯一ID序列;
ID组件模块,用于获取本次发号采用的用户自定义序列规则;当接收到节点的起始发业务流水号请求时,采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;当接收到节点的发号消费请求时,从所述缓冲队列中依次读取全局唯一ID序列。
8. 根据权利要求7所述的装置,其特征在于,所述ID组件模块还用于当所述缓冲队列达到存储阈值时,清空缓冲队列,并根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列。
9. 根据权利要求8所述的装置,其特征在于,所述ID组件模块包括:

接收模块,用于接收用户输入的关键词;

匹配模块,用于根据所述关键词匹配用户自定义序列库,显示匹配到的至少一个用户自定义序列规则;

确定模块,用于根据用户对所述至少一个用户自定义序列规则的选取操作确定本次发号采用的用户自定义序列规则。

10.根据权利要求9所述的装置,其特征在于,所述全局唯一ID序列的取值范围采用分片原理生成。

11.根据权利要求7所述的装置,其特征在于,所述装置还包括:

存储模块,用于将从缓存队列读取的全局唯一ID序列存入数据库。

12.根据权利要求7所述的装置,其特征在于,所述自定义序列号规则的结构包括序列号名称、开始索引、序列最小值、序列最大值、递增步长、前一个索引、当前索引和序列号格式。

13.一种电子设备,包括:

处理器;以及

存储计算机可执行指令的存储器,所述计算机可执行指令在被执行时使所述处理器执行根据权利要求1-6中任一项所述的方法。

14.一种计算机可读存储介质,其中,所述计算机可读存储介质存储一个或多个程序,当所述一个或多个程序被处理器执行时,实现权利要求1-6中任一项所述的方法。

一种基于原子操作的分布式发号方法、装置及电子设备

技术领域

[0001] 本发明涉及计算机网络技术领域,具体而言,涉及一种基于原子操作的分布式发号方法、装置、电子设备及计算机可读介质。

背景技术

[0002] 在使用数据库时,表的主键经常会使用数据库的自增(auto_increment)来产生。比如mysql的自增主键,MongoDB中的ObjectId等都是数据库用来生成全局唯一的流水号ID,表示唯一记录的方式。但当数据量越来越大时,需要对数据分库分表。由于不同的数据库之间的全局唯一的流水号ID可能是一样的,导致分库分表后数据库自带的自增ID无法满足ID唯一的需求。这就需要有一个能够在分布式的环境中生产全局唯一的流水号ID的发号器。

发明内容

[0003] 本发明旨在解决现有全局唯一流水号生成技术无法满足数据量大时分库分表的全局唯一需求的技术问题。

[0004] 为了解决上述技术问题,本发明第一方面提出一种基于原子操作的分布式发号方法,所述方法包括:

[0005] 获取本次发号采用的用户自定义序列规则;

[0006] 当接收到节点的起始发号请求时,采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;

[0007] 根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;

[0008] 当接收到节点的发号消费请求时,从所述缓存队列中依次读取全局唯一ID序列。

[0009] 根据本发明一种优选实施方式,当所述缓冲队列达到存储阈值时,清空缓冲队列,并根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列。

[0010] 根据本发明一种优选实施方式,所述获取本次发号采用的用户自定义序列规则包括:

[0011] 接收用户输入的关键词;

[0012] 根据所述关键词匹配用户自定义序列库,显示匹配到的至少一个用户自定义序列规则;

[0013] 根据用户对所述至少一个用户自定义序列规则的选取操作确定本次发号采用的用户自定义序列规则。

[0014] 根据本发明一种优选实施方式,所述全局唯一ID序列的取值范围采用分片原理生成。

[0015] 根据本发明一种优选实施方式,所述方法还包括:

- [0016] 将从缓存队列读取的全局唯一ID序列存入数据库。
- [0017] 根据本发明一种优选实施方式,所述自定义序列号规则的结构包括序列号名称、开始索引、序列最小值、序列最大值、递增步长、前一个索引、当前索引和序列号格式。
- [0018] 为了解决上述技术问题,本发明第二方面提供一种基于原子操作的分布式发号装置,所述装置包括:
- [0019] 获取模块,用于获取本次发号采用的用户自定义序列规则;
- [0020] 第一生成模块,用于当接收到节点的起始发号请求时,采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;
- [0021] 第二生成模块,用于根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;
- [0022] 读取模块,用于当接收到节点的发号消费请求时,从所述缓存队列中依次读取全局唯一ID序列。
- [0023] 根据本发明一种优选实施方式,所述第二生成模块还用于当所述缓冲队列达到存储阈值时,清空缓冲队列,并根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列。
- [0024] 根据本发明一种优选实施方式,所述获取模块包括:
- [0025] 接收模块,用于接收用户输入的关键词;
- [0026] 匹配模块,用于根据所述关键词匹配用户自定义序列库,显示匹配到的至少一个用户自定义序列规则;
- [0027] 确定模块,用于根据用户对所述至少一个用户自定义序列规则的选取操作确定本次发号采用的用户自定义序列规则。
- [0028] 根据本发明一种优选实施方式,所述全局唯一ID序列的取值范围采用分片原理生成。
- [0029] 根据本发明一种优选实施方式,所述装置还包括:
- [0030] 存储模块,用于将从缓存队列读取的全局唯一ID序列存入数据库。
- [0031] 根据本发明一种优选实施方式,所述自定义序列号规则的结构包括序列号名称、开始索引、序列最小值、序列最大值、递增步长、前一个索引、当前索引和序列号格式。
- [0032] 为解决上述技术问题,本发明第三方面提供一种电子设备,包括:
- [0033] 处理器;以及
- [0034] 存储计算机可执行指令的存储器,所述计算机可执行指令在被执行时使所述处理器执行上述的方法。
- [0035] 为解决上述技术问题,本发明第四方面提供一种计算机可读存储介质,其中,所述计算机可读存储介质存储一个或多个程序,当所述一个或多个程序被处理器执行时,实现上述方法。
- [0036] 本发明通过嵌入式模式在分布式系统中生成全局唯一ID组件,各个ID组件可以采用用户自定义序列规则生成起始的全局唯一ID序列,通过原子操作的方式将当前全局唯一ID序列值自增1,达到自增序列的目的。此外,本发明采用生产/消费模式将各个ID组件预先

生成的全局唯一ID序列存储到缓冲队列中,调用时直接从内存获取ID序列,能有效减小流水号的存储空间。本发明通过嵌入式的ID组件在分布式系统中完成全局唯一流水号的发号,同时本发明支持自定义格式的业务流水号场景,并能有效减小流水号的存储空间。

附图说明

[0037] 为了使本发明所解决的技术问题、采用的技术手段及取得的技术效果更加清楚,下面将参照附图详细描述本发明的具体实施例。但需声明的是,下面描述的附图仅仅是本发明的示例性实施例的附图,对于本领域的技术人员来讲,在不付出创造性劳动的前提下,可以根据这些附图获得其他实施例的附图。

[0038] 图1是本发明一种基于原子操作的分布式发号方法的流程示意图;

[0039] 图2是本发明实施基于原子操作的分布式发号的示意图;

[0040] 图3是本发明一种分布式发号装置的结构框架示意图;

[0041] 图4是根据本发明的一种电子设备的示例性实施例的结构框图;

[0042] 图5是本发明一种计算机可读介质实施例的示意图。

具体实施方式

[0043] 现在将参考附图来更加全面地描述本发明的示例性实施例,虽然各示例性实施例能够以多种具体的方式实施,但不应理解为本发明仅限于在此阐述的实施例。相反,提供这些示例性实施例是为了使本发明的内容更加完整,更加便于将发明构思全面地传达给本领域的技术人员。

[0044] 在符合本发明的技术构思的前提下,在某个特定的实施例中描述的结构、性能、效果或者其他特征可以以任何合适的方式结合到一个或更多其他的实施例中。

[0045] 在对于具体实施例的介绍过程中,对结构、性能、效果或者其他特征的细节描述是为了使本领域的技术人员对实施例能够充分理解。但是,并不排除本领域技术人员可以在特定情况下,以不含有上述结构、性能、效果或者其他特征的技术方案来实施本发明。

[0046] 附图中的流程图仅是一种示例性的流程演示,不代表本发明的方案中必须包括流程图中的所有内容、操作和步骤,也不代表必须按照图中所显示的的顺序执行。例如,流程图中的操作/步骤可以分解,有的操作/步骤可以合并或部分合并,等等,在不脱离本发明的发明主旨的情况下,流程图中显示的执行顺序可以根据实际情况改变。

[0047] 附图中的框图一般表示的是功能实体,并不一定必然与物理上独立的实体相对应。即,可以采用软件形式来实现这些功能实体,或在一个或多个硬件模块或集成电路中实现这些功能实体,或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

[0048] 各附图中相同的附图标记表示相同或类似的元件、组件或部分,因而下文中可能省略了对相同或类似的元件、组件或部分的重复描述。还应理解,虽然本文中可能使用第一、第二、第三等表示编号的定语来描述各种器件、元件、组件或部分,但是这些器件、元件、组件或部分不应受这些定语的限制。也就是说,这些定语仅是用来将一者与另一者区分。例如,第一器件亦可称为第二器件,但不偏离本发明实质的技术方案。此外,术语“和/或”、“及/或”是指包括所列出项目中的任一个或多个的所有组合。

[0049] 请参阅图1,图1是本发明提供的一种基于原子操作的分布式发号方法的流程图。该方法以嵌入式模式生成ID组件,通过ID组件完成不同业务的流水号发布,所述ID组件可以以插件的方式集成到分布式系统的各个节点中,具有即插即用,零成本接入的优点。如图1所示,所述方法包括:

[0050] S1、获取本次发号采用的用户自定义序列规则;

[0051] 本发明支持用户自定义格式的业务流水号场景,用户可以对不同的业务设置不同的业务流水号格式,这样,通过业务流水号格式就可以区分各个业务类型。在一种示例中,可以通过前端web页面,比如采用spring boot +thymeleaf的web工程来管理用户自定义序列规则,并根据用户选择获得本次发号的用户自定义序列规则。因此,所述获取本次发号采用的用户自定义序列规则包括:

[0052] S11、接收用户输入的关键词;

[0053] 其中,所述关键词可以是序列号名称、开始索引、当前索引中的任一个。

[0054] S12、根据所述关键词匹配用户自定义序列库,显示匹配到的至少一个用户自定义序列规则;

[0055] 本发明中可以采用模糊匹配,也可以采用精确匹配。

[0056] S13、根据用户对所述至少一个用户自定义序列规则的选取操作确定本次发号采用的用户自定义序列规则。

[0057] 所述选取操作可以是用户对本次发号采用的用户自定义序列规则的点击操作。

[0058] 此外,在本步骤之前,还可以通过前端web页面接收用户输入的正则表达式,根据所述正则表达式及用户自定义序列规则的结构生成用户自定义序列规则,各个用户自定义序列规则集合生成用户自定义序列库。其中,正则表达式是对字符串(包括普通字符(例如,a到z之间的字母)和特殊字符(称为“元字符”))操作的一种逻辑公式,也就是将预设的一些特定字符、及这些特定字符的组合,组成一个“规则字符串”,这个“规则字符串”用来表达对字符串的一种过滤逻辑。本发明中通过正则表达式描述在生成全局唯一ID序列时要匹配的一个或多个规则。示例性的,可以采用正则表达式%s描述用户自定义序列规则,其中,%表示在全局唯一ID序列的所用行中处理,s表示查找功能,正则表达式%s表示在所有行中能够查找到指定字符。例如ABC%s代表在所有行中能够查找到指定字符ABC。所述用户自定义序列规则的结构包括序列号名称、开始索引、序列最小值、序列最大值、递增步长、前一个索引、当前索引和序列号格式,其中,所述序列号格式用于指定序列号包含的数字、字符位置及个数。在具体生成用户自定义序列规则时,可以为用户自定义序列规则的每个结构设置一个对应的标识符,通过接收用户输入的与该标识符对应的正则表达式%s完成对各个结构的自定义,将自定义后的各个结构拼接即可得到完整的用户自定义序列规则。

[0059] S2、当接收到节点的起始发号请求时,采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;

[0060] 在分布式系统中,数据和程序可以分散到不同的节点中,所有节点共同运行实现一系列业务功能。本发明实施例中,当分布式系统包含多个业务模块时,各个业务节点通过发送发号请求通知ID组件来生产全局唯一ID序列,并通过发送发号消费请求从缓冲队列中读取对应的全局唯一ID序列。其中,ID组件通过嵌入式方式采用spring boot starter的形式来开发,所述ID组件可以提供接口API,这样可以以插件的形式集成到分布式系统中。

[0061] 如图2所示,分布式系统包含节点1、2...N,本发明的ID组件可以通过嵌入式的方式插入各个节点中,每个节点可对应一个ID组件用以生成该节点请求的全局唯一ID序列。对于应用层的一个新业务(比如订单号业务、日志号业务等),在业务开始时,通过前端页面接收本次发号采用的用户自定义序列规则,该业务的启动节点,如节点2会向其对应的ID组件2发送起始发号请求,当接收到节点2的起始发号请求时,ID组件2采用本次发号的自定义序列规则生成起始全局唯一ID序列。此外,本发明为了节省全局唯一ID序列占有存储空间大,影响数据库插入和查询性能的问题,采用生产/消费模式将各个ID组件预先生成的全局唯一ID序列存储到缓冲队列中,调用时直接从内存获取ID序列,有效减小流水号的存储空间。因此,在生成起始全局唯一ID序列后,ID组件2将所述起始全局唯一ID序列存储到缓冲队列中,完成新业务中起始全局唯一ID序列的生产及存储。

[0062] 如图2所示,所述缓冲队列可以是多个,优选为阻塞队列,通过阻塞队列实现了一个用于消费者(多个)和生产者(多个)交换产品的中介,生产者线程(即ID组件)在队列满时阻塞,消费者线程(各个节点)在队列空时阻塞。

[0063] S3、根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;

[0064] 为了保证分布式系统中各个ID组件后续生成的全局唯一ID序列的唯一性,本发明的各个ID组件根据节点发号请求的顺序采用原子操作(atomic operation)的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,达到自增序列的目的。其中,原子操作指不可被中断的一个或一系列操作。其可以在处理器中实现、也可以在Java中实现。本发明通过Java原子操作类AtomicLong,以原子方式将当前全局唯一ID序列值自增1,并返回自增后的值,得到新的全局唯一ID序列,再以生产/消费的方式将所述新的全局唯一ID序列存储到缓冲队列。

[0065] 其中,每个缓冲队列都设有存储阈值,当所述缓冲队列达到存储阈值时,即缓冲队列阻塞时,清空缓冲队列,重新生成新的全局唯一ID序列,即根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列。

[0066] S4、当接收到节点的发号消费请求时,从所述缓存队列中依次读取全局唯一ID序列。

[0067] 当某个具体业务在节点执行完成后,该节点会向其对应的ID组件发送发号消费请求,以获取该业务的全局唯一ID序列作为业务流水号。本发明采用先进先出的方式从缓存队列中依次读取全局唯一ID序列。

[0068] 进一步的,如图2,本发明还可以将从缓存队列读取的全局唯一ID序列存入数据库,方便后续查询。所述数据库可以采用mysql或者redis。本发明中,所述全局唯一ID序列的取值范围采用分片原理生成。即将存储全局唯一ID序列的数据库拆分,将其分散在不同的机器上。

[0069] 图3是本发明一种分布式发号装置的架构示意图,如图3所示,所述装置包括:

[0070] 获取模块31,用于获取本次发号采用的用户自定义序列规则;其中,所述自定义序列号规则的结构包括序列号名称、开始索引、序列最小值、序列最大值、递增步长、前一个索引、当前索引和序列号格式。

[0071] 第一生成模块32,用于当接收到节点的起始发号请求时,采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;

[0072] 第二生成模块33,用于根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;

[0073] 读取模块34,用于当接收到节点的发号消费请求时,从所述缓存队列中依次读取全局唯一ID序列。

[0074] 在一种具体实施方式中,所述第二生成模块33还用于当所述缓冲队列达到存储阈值时,清空缓冲队列,并根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列。

[0075] 所述获取模块31包括:

[0076] 接收模块,用于接收用户输入的关键词;

[0077] 匹配模块,用于根据所述关键词匹配用户自定义序列库,显示匹配到的至少一个用户自定义序列规则;

[0078] 确定模块,用于根据用户对所述至少一个用户自定义序列规则的选取操作确定本次发号采用的用户自定义序列规则。

[0079] 本发明中,所述全局唯一ID序列的取值范围采用分片原理生成。

[0080] 进一步的,所述装置还包括:

[0081] 存储模块35,用于将从缓存队列读取的全局唯一ID序列存入数据库。

[0082] 本领域技术人员可以理解,上述装置实施例中的各模块可以按照描述分布于装置中,也可以进行相应变化,分布于不同于上述实施例的一个或多个装置中。上述实施例的模块可以合并为一个模块,也可以进一步拆分成多个子模块。

[0083] 下面描述本发明的电子设备实施例,该电子设备可以视为对于上述本发明的方法和装置实施例的实体形式的实施方式。对于本发明电子设备实施例中描述的细节,应视为对于上述方法或装置实施例的补充;对于在本发明电子设备实施例中未披露的细节,可以参照上述方法或装置实施例来实现。

[0084] 图4是根据本发明的一种电子设备的示例性实施例的结构框图。图4显示的电子设备仅仅是一个示例,不对本发明实施例的功能和使用范围带来任何限制。

[0085] 如图4所示,该示例性实施例的电子设备400以通用数据处理设备的形式表现。电子设备400的组件可以包括但不限于:至少一个处理单元410、至少一个存储单元420、连接不同电子设备组件(包括存储单元420和处理单元410)的总线430、显示单元440等。

[0086] 其中,所述存储单元420存储有计算机可读程序,其可以是源程序或都只读程序的代码。所述程序可以被处理单元410执行,使得所述处理单元410执行本发明各种实施方式的步骤。例如,所述处理单元410可以执行如图1所示的步骤。

[0087] 所述存储单元420可以包括易失性存储单元形式的可读介质,例如随机存取存储单元(RAM)4201和/或高速缓存存储单元4202,还可以进一步包括只读存储单元(ROM)4203。所述存储单元420还可以包括具有一组(至少一个)程序模块4205的程序/实用工具4204,这样的程序模块4205包括但不限于:操作电子设备、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0088] 总线430可以为表示几类总线结构中的一种或多种,包括存储单元总线或者存储单元控制器、外围总线、图形加速端口、处理单元或者使用多种总线结构中的任意总线结构的局域总线。

[0089] 电子设备400也可以与一个或多个外部设备300(例如键盘、显示器、网络设备、蓝牙设备等)通信,使得用户能经由这些外部设备300与该电子设备400交互,和/或使得该电子设备400能与一个或多个其它数据处理设备(例如路由器、调制解调器等等)进行通信。这种通信可以通过输入/输出(I/O)接口450进行,还可以通过网络适配器460与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)进行。网络适配器460可以通过总线430与电子设备400的其它模块通信。应当明白,尽管图4中未示出,电子设备400中可使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID电子设备、磁带驱动器以及数据备份存储电子设备等。

[0090] 图5是本发明的一个计算机可读介质实施例的示意图。如图5所示,所述计算机程序可以存储于一个或多个计算机可读介质上。计算机可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以为但不限于电、磁、光、电磁、红外线、或半导体的电子设备、装置或器件,或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。当所述计算机程序被一个或多个数据处理设备执行时,使得该计算机可读介质能够实现本发明的上述方法,即:获取本次发号采用的用户自定义序列规则;当接收到节点的起始发号请求时,采用所述自定义序列号规则生成起始全局唯一ID序列,并将所述起始全局唯一ID序列存储到缓冲队列;根据节点发号请求的顺序通过原子操作的方式将当前全局唯一ID序列值自增1生成新的全局唯一ID序列,并将所述新的全局唯一ID序列存储到缓冲队列;当接收到节点的发号消费请求时,从所述缓存队列中依次读取全局唯一ID序列。

[0091] 通过以上的实施方式的描述,本领域的技术人员易于理解,本发明描述的示例性实施例可以通过软件实现,也可以通过软件结合必要的硬件的方式来实现。因此,根据本发明实施方式的技术方案可以以软件产品的形式体现出来,该软件产品可以存储在一个计算机可读的存储介质(可以是CD-ROM,U盘,移动硬盘等)中或网络上,包括若干指令以使得一台数据处理设备(可以是个人计算机、服务器、或者网络设备等)执行根据本发明的上述方法。

[0092] 所述计算机可读存储介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了可读程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。可读存储介质还可以是可读存储介质以外的任何可读介质,该可读介质可以发送、传播或者传输用于由指令执行电子设备、装置或者器件使用或者与其结合使用的程序。可读存储介质上包含的程序代码可以用任何适当的介质传输,包括但不限于无线、有线、光缆、RF等等,或者上述的任意合适的组合。

[0093] 可以以一种或多种程序设计语音的任意组合来编写用于执行本发明操作的程序代码,所述程序设计语音包括面向对象的设计语音—诸如Java、C++等,还包括常规的过程式程序设计语音—诸如“C”语音或类似的设计语音。程序代码可以完全地在用户

计算设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。在涉及远程计算设备的情形中,远程计算设备可以通过任意种类的网络,包括局域网(LAN)或广域网(WAN),连接到用户计算设备,或者,可以连接到外部计算设备(例如利用因特网服务提供商来通过因特网连接)。

[0094] 综上所述,本发明可以执行计算机程序的方法、装置、电子设备或计算机可读介质来实现。可以在实践中使用微处理器或者数字信号处理器(DSP)等通用数据处理设备来实现本发明的一些或者全部功能。

[0095] 以上所述的具体实施例,对本发明的目的、技术方案和有益效果进行了进一步详细说明,应理解的是,本发明不与任何特定计算机、虚拟装置或者电子设备固有相关,各种通用装置也可以实现本发明。以上所述仅为本发明的具体实施例而已,并不用于限制本发明,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

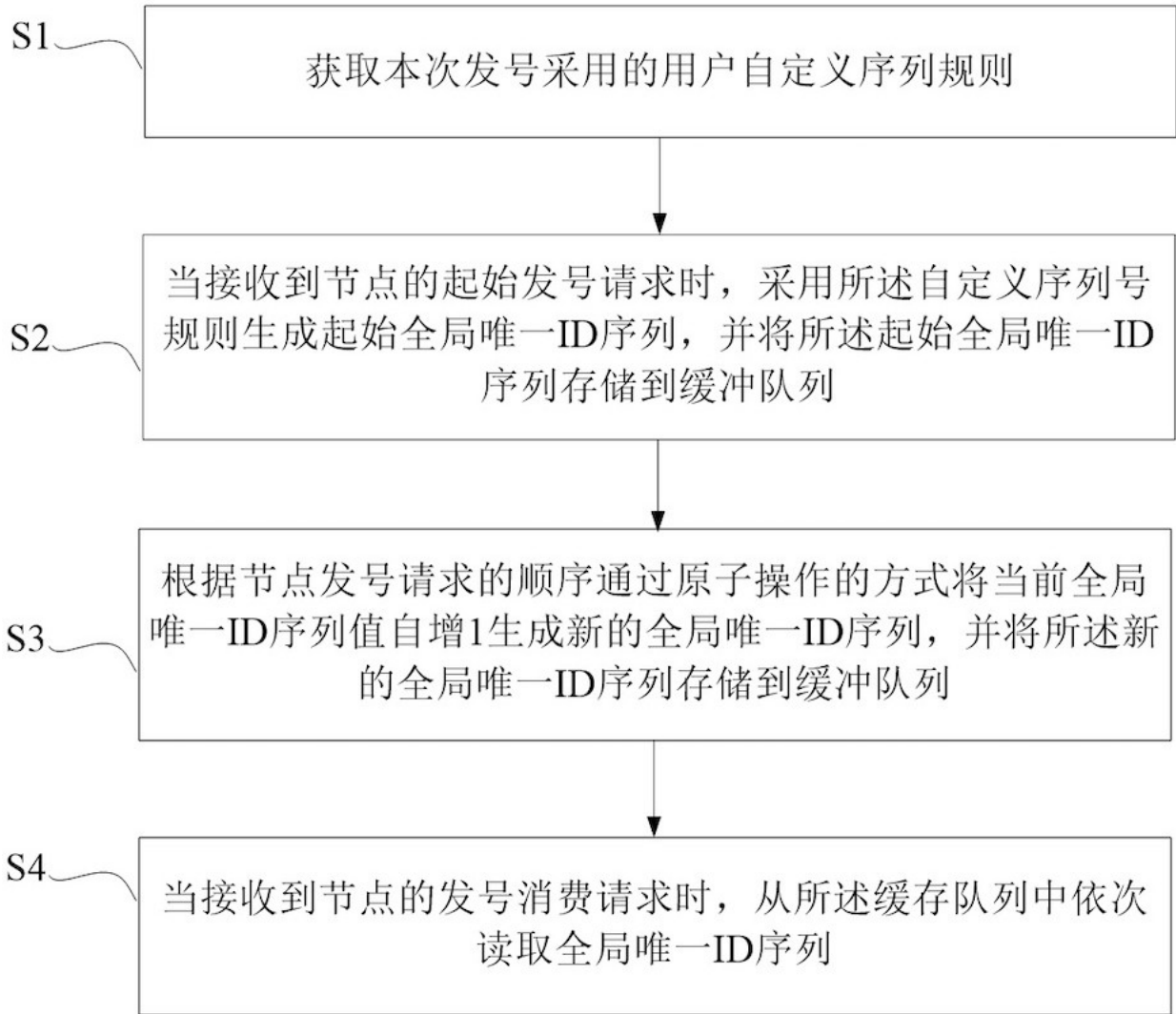


图 1

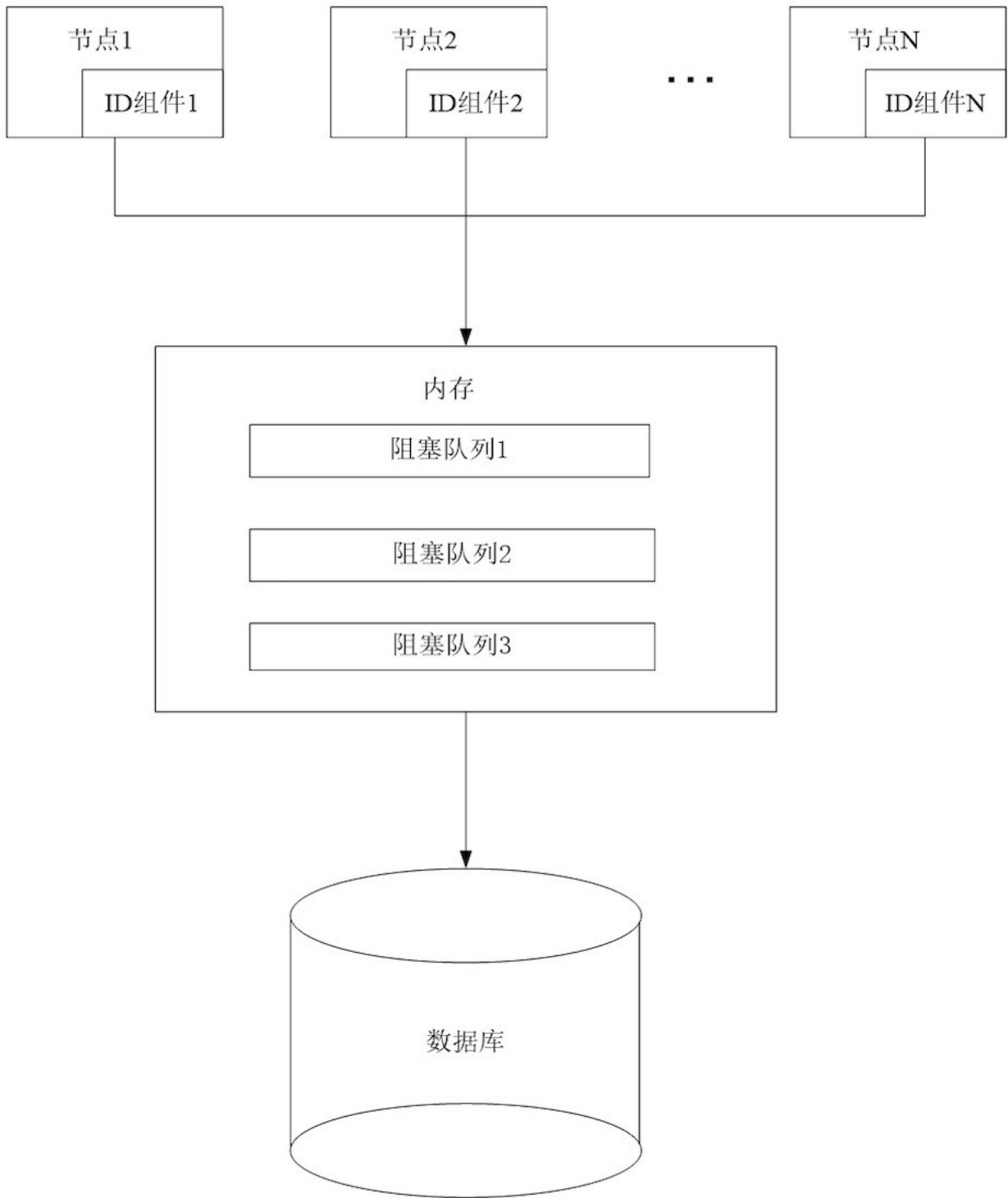


图 2

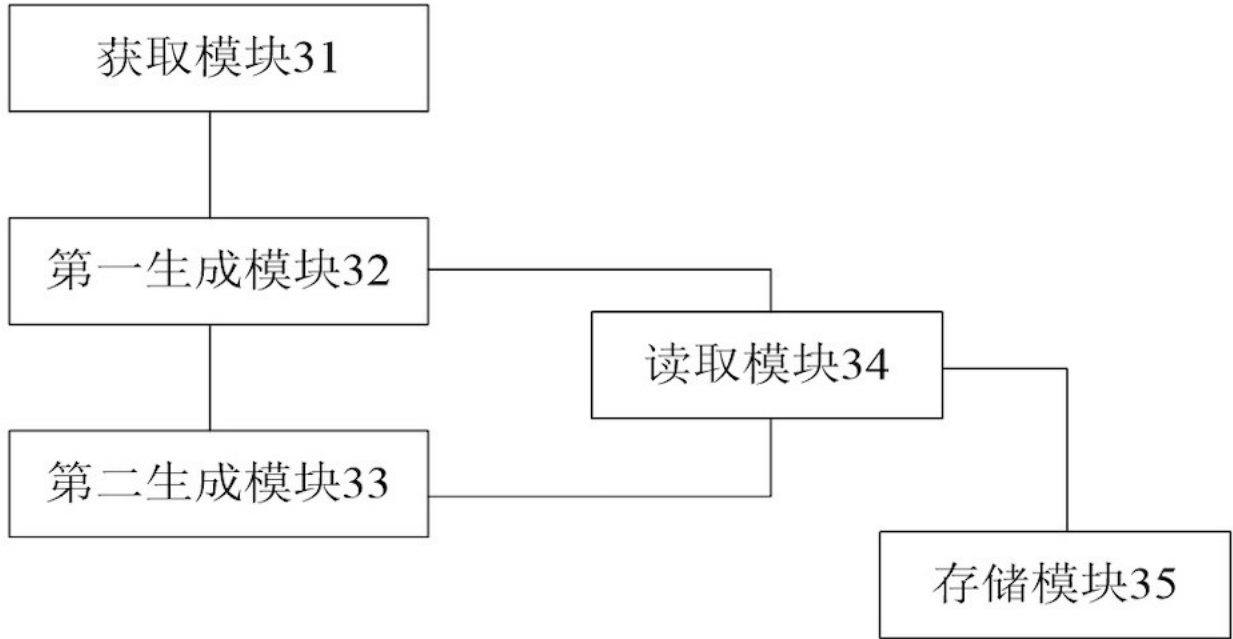


图 3

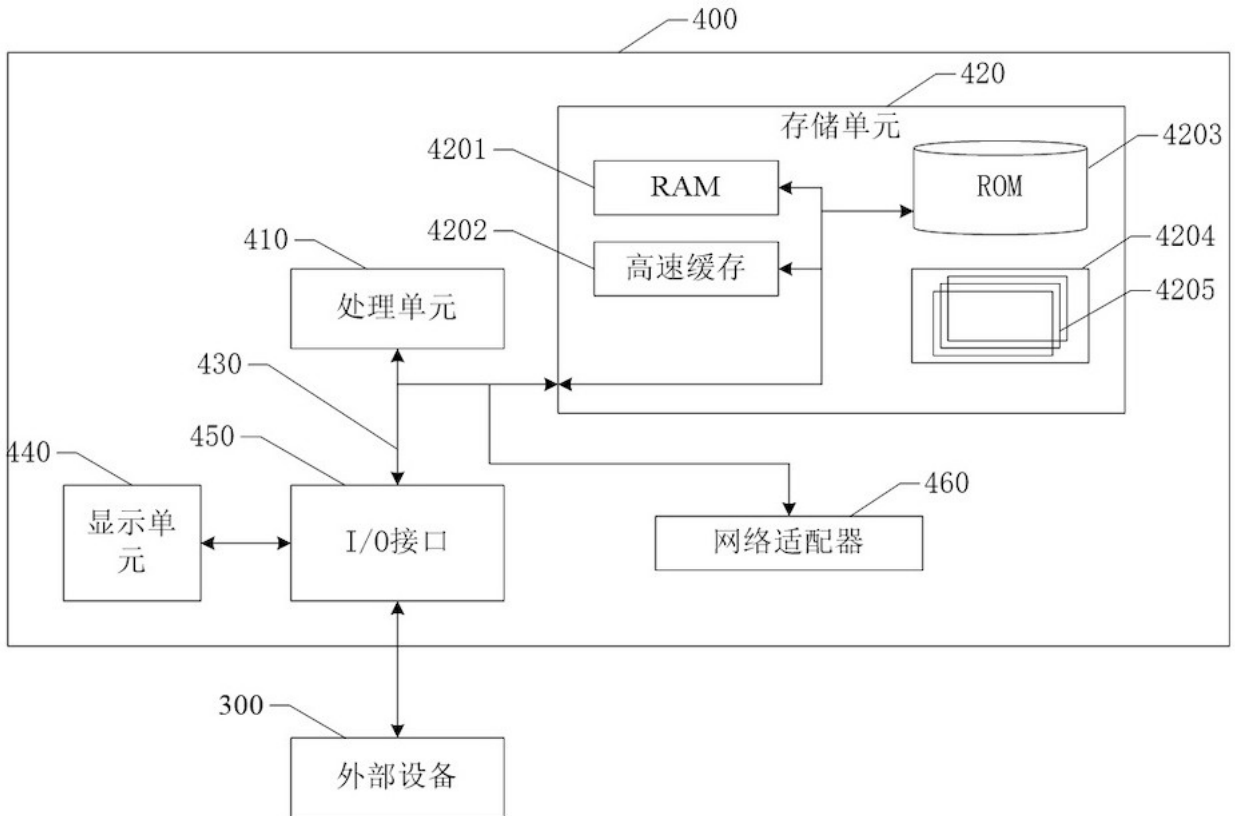


图 4

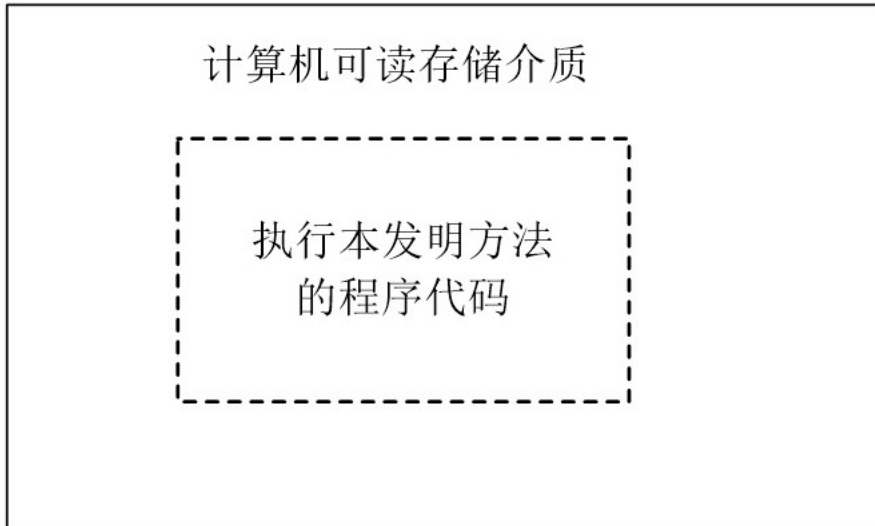


图 5