



(12)发明专利申请

(10)申请公布号 CN 109783126 A
(43)申请公布日 2019.05.21

(21)申请号 201910022807.8

(22)申请日 2019.01.10

(71)申请人 珠海金山网络游戏科技有限公司
地址 519000 广东省珠海市香洲区前岛环
路325号金山软件园3号楼
申请人 广州西山居世游网络科技有限公司

(72)发明人 江卓浩 黄龙程 卢歆翻 陈汉辉
李茂 吴海权 陈镇秋 杨键
陈晨

(74)专利代理机构 广州嘉权专利商标事务所有
限公司 44205
代理人 俞梁清

(51)Int.Cl.
G06F 8/70(2018.01)

权利要求书1页 说明书4页 附图3页

(54)发明名称

一种分布式全局ID生成方法及其装置

(57)摘要

本发明涉及一种分布式全局ID生成方法,包括以下步骤:S1、接收指令;S2、提取本地时间戳;S3、判定是否已存在对应节点;S4、获取本地序列码;S5、获取请求方IP地址和端口码;S6、生成请求序列信息并进行判定;S7、生成全局ID;S8、上传全局ID至后台;一种分布式全局ID生成装置,包括接收模块、提取模块、第一判定模块、第一获取模块、第二获取模块、第二判定模块、生成模块以及上传模块。本发明可以实现统一管理生成分布式全局ID,保证了高效、可读、有序且单一等优势,外部业务系统通过HTTP接口即可快速接入,适用于不同的业务场景,可以使业务系统专注于自身业务工作,有效提高了工作效率。



1. 一种分布式全局ID生成方法,包括以下步骤:

S1、接收全局ID生成指令;

S2、提取设备申请全局ID时的本地时间戳;

S3、判定设备ID是否已存在对应节点;

S4、通过判定结果采取对应方式获取本地序列码;

S5、获取请求方IP地址和端口码;

S6、通过请求方IP地址和端口码生成请求序列信息并进行判定;

S7、通过本地时间戳、设备ID以及本地序列码生成全局ID;

S8、上传全局ID至后台。

2. 根据权利要求1所述的分布式全局ID生成方法,其特征在于,所述全局ID采用64位方式存储,其中最高位不使用且始终为0。

3. 根据权利要求1所述的分布式全局ID生成方法,其特征在于,所述设备ID由10bit组成,其分配通过Zookeeper进行管理,所述Zookeeper是一种开源的分布式应用程序协调服务。

4. 根据权利要求1所述的分布式全局ID生成方法,其特征在于,所述本地时间戳由41位bit组成,最小单位为毫秒。

5. 根据权利要求1所述的分布式全局ID生成方法,其特征在于,步骤S4中的判定结果包括已存在、未存在对应节点;若判定结果为已存在对应节点,系统自动获取设备ID对应的本地序列码;若判定结果为未存在对应节点,系统初始化当前节点并获取本地序列码。

6. 根据权利要求5所述的分布式全局ID生成方法,其特征在于,若同一设备在相同时间接入多于一个节点,不同节点依次在已生成的最大本地序列码数值上加一生成新的序列码。

7. 根据权利要求5所述的分布式全局ID生成方法,其特征在于,一个节点对应一个进程,系统支持同一设备在相同时间最多对应4096个进程。

8. 根据权利要求1所述的分布式全局ID生成方法,其特征在于,所述本地序列码由12bit组成。

9. 根据权利要求1所述的分布式全局ID生成方法,其特征在于,步骤S6中的判定包括若判定结果为不通过,则终止本次指令生成操作;若判定结果为通过,则进行下一步。

10. 一种分布式全局ID生成装置,包括:

接收模块,用于接收全局ID生成指令;

提取模块,用于提取设备申请全局ID时的本地时间戳;

第一判定模块,用于判定设备ID是否已存在对应节点;

第一获取模块,用于通过判定结果采取对应方式获取本地序列码;

第二获取模块,用于获取请求方IP地址和端口码;

第二判定模块,用于通过请求方IP地址和端口码生成请求序列信息并进行判定;

生成模块,用于通过本地时间戳、设备ID以及本地序列码生成全局ID;以及

上传模块,用于上传全局ID至后台。

一种分布式全局ID生成方法及其装置

技术领域

[0001] 本发明涉及一种分布式全局ID生成方法及其装置,属于计算机技术领域。

背景技术

[0002] 随着互联网技术的发展,人们在不同业务场景下产生的进程越来越频繁,如何对这些进程进行有效的管理是急需解决的问题。目前主要采用的解决方案有两种,第一种是使用数据库自增序列,这种方式有着数据连续性、唯一性、代码入侵少以及使用简单等优点但存在需要部署独立数据库实例、可见的性能瓶颈且易发生单点故障等缺点;第二种是使用UUID,它的优势在于ID生成由本地完成,性能好,但也存在ID过长、存储冗余、ID无序不可读且作为主键索引性能低下等缺点。

发明内容

[0003] 针对现有技术的不足,本发明提供了一种分布式全局ID生成方法,包括以下步骤:

[0004] S1、接收全局ID生成指令;

[0005] S2、提取设备申请全局ID时的本地时间戳;

[0006] S3、判定设备ID是否已存在对应节点;

[0007] S4、基于判定结果采取对应方式获取本地序列码;

[0008] S5、获取请求方IP地址和端口码;

[0009] S6、基于请求方IP地址和端口码生成请求序列信息并进行判定;

[0010] S7、基于本地时间戳、设备ID以及本地序列码生成全局ID;

[0011] S8、上传全局ID至后台。

[0012] 进一步,所述全局ID采用64位方式存储,其中最高位不使用且始终为0。

[0013] 进一步,所述设备ID由10bit组成,其分配通过Zookeeper进行管理,所述Zookeeper是一种开源的分布式应用程序协调服务。

[0014] 进一步,所述本地时间戳由41位bit组成,最小单位为毫秒。

[0015] 进一步,步骤S4中的判定结果包括已存在、未存在对应节点;若判定结果为已存在对应节点,系统自动获取设备ID对应的本地序列码;若判定结果为未存在对应节点,系统初始化当前节点并获取本地序列码。

[0016] 进一步,若同一设备在相同时间接入多于一个节点,不同节点依次在已生成的最大本地序列码数值上加一生成新的序列码。

[0017] 进一步,一个节点对应一个进程,系统支持同一设备在相同时间最多对应4096个进程。

[0018] 进一步,所述本地序列码由12bit组成。

[0019] 进一步,步骤S6中的判定包括若判定结果为不通过,则终止本次指令生成操作;若判定结果为通过,则进行下一步。

[0020] 一种分布式全局ID生成装置,包括:

- [0021] 接收模块,用于接收全局ID生成指令;
- [0022] 提取模块,用于提取设备申请全局ID时的本地时间戳;
- [0023] 第一判定模块,用于判定设备ID是否已存在对应节点;
- [0024] 第一获取模块,用于通过判定结果采取对应方式获取本地序列码;
- [0025] 第二获取模块,用于获取请求方IP地址和端口码;
- [0026] 第二判定模块,用于通过请求方IP地址和端口码生成请求序列信息并进行判定;
- [0027] 生成模块,用于通过本地时间戳、设备ID以及本地序列码生成全局ID;以及
- [0028] 上传模块,用于上传全局ID至后台。
- [0029] 本发明的有益效果为:统一管理生成分布式全局ID,保证了高效、可读、有序且单一等优势,外部业务系统通过HTTP接口即可快速接入,适用于不同的业务场景,可以使业务系统专注于自身业务工作,有效提高了工作效率。

附图说明

- [0030] 图1所示为根据本发明的总体流程图;
- [0031] 图2所示为根据本发明的装置连接图;
- [0032] 图3所示为根据本发明的具体实施例A;
- [0033] 图4所示为根据本发明的具体实施例B。

具体实施方式

[0034] 应当认识到,本发明的实施例可以由计算机硬件、硬件和软件的组合、或者通过存储在非暂时性计算机可读存储器中的计算机指令来实现或实施。所述方法可以使用标准编程技术-包括配置有计算机程序的非暂时性计算机可读存储介质在计算机程序中实现,其中如此配置的存储介质使得计算机以特定和预定义的方式操作——根据在具体实施例中描述的方法和附图。每个程序可以以高级过程或面向对象的编程语言来实现以与计算机系统通信。然而,若需要,该程序可以以汇编或机器语言实现。在任何情况下,该语言可以是编译或解释的语言。此外,为此目的该程序能够在编程的专用集成电路上运行。

[0035] 此外,可按任何合适的顺序来执行本文描述的过程的操作,除非本文另外指示或以其他方式明显地与上下文矛盾。本文描述的过程(或变型和/或其组合)可在配置有可执行指令的一个或多个计算机系统的控制下执行,并且可作为共同地在一个或多个处理器上执行的代码(例如,可执行指令、一个或多个计算机程序或一个或多个应用)、由硬件或其组合来实现。所述计算机程序包括可由一个或多个处理器执行的多个指令。

[0036] 进一步,所述方法可以在可操作地连接至合适的任何类型的计算平台中实现,包括但不限于个人电脑、迷你计算机、主框架、工作站、网络或分布式计算环境、单独的或集成的计算机平台、或者与带电粒子工具或其它成像装置通信等等。本发明的各方面可以以存储在非暂时性存储介质或设备上的机器可读代码来实现,无论是可移动的还是集成至计算平台,如硬盘、光学读取和/或写入存储介质、RAM、ROM等,使得其可由可编程计算机读取,当存储介质或设备由计算机读取时可用于配置和操作计算机以执行在此所描述的过程。此外,机器可读代码,或其部分可以通过有线或无线网络传输。当此类媒体包括结合微处理器或其他数据处理器实现上文所述步骤的指令或程序时,本文所述的发明包括这些和其他不

同类型的非暂时性计算机可读存储介质。当根据本发明所述的方法和技术编程时,本发明还包括计算机本身。

[0037] 计算机程序能够应用于输入数据以执行本文所述的功能,从而转换输入数据以生成存储至非易失性存储器的输出数据。输出信息还可以应用于一个或多个输出设备如显示器。在本发明优选的实施例中,转换的数据表示物理和有形的对象,包括显示器上产生的物理和有形对象的特定视觉描绘。

[0038] 需要说明的是,如无特殊声明,在本公开中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义。此外,除非另有定义,本文所使用的所有的技术和科学术语与本技术领域技术人员通常理解的含义相同。本文说明书中所使用的术语只是为了描述具体的实施例,而不是为了限制本发明。本文所使用的术语“和/或”包括一个或多个的所列项目的任意的组合。

[0039] 应当理解,本文所提供的任何以及所有实例或示例性语言(“例如”、“如”等)的使用仅意图更好地说明本发明的实施例,并且除非另外要求,否则不会对本发明的范围施加限制。

[0040] 接下来结合附图对本发明的具体实施例作进一步说明:

[0041] 参照图1所示为根据本发明的总体流程图,具体包括以下步骤:

[0042] S1、接收全局ID生成指令;所述全局ID采用64位方式存储,其中最高位不使用且始终为0;

[0043] S2、提取设备申请全局ID时的本地时间戳;所述设备ID由10bit组成,其分配通过Zookeeper进行管理,所述Zookeeper是一种开源的分布式应用程序协调服务;所述本地时间戳由41位bit组成,最小单位为毫秒;

[0044] S3、判定设备ID是否已存在对应节点;

[0045] S4、通过判定结果采取对应方式获取本地序列码;所述判定结果包括已存在、未存在对应节点;若判定结果为已存在对应节点,系统自动获取设备ID对应的本地序列码;若判定结果为未存在对应节点,系统初始化当前节点并获取本地序列码;若同一设备在相同时间接入多于一个节点,不同节点依次在已生成的最大本地序列码数值上加一生成新的序列码;一个节点对应一个进程,系统支持同一设备在相同时间最多对应4096个进程;所述本地序列码由12bit组成;

[0046] S5、获取请求方IP地址和端口码;

[0047] S6、通过请求方IP地址和端口码生成请求序列信息并进行判定;所述判定结果若为不通过,则终止本次生成操作;若判定结果为通过,则进行下一步;

[0048] S7、通过本地时间戳、设备ID以及本地序列码生成全局ID;

[0049] S8、上传全局ID至后台。

[0050] 参照图2所示为根据本发明的装置连接图,包括:

[0051] 接收模块,与提取模块连接实现交互,用于接收全局ID生成指令;

[0052] 提取模块,分别与第一判定模块、第二获取模块连接实现交互,用于提取设备申请全局ID时的本地时间戳;

[0053] 第一判定模块,与第一获取模块连接实现交互,用于判定设备ID是否已存在对应节点;

[0054] 第一获取模块,与第二判定模块连接实现交互,用于通过判定结果采取对应方式获取本地序列码;

[0055] 第二获取模块,与第二判定模块连接实现交互,用于获取请求方IP地址和端口码;

[0056] 第二判定模块,与生成模块连接实现交互,用于通过请求方IP地址和端口码生成请求序列信息并进行判定;

[0057] 生成模块,与上传模块连接实现交互,用于通过本地时间戳、设备ID以及本地序列码生成全局ID;以及

[0058] 上传模块,用于上传全局ID至后台。

[0059] 参照图3所示为根据本发明的具体实施例A,包括:

[0060] 全局ID以64位bit的方式进行存储,分别由4部分组成;

[0061] 最高位为符号位,始终为0;

[0062] 本地时间戳由41位bit展示,最小单位精确至毫秒,存储范围达到69年;

[0063] 设备ID由10位bit展示,最多支持系统部署1024个节点;

[0064] 本地序列码由12位bit展示,最高支持每个节点每毫秒同时产生4096个进程。

[0065] 参照图4所示为根据本发明的具体实施例B,包括:

[0066] 实施例B展示了本发明的在游戏场景下的实际应用,游戏客户端向游戏后台请求交易;游戏后台在完成自身业务逻辑后向KD-SEQ发送全局ID申请指令,此处的KD-SEQ是作为独立组件布置游戏后台的后端,仅用于生成全局ID;记录请求设备的设备ID,即请求机器编码;记录请求方IP地址和端口码;基于设备ID、本地时间戳以及本地序列码生成全局ID;发送全局ID至游戏后台;游戏后台发送全局ID至游戏客户端;指令完成。

[0067] 以上所述,只是本发明的较佳实施例而已,本发明并不局限于上述实施方式,只要其以相同的手段达到本发明的技术效果,都应属于本发明的保护范围。在本发明的保护范围内其技术方案和/或实施方式可以有各种不同的修改和变化。

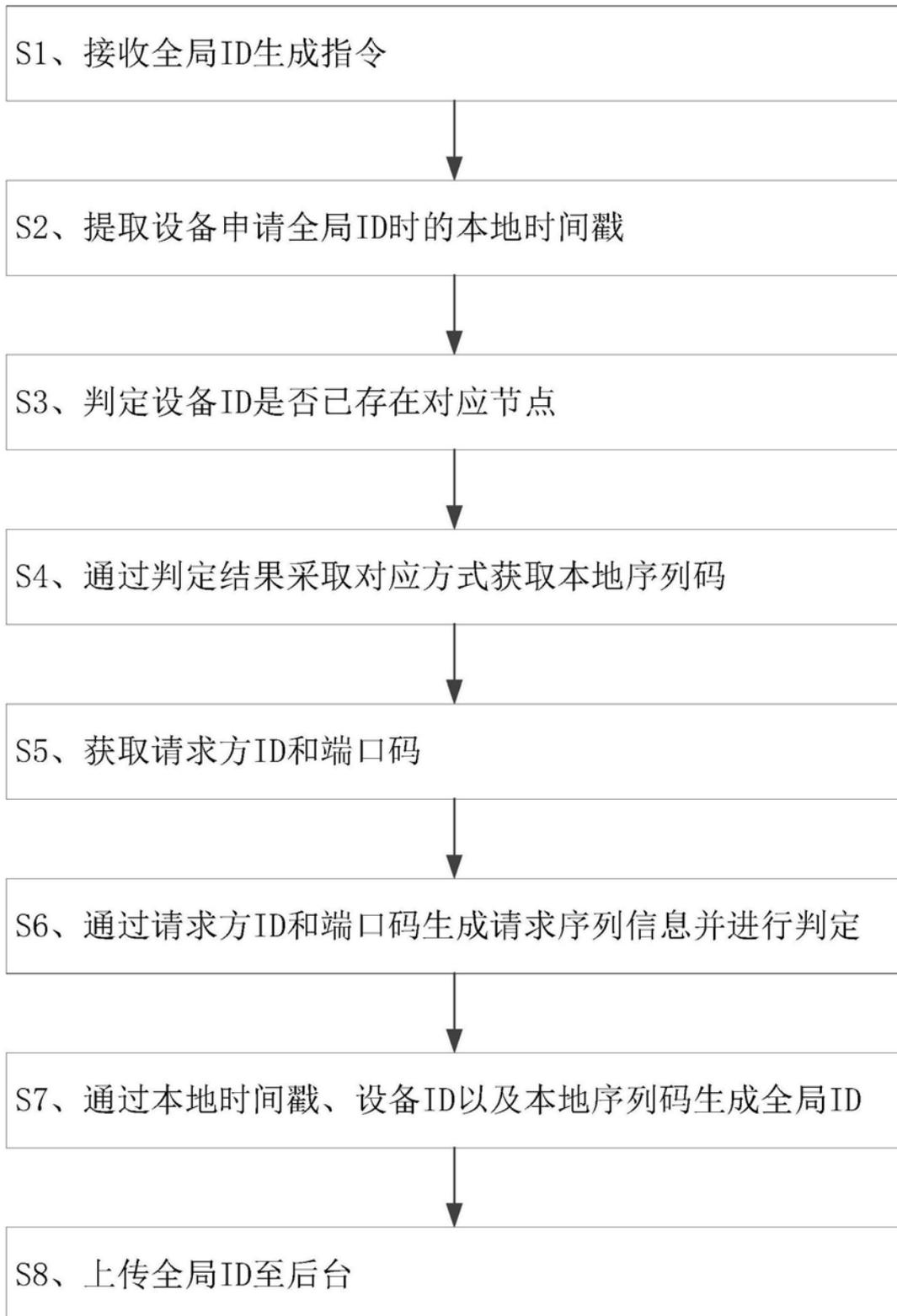


图1

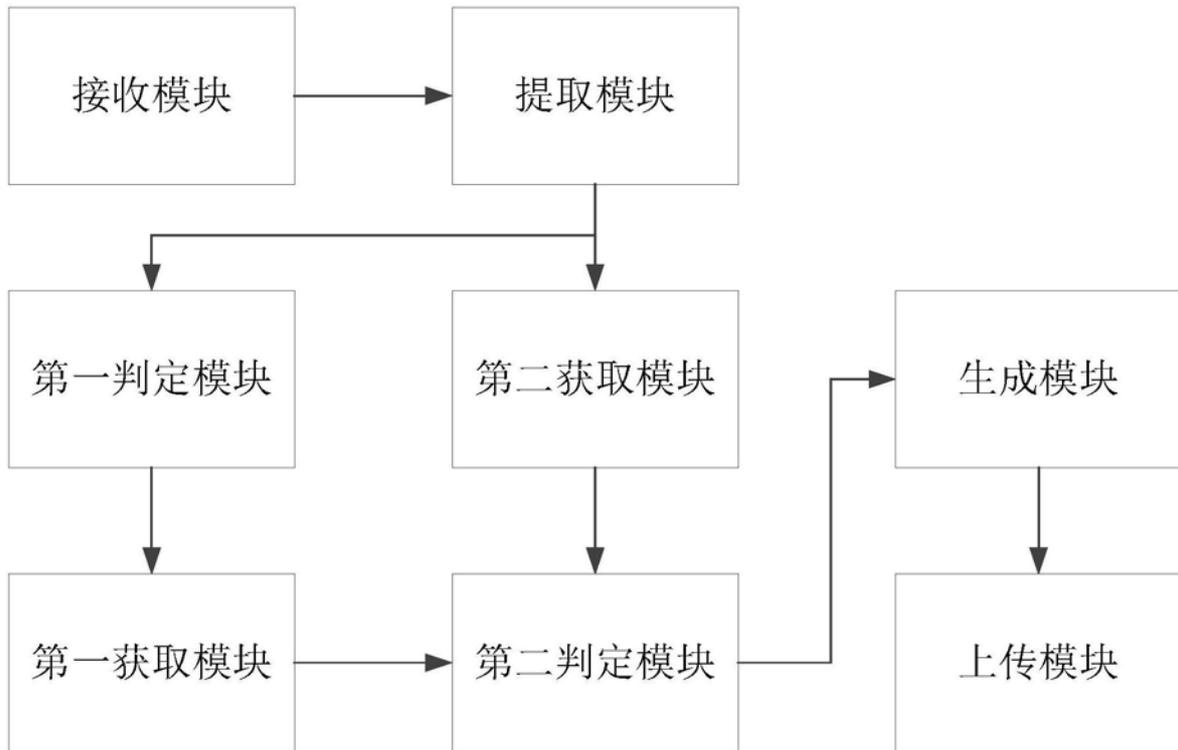


图2

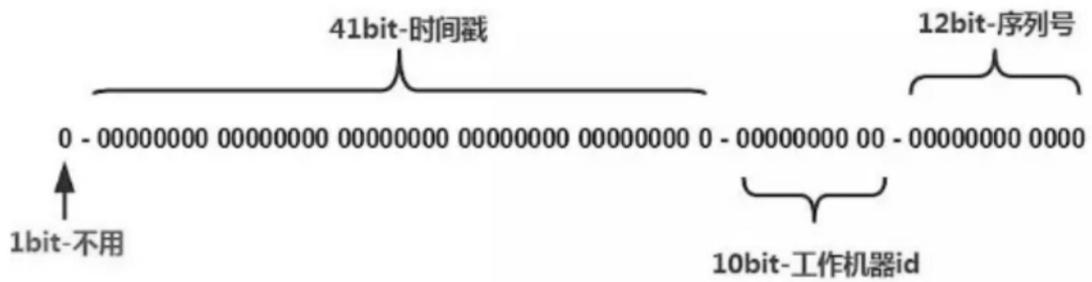


图3

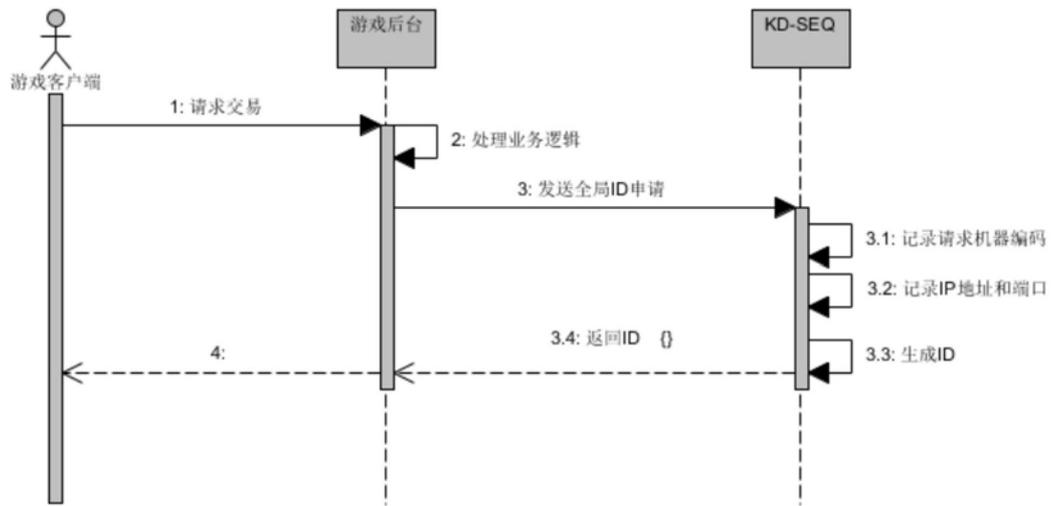


图4