



(12) 发明专利申请

(10) 申请公布号 CN 103731682 A

(43) 申请公布日 2014. 04. 16

(21) 申请号 201410037487. 0

(22) 申请日 2014. 01. 26

(71) 申请人 飞狐信息技术(天津)有限公司
地址 300457 天津市塘沽区经济技术开发区
南港工业区综合服务区办公楼 C 座二
层 210-01 室

(72) 发明人 王震 夏鹏

(74) 专利代理机构 北京集佳知识产权代理有限
公司 11227

代理人 王宝筠

(51) Int. Cl.

H04N 21/239(2011. 01)

H04N 21/437(2011. 01)

H04N 21/242(2011. 01)

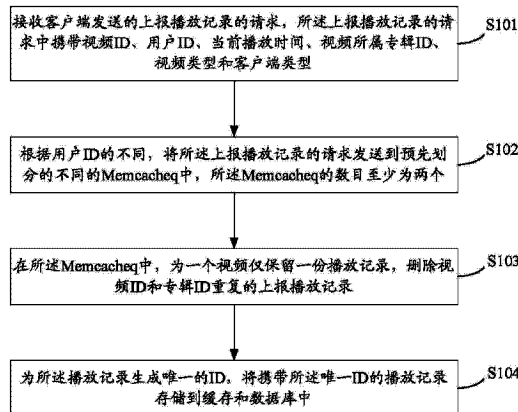
权利要求书2页 说明书12页 附图4页

(54) 发明名称

多种观影设备之间保持同步观看记录的方法、设备及系统

(57) 摘要

本发明提供一种多种观影设备之间保持同步观看记录的方法、设备及系统,方法包括:接收客户端发送的上报播放记录的请求,上报播放记录的请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;根据用户 ID 的不同,将上报播放记录的请求发送到预先划分的不同的消息队列中间件中,消息队列中间件的数目至少为两个;在消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;为播放记录生成唯一的 ID,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。本发明可以使响应客户端的请求加快,并且减少缓存或数据库的占用空间。



1. 一种多种观影设备之间保持同步观看记录的方法,其特征在于,包括:

接收客户端发送的上报播放记录的请求,所述上报播放记录的请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;

在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;

为所述播放记录生成唯一的 ID,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

2. 根据权利要求 1 所述的多种观影设备之间保持同步观看记录的方法,其特征在于,所述根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件之前,还包括:

将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中,每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

3. 根据权利要求 1 所述的多种观影设备之间保持同步观看记录的方法,其特征在于,所述为所述播放记录生成唯一的 ID,具体为:

使用 REDIS 作为分布式 ID 生成器,为所述播放记录生成全局唯一的 ID。

4. 根据权利要求 1-3 任一项所述的多种观影设备之间保持同步观看记录的方法,其特征在于,还包括:

接收客户端发送的获取播放记录的请求,所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型;

根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表;如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找;

通过查找到的所述视频 ID 的列表,从缓存中获取视频的播放记录;如果缓存中没有所述播放记录再从数据库中获取;

将所述播放记录返回给客户端。

5. 一种多种观影设备之间保持同步观看记录的设备,其特征在于,包括:上报播放记录请求接收模块、上报播放记录请求划分模块、播放记录去重模块、播放记录 ID 生成模块和播放记录存储模块;

所述上报播放记录请求接收模块,用于接收客户端发送的上报播放记录的请求,所述上报播放记录请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

所述上报播放记录请求划分模块,用于根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;

所述播放记录去重模块,用于在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;

所述播放记录 ID 生成模块,用于为所述播放记录生成唯一的 ID;

所述播放记录存储模块,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

6. 根据权利要求 5 所述的多种观影设备之间保持同步观看记录的设备,其特征在于,

还包括上报播放记录请求去重模块；

所述上报播放记录请求去重模块，用于将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中，每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

7. 根据权利要求 5 所述的多种观影设备之间保持同步观看记录的设备，其特征在于，所述播放记录 ID 生成模块包括播放记录 ID 生成子模块，用于使用 REDIS 作为分布式 ID 生成器，为所述播放记录生成全局唯一的 ID。

8. 根据权利要求 5-7 任一项所述的多种观影设备之间保持同步观看记录的设备，其特征在于，还包括：获取播放记录请求接收模块、视频 ID 列表查找模块、播放记录获取模块和播放记录返回模块；

所述获取播放记录请求接收模块，用于接收客户端发送的获取播放记录的请求，所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型；

所述视频 ID 列表查找模块，用于根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表；如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找；

所述播放记录获取模块，用于通过查找到的所述视频 ID 的列表，从缓存中获取视频的播放记录；如果缓存中没有所述播放记录再从数据库中获取；

所述播放记录返回模块，用于将所述播放记录返回给客户端。

9. 一种多种观影设备之间保持同步观看记录的系统，其特征在于，包括：客户端、服务器端；

所述客户端，用于向所述服务器端发送上报播放记录的请求，所述上报播放记录的请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型；

所述服务器端，用于根据用户 ID 的不同，将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中，所述消息队列中间件的数目至少为两个；在所述消息队列中间件中，为一个视频仅保留一份播放记录，删除视频 ID 和专辑 ID 重复的上报播放记录；为所述播放记录生成唯一的 ID，将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

10. 根据权利要求 9 所述的多种观影设备之间保持同步观看记录的系统，其特征在于，所述服务器端根据用户 ID 的不同，将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件之前，还包括：

将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中，每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

11. 根据权利要求 9 或 10 所述的多种观影设备之间保持同步观看记录的系统，其特征在于，所述客户端，还用于向所述服务器端发送获取播放记录的请求，所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型；

所述服务器端，还用于根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表；如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找；通过查找到的所述视频 ID 的列表，从缓存中获取视频的播放记录；如果缓存中没有所述播放记录再从数据库中获取；将所述播放记录返回给客户端。

多种观影设备之间保持同步观看记录的方法、设备及系统

技术领域

[0001] 本发明涉及视频播放技术领域,特别涉及一种多种观影设备之间保持同步观看记录的方法、设备及系统。

背景技术

[0002] 首先介绍本领域的几个专业名称:

[0003] 云记录:一种服务端计算方式,客户端不保留任何数据,直接将用户记录快速上报到服务器端,通过服务器端复杂的计算,将结果有保留性的保存。

[0004] 用户生成内容(UGC, User Generated Content):用户将自己原创的内容通过互联网平台进行展示或者提供给其他用户。

[0005] Memcacheq:一种消息中间件产品,作用是作为消息生产者与消费者的数据中转平台。

[0006] LVS:实现 Web 集群间负载均衡的调度器。

[0007] NGINX(Engine x):一个高性能的 HTTP 和反向代理服务器。

[0008] REDIS:一个高性能的键-值(Key-value)存储系统。

[0009] MYSQL:一个高性能的关系型数据库管理系统。

[0010] JSON(JavaScript Object Notation):是一种轻量级的数据交换格式。它是基于 JavaScript(Standard ECMA-2623rd Edition-December1999)的一个子集。JSON 采用完全独立于语言的文本格式,但是也使用了类似于 C 语言家族的习惯(包括 C, C++, C#, Java, JavaScript, Perl, Python 等)。这些特性使 JSON 成为理想的数据交换语言。

[0011] 由于云计算正处于高度发展时期,因此,越来越多的商家和互联网巨头开始投入到技术的研发和应用的推广上,云计算的技术也逐步走向成熟和完善。云服务端可以看成一个小型的云系统。客户端只需要简单的上报和获取播放记录,复杂的计算和存储问题交给云服务端处理。

[0012] 云服务端使用 Web 集群处理请求,使用 LVS 来实现服务器之间的负载均衡。使用分布式服务保证服务的稳定性,当有服务因为网络或机器原因挂掉时,可以切换到其他服务。使用消息队列中间件保存上报请求,使用 JAVA 线程池技术异步处理上报请求。存储方面,使用 REDIS 作为缓存,使用 MYSQL 作为数据库。

[0013] 对于视频网站,用户观看视频时,观影设备会向云记录服务端上报该用户的观看记录,其中观看记录包括用户观看的视频是什么以及该视频的播放进度。当用户使用 PC 端的客户端观看视频时,观影设备会定时上报观看记录,例如每隔两分钟上报一次。当用户使用手机上的客户端观看视频时,在退出登录时观影设备会向云记录服务端上报播放记录。

[0014] 对于云记录服务端,接收到观影设备上报播放记录的请求时,存入缓存和数据库;接收到观影设备获取播放记录的请求时,优先从缓存中查找,缓存中没有再去数据库中查找,返回 JSON 数据格式。

[0015] 目前,随着网站的用户量上升以及观影设备的多样性增加,视频的访问量迅速增

加,客户端上传播放记录的并发量就会非常大,直接调用服务可能得不到及时的响应,甚至会引起调用的超时或者服务的崩溃。因此,现有技术存在上传播放记录反应慢或者上传失败的问题,并且客户端在获取播放记录时,响应也有一定的延迟。

发明内容

[0016] 本发明要解决的技术问题是提供一种多种观影设备之间保持同步观看记录的方法、设备及系统,能够及时响应播放记录的上报和获取。

[0017] 本发明实施例提供一种多种观影设备之间保持同步观看记录的方法,包括:

[0018] 接收客户端发送的上报播放记录的请求,所述上报播放记录的请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

[0019] 根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;

[0020] 在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;

[0021] 为所述播放记录生成唯一的 ID,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

[0022] 优选地,所述根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件之前,还包括:

[0023] 将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中,每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

[0024] 优选地,所述为所述播放记录生成唯一的 ID,具体为:

[0025] 使用 REDIS 作为分布式 ID 生成器,为所述播放记录生成全局唯一的 ID。

[0026] 优选地,还包括:

[0027] 接收客户端发送的获取播放记录的请求,所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型;

[0028] 根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表;如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找;

[0029] 通过查找到的所述视频 ID 的列表,从缓存中获取视频的播放记录;如果缓存中没有所述播放记录再从数据库中获取;

[0030] 将所述播放记录返回给客户端。

[0031] 本发明实施例还提供一种多种观影设备之间保持同步观看记录的设备,包括:上报播放记录请求接收模块、上报播放记录请求划分模块、播放记录去重模块、播放记录 ID 生成模块和播放记录存储模块;

[0032] 所述上报播放记录请求接收模块,用于接收客户端发送的上报播放记录的请求,所述上报播放记录请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

[0033] 所述上报播放记录请求划分模块,用于根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;

[0034] 所述播放记录去重模块,用于在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;

[0035] 所述播放记录 ID 生成模块,用于为所述播放记录生成唯一的 ID;

[0036] 所述播放记录存储模块,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

[0037] 优选地,还包括上报播放记录请求去重模块;

[0038] 所述上报播放记录请求去重模块,用于将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中,每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

[0039] 优选地,所述播放记录 ID 生成模块包括播放记录 ID 生成子模块,用于使用 REDIS 作为分布式 ID 生成器,为所述播放记录生成全局唯一的 ID。

[0040] 优选地,还包括:获取播放记录请求接收模块、视频 ID 列表查找模块、播放记录获取模块和播放记录返回模块;

[0041] 所述获取播放记录请求接收模块,用于接收客户端发送的获取播放记录的请求,所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型;

[0042] 所述视频 ID 列表查找模块,用于根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表;如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找;

[0043] 所述播放记录获取模块,用于通过查找到的所述视频 ID 的列表,从缓存中获取视频的播放记录;如果缓存中没有所述播放记录再从数据库中获取;

[0044] 所述播放记录返回模块,用于将所述播放记录返回给客户端。

[0045] 本发明实施例还提供一种多种观影设备之间保持同步观看记录的系统,包括:客户端、服务器端;

[0046] 所述客户端,用于向所述服务器端发送上报播放记录的请求,所述上报播放记录的请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

[0047] 所述服务器端,用于根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;为所述播放记录生成唯一的 ID,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

[0048] 10、根据权利要求 9 所述的多种观影设备之间保持同步观看记录的系统,其特征在于,所述服务器端根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件之前,还包括:

[0049] 将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中,每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

[0050] 11、根据权利要求 9 或 10 所述的多种观影设备之间保持同步观看记录的系统,其特征在于,所述客户端,还用于向所述服务器端发送获取播放记录的请求,所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型;

[0051] 所述服务器端,还用于根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表;如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数

数据库中查找；通过查找到的所述视频 ID 的列表，从缓存中获取视频的播放记录；如果缓存中没有所述播放记录再从数据库中获取；将所述播放记录返回给客户端。

[0052] 与现有技术相比，本发明具有以下优点：

[0053] 本发明实施例提供的多种观影设备之间保持同步观看记录的方法，根据用户 ID 的不同，将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中，这样可以避免上报播放记录的队列太长，从而使响应客户端的请求加快，另外，为了一个视频仅保留一份播放记录，删除视频 ID 和专辑 ID 重复的上报播放记录，这样可以减少缓存或数据库的占用空间，并且在客户端获取播放记录时能够更快地查找到请求的播放记录，从而加快请求的响应。

附图说明

[0054] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0055] 图 1 是本发明提供的多种观影设备之间保持同步观看记录的方法实施例一示意图；

[0056] 图 2 是本发明提供的多种观影设备之间保持同步观看记录的方法实施例二示意图；

[0057] 图 3 是本发明提供的多种观影设备之间保持同步观看记录的方法实施例三示意图；

[0058] 图 4 是本发明提供的多种观影设备之间保持同步观看记录的设备实施例一示意图；

[0059] 图 5 是本发明提供的多种观影设备之间保持同步观看记录的设备实施例二示意图；

[0060] 图 6 是本发明提供的多种观影设备之间保持同步观看记录的设备实施例三示意图；

[0061] 图 7 是本发明提供的多种观影设备之间保持同步观看记录的系统实施例一示意图。

具体实施方式

[0062] 下面将结合本发明实施例中的附图，对本发明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发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0063] 为使本发明的上述目的、特征和优点能够更加明显易懂，下面结合附图对本发明的具体实施方式做详细的说明。

[0064] 方法实施例一：

[0065] 参见图 1，该图为本发明提供的多种观影设备之间保持同步观看记录的方法实施

例一示意图。

[0066] 本实施例提供的多种观影设备之间保持同步观看记录的方法,包括以下步骤:

[0067] S101:接收客户端发送的上报播放记录的请求,所述上报播放记录的请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

[0068] 需要说明的是,当用户使用客户端观看视频时,需要上报播放记录,播放记录包括视频 ID、用户 ID、当前播放时间。视频所述专辑 ID、视频类型和客户端类型;

[0069] 其中,当前播放时间指的是,用户观看的视频当前观看到了第几分第几秒等。

[0070] 其中,客户端类型可以分为 PC 客户端,以及移动客户端等。移动客户端例如是手机客户端, PAD 客户端等。

[0071] 当用户使用 PC 端的客户端观看视频时,观影设备会定时上报观看记录,例如每隔两分钟上报一次。当用户使用手机上的客户端观看视频时,在退出登录时观影设备会向云记录服务端上报播放记录。

[0072] 其中,视频所述专辑指的是拥有共同特性的视频集合,例如某部电视剧可以称为一个专辑,一个电视剧的视频属于同一个专辑中,但是一个视频可能属于一个或者多个专辑,也可能不属于任何专辑。

[0073] 客户端首先将上报播放记录请求发送到 LVS;

[0074] LVS 根据负载均衡策略将上报播放记录请求转发到某一台 NGINX 服务器上。

[0075] NGINX 作为反向代理,将上报播放记录请求转发到对应的 WEB 端服务器。

[0076] S102:根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;

[0077] 需要说明的是,所述消息队列中间件可以由 Memcacheq 来实现,也可以由其他的消息队列中间件来实现。

[0078] 具体如何根据用户 ID 的不同来划分上报播放记录到不同的 Memcacheq 队列中,可以自由来设定,例如,当设置两个不同的 Memcacheq 队列时,可以将用户 ID 为偶数的放在一个 Memcacheq 队列中,将用户 ID 为奇数的放在另一个 Memcacheq 队列中。

[0079] 当设置三个不同的 Memcacheq 队列时,可以利用求 3 的余数的算法($\%3$),将用户 ID 对 3 求余数,根据余数为 0、1 和 2 分别放入第一个 Memcacheq 队列中、第二个 Memcacheq 队列中以及第三个 Memcacheq 队列中。

[0080] 可以理解的是,当有两个 Memcacheq 队列时,也可以利用用户 ID 对 2 求余数,余数为 0 和 1 放入两个不同的 Memcacheq 队列中,例如,余数为 0 放入第一个 Memcacheq 队列中,余数为 1 放入第二个 Memcacheq 队列中。

[0081] 例如用户 ID 为 3438434,可以利用该数值对 2 求余数。

[0082] 可以理解的是,服务器端会首先解析客户端发送的上报播放记录请求中的各种信息,该信息包括视频 ID、用户 ID、当前播放时间。视频所述专辑 ID、视频类型和客户端类型;

[0083] 需要说明的是,本发明实施例中首先将上报播放记录的请求发送到不同的消息队列中间件中,这样主要是为了防止上报播放记录的请求队列太长,服务器响应太慢。

[0084] 可以理解的是,消息队列中间件的具体数目在本发明实施例中不做具体限定,但是最少需要两个,一个消息队列中间件与现有技术没有区别,起不到防止队列太长的作用。

[0085] S103:在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID

和专辑 ID 重复的上报播放记录；

[0086] 因为,用户上报的播放记录会有很多,例如,用户观看的视频为一个电视剧,现在看到了第 16 集,会将第 1 集-第 15 集的播放记录全部删除,只保留第 16 集的播放记录。因为客户不希望追看一部电视剧时,看到自己播放记录中有同一部电视剧的多个播放记录。

[0087] S104:为所述播放记录生成唯一的 ID,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

[0088] 本发明实施例提供的多种观影设备之间保持同步观看记录的方法,根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,这样可以避免上报播放记录的队列太长,从而使响应客户端的请求加快,另外,为了一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录,这样可以减少缓存或数据库的占用空间,并且在客户端获取播放记录时能够更快地查找到请求的播放记录,从而加快请求的响应。

[0089] 方法实施例二:

[0090] 参见图 2,该图为本发明提供的多种观影设备之间保持同步观看记录的方法实施例二示意图。

[0091] 由于当客户端为 PC 客户端时,由于 PC 客户端发送规则比较乱,容易存在很多重复、间隔时间很短的上报播放记录的请求,例如,PC 客户端每隔两分钟发送一次上报播放记录的请求,这样在一小时之内就会有大量的上报播放记录。本实施例中为了减轻服务器端的压力,在 WEB 端对上报播放记录的请求去重之后才发送到不同的消息队列中间件中。

[0092] 下面以方法实施例二为例进行说明。

[0093] 本实施例中的 S201 和方法实施例一中的 S101 相同,在此不再赘述。

[0094] S202:根据客户端类型判断客户端是否为 PC 客户端,如果不是 PC 客户端,则执行 S204;如果是 PC 客户端,则执行 S203;

[0095] S203:将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中,每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

[0096] 具体去重的方式为,运行一个大小可以扩展的线程池,线程池中每个线程都有自己的阻塞队列,每个阻塞队列用来保存客户端的上报播放记录的请求。每个线程每个预定时间去自己的阻塞队列中取出所述上报播放记录的请求,使用 hash 去重,然后将去重后的上报播放记录进行 S204 的处理。

[0097] 具体地,WEB 端首先在线程池中选择一个不在运行的线程,将来自 PC 客户端的上报播放记录发送到该线程的阻塞队列中。

[0098] S204 与方法实施例一中的 S102 相同,在此不再赘述。

[0099] S205:首先判断消息队列中间件中的上报播放记录的请求中的用户 ID 是否有效以及专辑 ID 是否有效;如果有效则,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;

[0100] S206:使用 REDIS 作为分布式的 ID 生成器,为播放记录生成全局唯一的 ID,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

[0101] 由于有多个服务器在处理消息队列中间件中的上报播放记录的请求,如果使用传统的数据库中 ID 自增来生成 ID 的方法,会对数据库造成影响,因为请求量大时,高峰时期

每秒会有一千以上的请求增量,这样对数据库的影响很大,数据库面临很大的压力。因此,本实施例中采用 REDIS 来为播放记录生成唯一的 ID。

[0102] 多个服务器同时处理消息队列中间件中的请求,这样可以加快响应速度,避免一个服务器处理时的请求堆积。

[0103] 本实施例提供的方法的优点是,在将 PC 客户端的上报播放记录请求放入消息队列中间件中之前,先对 PC 客户端的上报播放记录请求进行去重处理,因为 PC 客户端发送的上报播放记录请求存在很多重复、时间间隔很短的请求,删除重复的请求之后可以减轻服务器端的压力,从而加快响应时间。

[0104] 需要说明的是,在服务器端将客户端上报的播放记录保存成功以后,会以 JSON 数据返回保存是否成功的回馈消息。

[0105] 方法实施例三:

[0106] 方法实施例一和方法实施例二是以客户端上报播放记录为例进行描述的,下面结合附图介绍客户端获取播放记录为例进行描述。

[0107] 参见图 3,该图为本发明提供的多种观影设备之间保持同步观看记录的方法实施例三示意图。

[0108] S301:接收客户端发送的获取播放记录的请求,所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型;

[0109] 需要说明的是,用户主动点击视频时,说明是需要获取播放记录,此时观影设备的客户端自动向服务器端发送获取播放记录的请求。

[0110] S302:根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表;如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找;

[0111] S303:通过查找到的所述视频 ID 的列表,从缓存中获取视频的播放记录;如果缓存中没有所述播放记录再从数据库中获取;

[0112] S304:将所述播放记录返回给客户端。

[0113] 需要说明的是,本发明实施例提供的获取播放记录的方法是基于方法实施例或方法实施例二的基础之上的,服务器端接收到获取播放记录的请求后查找到该请求对应的播放记录返回给客户端。

[0114] 基于以上实施例提供的多种观影设备之间保持同步观看记录的方法,本发明实施例还提供了一种多种观影设备之间保持同步观看记录的设备,下面结合附图进行详细介绍。

[0115] 参见图 4,该图为本发明提供的多种观影设备之间保持同步观看记录的设备实施例一示意图。

[0116] 设备实施例一:

[0117] 本发明实施例提供的多种观影设备之间保持同步观看记录的设备,包括:上报播放记录请求接收模块 400、上报播放记录请求划分模块 500、播放记录去重模块 600、播放记录 ID 生成模块 700 和播放记录存储模块 800;

[0118] 上报播放记录请求接收模块 400,用于接收客户端发送的上报播放记录的请求,所述上报播放记录请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

[0119] 需要说明的是,当用户使用客户端观看视频时,需要上报播放记录,播放记录包括视频 ID、用户 ID、当前播放时间。视频所述专辑 ID、视频类型和客户端类型;

[0120] 其中,当前播放时间指的是,用户观看的视频当前观看了第几分第几秒等。

[0121] 其中,客户端类型可以分为 PC 客户端,以及移动客户端等。移动客户端例如是手机客户端, PAD 客户端等。

[0122] 当用户使用 PC 端的客户端观看视频时,观影设备会定时上报观看记录,例如每隔两分钟上报一次。当用户使用手机上的客户端观看视频时,在退出登录时观影设备会向云记录服务端上报播放记录。

[0123] 其中,视频所述专辑指的是拥有共同特性的视频集合,例如某部电视剧可以称为一个专辑,一个电视剧的视频属于同一个专辑中,但是一个视频可能属于一个或者多个专辑,也可能不属于任何专辑。

[0124] 客户端首先将上报播放记录请求发送到 LVS;

[0125] LVS 根据负载均衡策略将上报播放记录请求转发到某一台 NGINX 服务器上。

[0126] NGINX 作为反向代理,将上报播放记录请求转发到对应的 WEB 端服务器。

[0127] 上报播放记录请求划分模块 500,用于根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;

[0128] 可以理解的是,服务器端会首先解析客户端发送的上报播放记录请求中的各种信息,该信息包括视频 ID、用户 ID、当前播放时间。视频所述专辑 ID、视频类型和客户端类型;

[0129] 需要说明的是,所述消息队列中间件可以由 Memcacheq 来实现,也可以由其他的消息队列中间件来实现。

[0130] 具体如何根据用户 ID 的不同来划分上报播放记录到不同的 Memcacheq 队列中,可以自由来设定,例如,当设置两个不同的 Memcacheq 队列时,可以将用户 ID 为偶数的放在一个 Memcacheq 队列中,将用户 ID 为奇数的放在另一个 Memcacheq 队列中。

[0131] 当设置三个不同的 Memcacheq 队列时,可以利用求 3 的余数的算法(%3),将用户 ID 对 3 求余数,根据余数为 0、1 和 2 分别放入第一个 Memcacheq 队列中、第二个 Memcacheq 队列中以及第三个 Memcacheq 队列中。

[0132] 可以理解的是,当有两个 Memcacheq 队列时,也可以利用用户 ID 对 2 求余数,余数为 0 和 1 放入两个不同的 Memcacheq 队列中,例如,余数为 0 放入第一个 Memcacheq 队列中,余数为 1 放入第二个 Memcacheq 队列中。

[0133] 例如用户 ID 为 3438434,可以利用该数值对 2 求余数。

[0134] 需要说明的是,本发明实施例中首先将上报播放记录的请求发送到不同的消息队列中间件中,这样主要是为了防止上报播放记录的请求队列太长,服务器响应太慢。

[0135] 可以理解的是,消息队列中间件的具体数目在本发明实施例中不做具体限定,但是最少需要两个,一个消息队列中间件与现有技术没有区别,起不到防止队列太长的作用。

[0136] 播放记录去重模块 600,用于在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;

[0137] 因为,用户上报的播放记录会有很多,例如,用户观看的视频为一个电视剧,现在看到了第 16 集,会将第 1 集 - 第 15 集的播放记录全部删除,只保留第 16 集的播放记录。因

为客户不希望追看一部电视剧时,看到自己播放记录中有同一部电视剧的多个播放记录。

[0138] 播放记录 ID 生成模块 700,用于为所述播放记录生成唯一的 ID;

[0139] 播放记录存储模块 800,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

[0140] 本发明实施例提供的多种观影设备之间保持同步观看记录的设备,根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,这样可以避免上报播放记录的队列太长,从而使响应客户端的请求加快,另外,为了一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录,这样可以减少缓存或数据库的占用空间,并且在客户端获取播放记录时能够更快地查找到请求的播放记录,从而加快请求的响应。

[0141] 设备实施例二:

[0142] 参见图 5,该图为本发明提供的多种观影设备之间保持同步观看记录的设备实施例二示意图。

[0143] 由于当客户端为 PC 客户端时,由于 PC 客户端发送规则比较乱,容易存在很多重复、间隔时间很短的上报播放记录的请求,例如,PC 客户端每隔两分钟发送一次上报播放记录的请求,这样在一小时之内就会有大量的上报播放记录。本实施例中为了减轻服务器端的压力,在 WEB 端对上报播放记录的请求去重之后才发送到不同的消息队列中间件中。

[0144] 下面以设备实施例二为例进行说明。

[0145] 本实施例提供的设备,还包括上报播放记录请求去重模块 900;

[0146] 所述上报播放记录请求去重模块 900,用于将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中,每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

[0147] 具体去重的方式为,运行一个大小可以扩展的线程池,线程池中每个线程都有自己的阻塞队列,每个阻塞队列用来保存客户端的上报播放记录的请求。每个线程每个预定时间去自己的阻塞队列中取出所述上报播放记录的请求,使用 hash 去重,然后将去重后的上报播放记录进行 S204 的处理。

[0148] 具体地,WEB 端首先在线程池中选择一个不在运行的线程,将来自 PC 客户端的上报播放记录发送到该线程的阻塞队列中。

[0149] 本实施例中,所述播放记录 ID 生成模块 700 包括播放记录 ID 生成子模块 700a,用于使用 REDIS 作为分布式 ID 生成器,为所述播放记录生成全局唯一的 ID。

[0150] 由于有多个服务器在处理消息队列中间件中的上报播放记录的请求,如果使用传统的数据库中 ID 自增来生成 ID 的方法,会对数据库造成影响,因此,本实施例中采用 REDIS 来为播放记录生成唯一的 ID。

[0151] 本实施例提供的设备的优点是,在将 PC 客户端的上报播放记录请求放入消息队列中间件之前,先对 PC 客户端的上报播放记录请求进行去重处理,因为 PC 客户端发送的上报播放记录请求存在很多重复、时间间隔很短的请求,删除重复的请求之后可以减轻服务器端的压力,从而加快响应时间。

[0152] 需要说明的是,在服务器端将客户端上报的播放记录保存成功以后,会以 JSON 数据返回保存是否成功的回馈消息。

[0153] 设备实施例三:

[0154] 参见图 6, 该图为本发明提供的多种观影设备之间保持同步观看记录的设备实施例三示意图。

[0155] 方法实施例一和方法实施例二是以客户端上报播放记录为例进行描述的, 下面结合附图介绍客户端获取播放记录为例进行描述。

[0156] 本实施例提供的多种观影设备之间保持同步观看记录的设备, 还包括: 获取播放记录请求接收模块 100、视频 ID 列表查找模块 200、播放记录获取模块 300 和播放记录返回模块 400;

[0157] 所述获取播放记录请求接收模块 100, 用于接收客户端发送的获取播放记录的请求, 所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型;

[0158] 需要说明的是, 用户主动点击视频时, 说明是需要获取播放记录, 此时观影设备的客户端自动向服务器端发送获取播放记录的请求。

[0159] 所述视频 ID 列表查找模块 200, 用于根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表; 如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找;

[0160] 所述播放记录获取模块 300, 用于通过查找到的所述视频 ID 的列表, 从缓存中获取视频的播放记录; 如果缓存中没有所述播放记录再从数据库中获取;

[0161] 所述播放记录返回模块 400, 用于将所述播放记录返回给客户端。

[0162] 需要说明的是, 本发明实施例提供的获取播放记录的方法是基于方法实施例或方法实施例二的基础之上的, 服务器端接收到获取播放记录的请求后查找到该请求对应的播放记录返回给客户端。

[0163] 基于以上实施例提供的一种多种观影设备之间保持同步观看记录的方法和设备, 本发明实施例还提供了一种多种观影设备之间保持同步观看记录的系统, 下面结合附图进行详细的介绍。

[0164] 系统实施例一:

[0165] 参见图 7, 该图为本发明提供的多种观影设备之间保持同步观看记录的系统实施例一示意图。

[0166] 本实施例提供一种多种观影设备之间保持同步观看记录的系统, 包括: 客户端 1000、服务器端 2000;

[0167] 所述客户端 1000, 用于向所述服务器端 2000 发送上报播放记录的请求, 所述上报播放记录的请求中携带视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

[0168] 需要说明的是, 当用户使用客户端 1000 看视频时, 需要上报播放记录, 播放记录包括视频 ID、用户 ID、当前播放时间、视频所属专辑 ID、视频类型和客户端类型;

[0169] 其中, 当前播放时间指的是, 用户观看的视频当前观看了第几分第几秒等。

[0170] 其中, 客户端类型可以分为 PC 客户端, 以及移动客户端等。移动客户端例如是手机客户端, PAD 客户端等。

[0171] 当用户使用 PC 客户端观看视频时, 观影设备会定时上报观看记录, 例如每隔两分钟上报一次。当用户使用手机上的客户端观看视频时, 在退出登录时观影设备会向云记录服务端上报播放记录。

[0172] 其中,视频所述专辑指的是拥有共同特性的视频集合,例如某部电视剧可以称为一个专辑,一个电视剧的视频属于同一个专辑中,但是一个视频可能属于一个或者多个专辑,也可能不属于任何专辑。

[0173] 客户端 1000 首先将上报播放记录请求发送到 LVS;

[0174] LVS 根据负载均衡策略将上报播放记录请求转发到某一台 NGINX 服务器上。

[0175] NGINX 作为反向代理,将上报播放记录请求转发到对应的 WEB 端服务器。

[0176] 所述服务器端 2000,用于根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,所述消息队列中间件的数目至少为两个;在所述消息队列中间件中,为一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录;为所述播放记录生成唯一的 ID,将携带所述唯一 ID 的播放记录存储到缓存和数据库中。

[0177] 需要说明的是,所述消息队列中间件可以由 Memcacheq 来实现,也可以由其他的消息队列中间件来实现。

[0178] 具体如何根据用户 ID 的不同来划分上报播放记录到不同的 Memcacheq 队列中,可以自由来设定,例如,当设置两个不同的 Memcacheq 队列时,可以将用户 ID 为偶数的放在一个 Memcacheq 队列中,将用户 ID 为奇数的放在另一个 Memcacheq 队列中。

[0179] 当设置三个不同的 Memcacheq 队列时,可以利用求 3 的余数的算法(%3),将用户 ID 对 3 求余数,根据余数为 0、1 和 2 分别放入第一个 Memcacheq 队列中、第二个 Memcacheq 队列中以及第三个 Memcacheq 队列中。

[0180] 可以理解的是,当有两个 Memcacheq 队列时,也可以利用用户 ID 对 2 求余数,余数为 0 和 1 放入两个不同的 Memcacheq 队列中,例如,余数为 0 放入第一个 Memcacheq 队列中,余数为 1 放入第二个 Memcacheq 队列中。

[0181] 例如用户 ID 为 3438434,可以利用该数值对 2 求余数。

[0182] 可以理解的是,服务器端会首先解析客户端发送的上报播放记录请求中的各种信息,该信息包括视频 ID、用户 ID、当前播放时间。视频所述专辑 ID、视频类型和客户端类型;

[0183] 需要说明的是,本发明实施例中首先将上报播放记录的请求发送到不同的消息队列中间件中,这样主要是为了防止上报播放记录的请求队列太长,服务器响应太慢。

[0184] 可以理解的是,消息队列中间件的具体数目在本发明实施例中不做具体限定,但是最少需要两个,一个消息队列中间件与现有技术没有区别,起不到防止队列太长的作用。

[0185] 因为,用户上报的播放记录会有很多,例如,用户观看的视频为一个电视剧,现在看到了第 16 集,会将第 1 集 - 第 15 集的播放记录全部删除,只保留第 16 集的播放记录。因为客户不希望追看一部电视剧时,看到自己播放记录中有同一部电视剧的多个播放记录。

[0186] 本发明实施例提供的多种观影设备之间保持同步观看记录的系统,根据用户 ID 的不同,将所述上报播放记录的请求发送到预先划分的不同的消息队列中间件中,这样可以避免上报播放记录的队列太长,从而使响应客户端的请求加快,另外,为了一个视频仅保留一份播放记录,删除视频 ID 和专辑 ID 重复的上报播放记录,这样可以减少缓存或数据库的占用空间,并且在客户端获取播放记录时能够更快地查找到请求的播放记录,从而加快请求的响应。

[0187] 系统实施例二:

[0188] 本实施例提供的系统,所述服务器端 2000 根据用户 ID 的不同,将所述上报播放记录请求发送到预先划分的不同的消息队列中间件中之前,还包括:

[0189] 将所述客户端类型为 PC 客户端的上报播放记录请求先保存在线程池中,每隔预定时间使用 hash 对所述线程池中的上报播放记录请求进行去重。

[0190] 由于当客户端为 PC 客户端时,由于 PC 客户端发送规则比较乱,容易存在很多重复、间隔时间很短的上报播放记录的请求,例如,PC 客户端每隔两分钟发送一次上报播放记录的请求,这样在一小时之内就会有大量的上报播放记录。本实施例中为了减轻服务器端的压力,在 WEB 端对上报播放记录的请求去重之后才发送到不同的消息队列中间件中。

[0191] 具体去重的方式为,运行一个大小可以扩展的线程池,线程池中每个线程都有自己的阻塞队列,每个阻塞队列用来保存客户端的上报播放记录的请求。每个线程每个预定时间去自己的阻塞队列中取出所述上报播放记录的请求,使用 hash 去重。

[0192] 由于有多个服务器在处理消息队列中间件中的上报播放记录的请求,如果使用传统的数据库中 ID 自增来生成 ID 的方法,会对数据库造成影响,因此,本实施例中采用 REDIS 来为播放记录生成唯一的 ID。

[0193] 本实施例提供的系统的优点是,在将 PC 客户端的上报播放记录请求放入消息队列中间件中之前,先对 PC 客户端的上报播放记录请求进行去重处理,因为 PC 客户端发送的上报播放记录请求存在很多重复、时间间隔很短的请求,删除重复的请求之后可以减轻服务器端的压力,从而加快响应时间。

[0194] 需要说明的是,在服务器端将客户端上报的播放记录保存成功以后,会以 JSON 数据返回保存是否成功的回馈消息。

[0195] 以上实施例是以客户端上报播放记录为例进行描述的,下面介绍客户端获取播放记录为例进行描述。

[0196] 所述客户端 1000,还用于向所述服务器端 2000 发送获取播放记录的请求,所述获取播放记录的请求中携带用户 ID、视频 ID 和视频类型;

[0197] 所述服务器端 2000,还用于根据所述用户 ID 从缓存中查找该用户 ID 对应的用户观看的视频 ID 的列表;如果缓存中查找不到该用户 ID 对应的用户观看的视频 ID 的列表再从数据库中查找;通过查找到的所述视频 ID 的列表,从缓存中获取视频的播放记录;如果缓存中没有所述播放记录再从数据库中获取;将所述播放记录返回给客户端。

[0198] 本发明以上实施例提供的方法、设备和系统,能够较快响应客户端的请求,可以在多个终端之间实现用户播放记录的同步。

[0199] 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。任何熟悉本领域的技术人员,在不脱离本发明技术方案范围情况下,都可利用上述揭示的方法和技术内容对本发明技术方案做出许多可能的变动和修饰,或修改为等同变化的等效实施例。因此,凡是未脱离本发明技术方案的内容,依据本发明的技术实质对以上实施例所做的任何简单修改、等同变化及修饰,均仍属于本发明技术方案保护的范围内。

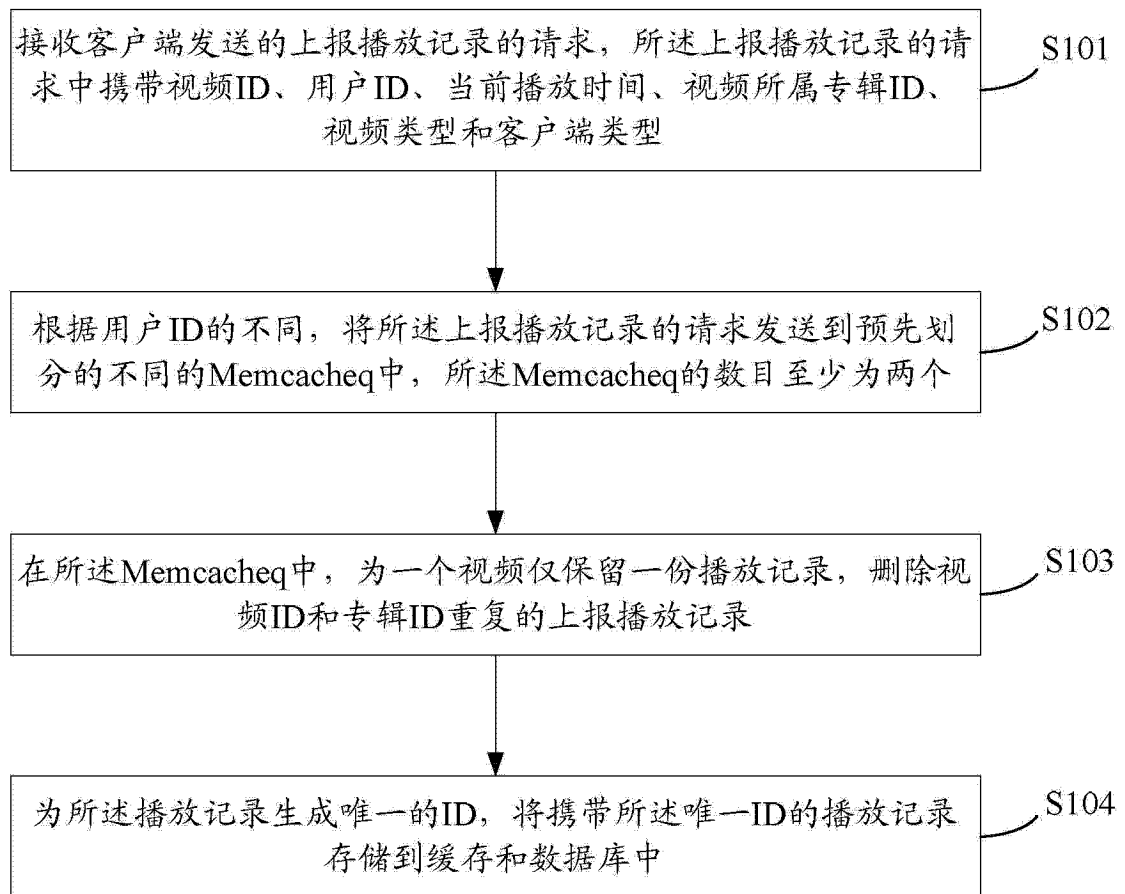


图 1

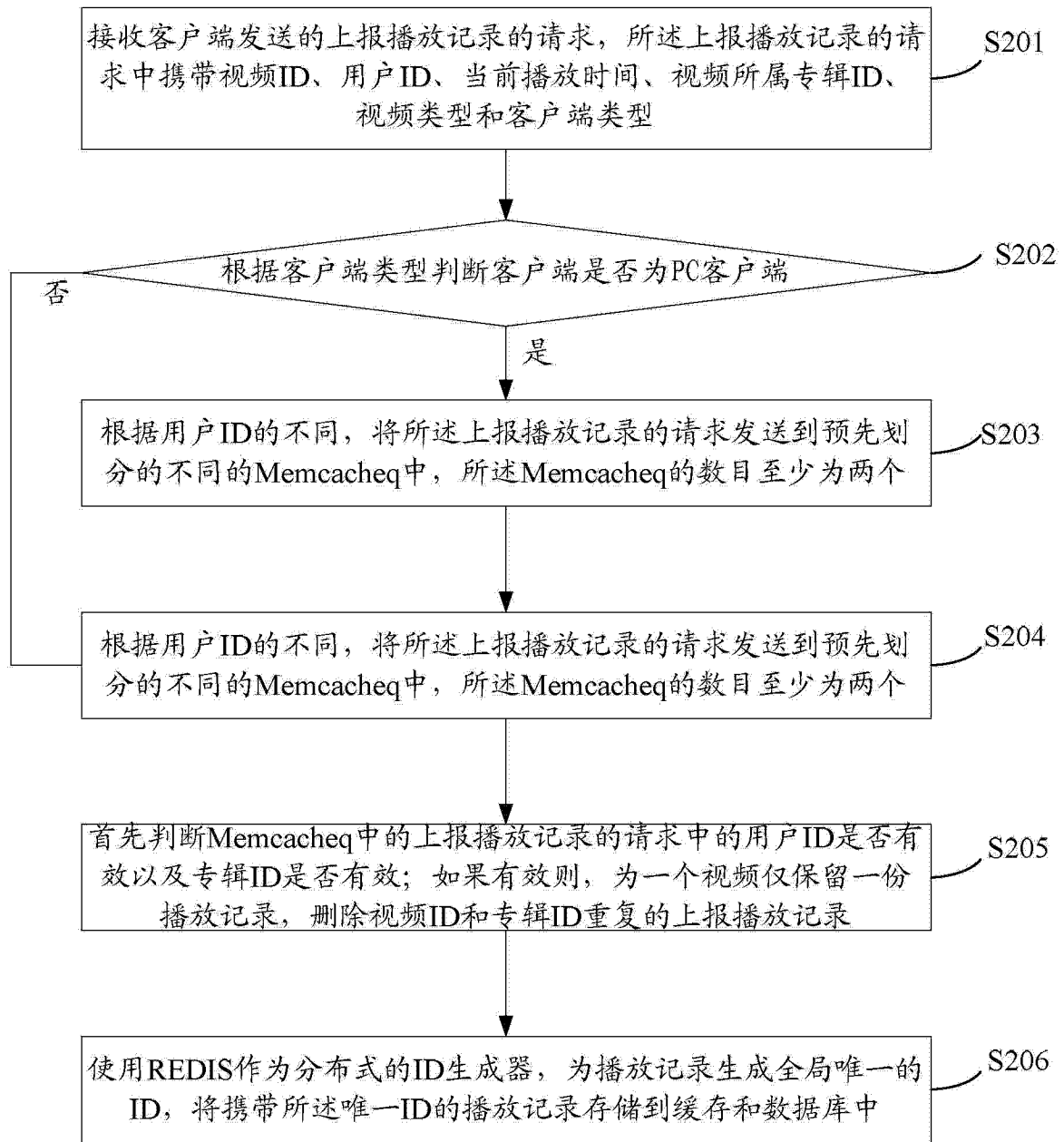


图 2

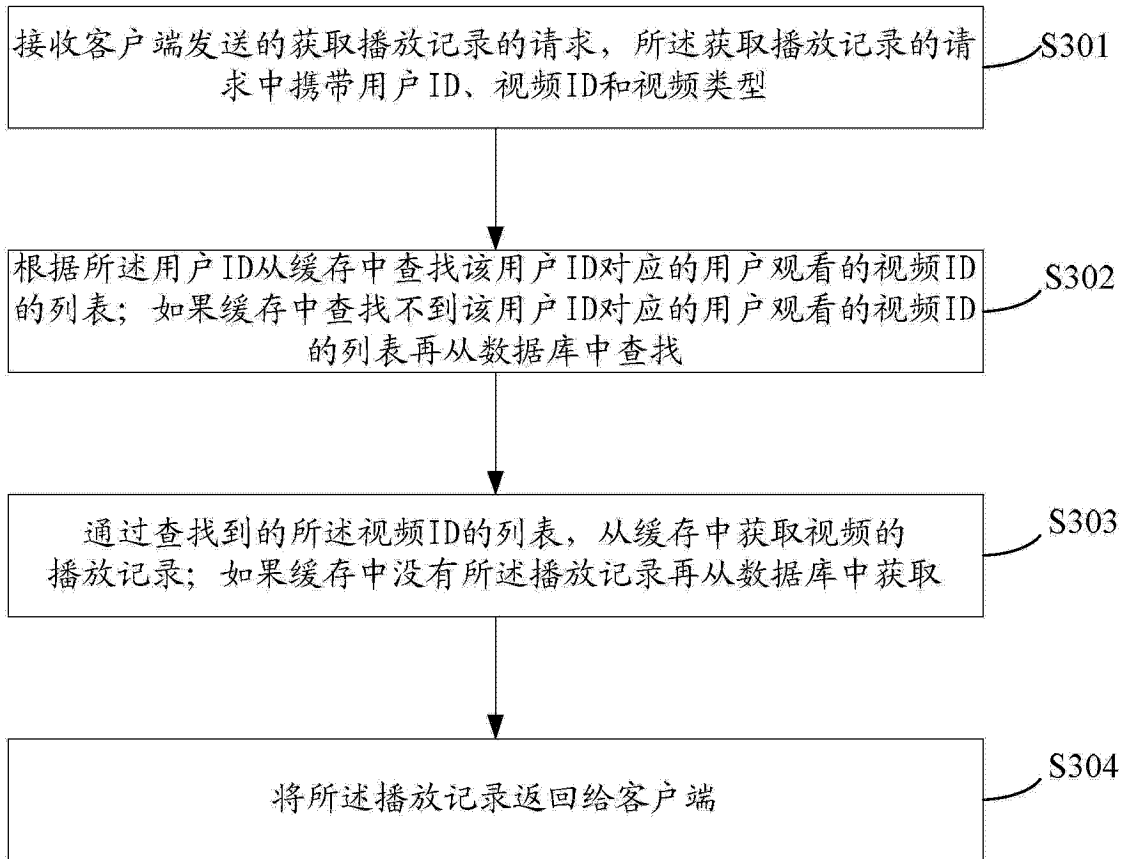


图 3

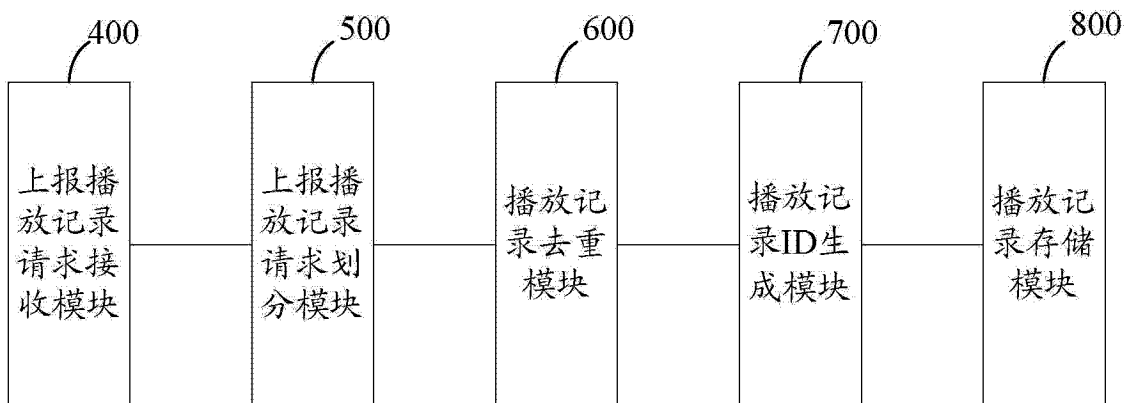


图 4

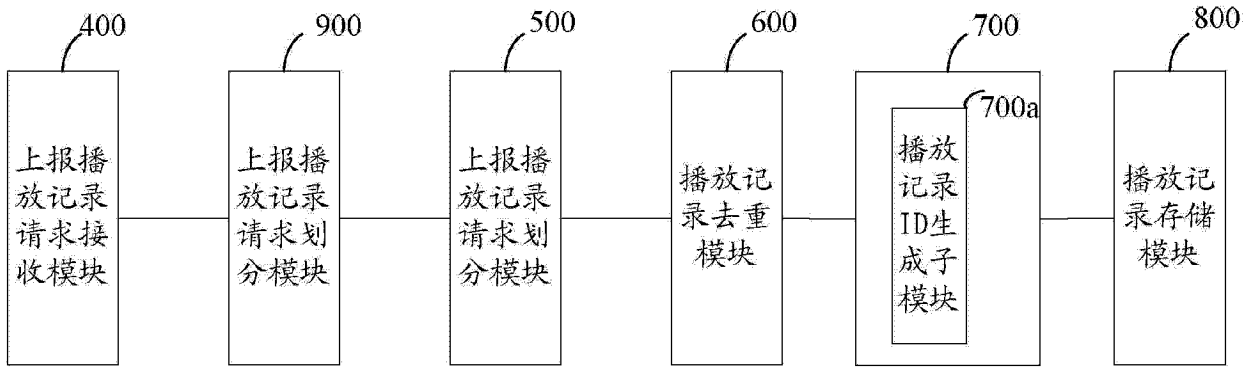


图 5

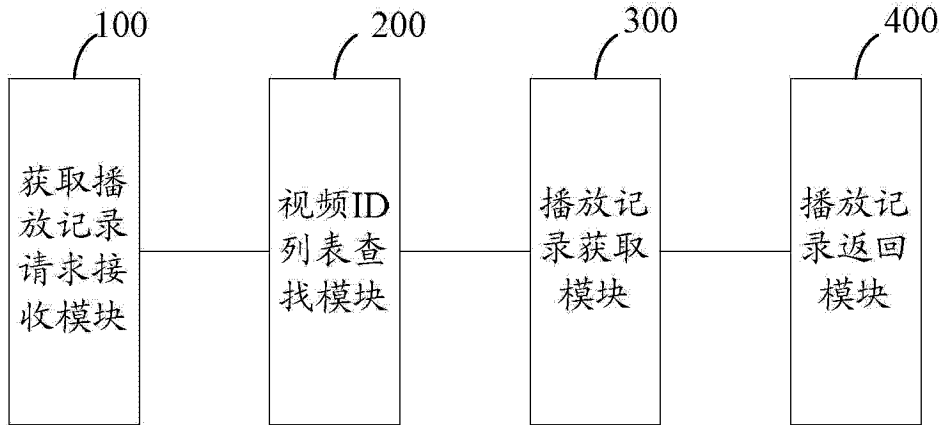


图 6

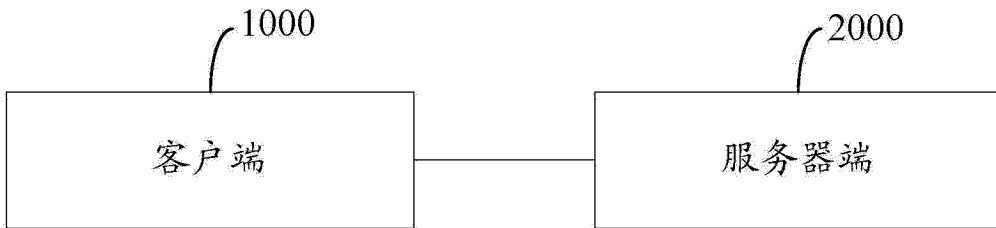


图 7