(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 101146127 B
(45)授权公告日 2010.06.09

(21)申请号 200710166042.2

(22)申请日 2007.10.30

(73)专利权人 金蝶软件（中国）有限公司
地址 518057 广东省深圳市南山区深南大道市高新技术产业园区 W1-B4

(72)发明人 殷慷 杨海悌

(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 逯长明

(51)Int.Cl.
H04L 29/08(2006.01)
G06F 9/44(2006.01)

(56)对比文件
CN 1454018 A,2003.11.05, 全文.
CN 1754155 A,2006.03.29, 全文.
JP 特开 2000-76168 A,2000.03.14, 全文.
CN 1513144 A,2004.07.14, 全文.

审查员 胡延

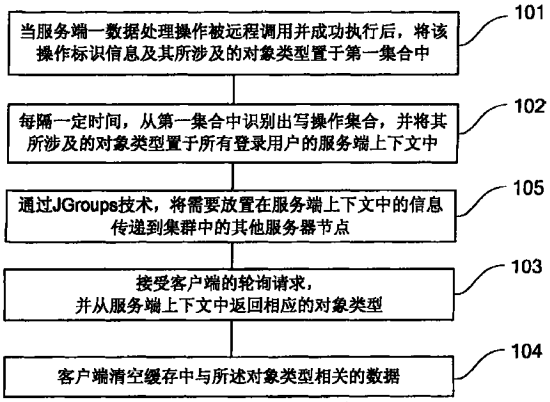
权利要求书 3 页 说明书 11 页 附图 5 页

(54)发明名称

一种分布式系统中客户端缓存更新的方法和装置

(57)摘要

本发明提供了一种更新分布式系统中客户端缓存数据的方法，包括：当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；每隔一定时间，从第一集合中识别出写操作集合，并将其所涉及的对象类型置于所有登录用户的服务端上下文中；接受客户端的轮询请求，并从服务端上下文中返回相应的对象类型；客户端清空缓存中与所述对象类型相关的数据；所述客户端的缓存数据按照对象类型进行分区管理。本发明通过在服务端对写方法进行监控、并由客户端发起轮询以及将客户端的缓存数据进行分区管理，这些措施的协同工作，可以确保连接到同一服务端的所有在线客户端的数据一致性。



CN 101146127 B

1. 一种更新分布式系统中客户端缓存数据的方法，其特征在于，包括：

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

每隔一定时间，从第一集合中识别出写操作集合，并将写操作集合所涉及的对象类型置于所有登录用户的服务端上下文中；

接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型；

客户端清空缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

2. 一种更新分布式系统中客户端缓存数据的方法，其特征在于，包括：

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型以及第二集合中的用户标识置于所有登录用户的服务端上下文中；

接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型和用户标识；

如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端清空缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

3. 如权利要求 2 所述的方法，其特征在于，当所述分布式系统采用服务器集群架构时，该方法还包括：

通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

4. 一种更新分布式系统中客户端缓存数据的方法，其特征在于，包括：

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

每隔一定时间，从第一集合中识别出写操作集合，将写操作集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；

接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识；

客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

5. 一种更新分布式系统中客户端缓存数据的方法，其特征在于，包括：

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识以及第二集合中的用户标识置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；

接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识和用户标识；

如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

6. 如权利要求 5 所述的方法，其特征在于，当所述分布式系统采用服务器集群架构时，该方法还包括：

通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

7. 一种更新分布式系统中客户端缓存数据的装置，其特征在于，包括：

位于服务端的操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

位于服务端的写操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，并将写操作集合所涉及的对象类型置于所有登录用户的服务端上下文中；

位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型；

位于客户端的清空模块，用于清空客户端缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

8. 一种更新分布式系统中客户端缓存数据的装置，其特征在于，包括：

位于服务端的普通操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

位于服务端的用户权限操作记录模块，用于当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

位于服务端的操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型以及第二集合中的用户标识置于所有登录用户的服务端上下文中；

位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型和用户标识；

位于客户端的清空模块，用于当客户端的登录用户标识存在于所收到的用户标识中时，则清空该客户端本地全部缓存数据；如果不存在，则客户端清空缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

9. 如权利要求 8 所述的装置，其特征在于，当所述分布式系统采用服务器集群架构时，该装置还包括：

位于服务端的信息通知模块，用于通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

10. 一种更新分布式系统中客户端缓存数据的装置，其特征在于，包括：

位于服务端的操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

位于服务端的写操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，将写操作集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识

置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；

位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识；

位于客户端的清空模块，用于依据所获得的缓存区标识，清空该客户端相应缓存区中的数据。

11. 一种更新分布式系统中客户端缓存数据的装置，其特征在于，包括：

位于服务端的普通操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

位于服务端的用户权限操作记录模块，用于当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

位于服务端的操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识以及第二集合中的用户标识置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；

位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识和用户标识；

位于客户端的清空模块，用于当客户端的登录用户标识存在于所收到的用户标识中时，则清空该客户端本地全部缓存数据；如果不存在，则客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

12. 如权利要求 11 所述的装置，其特征在于，当所述分布式系统采用服务器集群架构时，该装置还包括：

位于服务端的信息通知模块，用于通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

一种分布式系统中客户端缓存更新的方法和装置

技术领域

[0001] 本发明涉及分布式系统中的信息传输技术领域，特别是涉及一种分布式系统中更新客户端缓存的方法和装置。

背景技术

[0002] 分布式系统是支持分布式处理的系统，是在由通信网络互联的多处理机体系结构上执行任务的系统。对于部署在广域网上的大型分布式软件系统，一般分为客户端、服务端、数据库三层，业务数据一般存储在数据库中。客户端为了读取业务数据，需要与服务端进行远程方法调用，服务端再与数据库通讯，从中获取业务数据后，再进行若干处理，返回给客户端。

[0003] 在 Java 领域，所述远程方法调用是一种通信机制，可以在不同的 Java 虚拟机(JVM，java virtual machine)之间实现对象与对象的通信。JVM 可以位于相同或不同计算机上，在多个 JVM 中，一个 JVM 中的对象可以调用其它 JVM 中对象的方法。发起调用的 JVM 称为客户端，接受调用的 JVM 称为服务端。

[0004] 客户端与服务端的多次远程方法调用，属于多次广域网上的网络通讯，会受到网络延时及数据丢包的影响。通讯次数越多，数据量越大，受延时和丢包的影响就越大，总的响应时间就越不稳定。为了减少用户操作的等待时间，就需要减少客户端与服务端之间网络通讯的次数及数据量。为了减少客户端到服务端的远程方法调用，现有技术一般采用把部分不经常变化的业务数据缓存在客户端的方案。

[0005] 但是缓存在客户端的数据并不是真正固定不变的，例如，当某个客户端执行了写方法后（在远程调用方法中，读取服务端数据的方法称为读方法，修改服务端数据的方法称为写方法），缓存在其他在线客户端内存中的业务数据已经陈旧了，即与服务端的业务数据不再一致，因此需要更新。再例如，如果一个用户的权限发生变化，也需要更新这个用户登录的所有客户端的缓存数据，以防止此用户再访问自己无权访问的业务数据。

[0006] 而一般的现有技术都没有相应的更新机制，即使各个在线的客户端中缓存的业务数据存在不一致的情况，需要更新缓存的客户端也只有重新登录，才可以获取最新的业务数据，但该客户端实际上是无法知悉什么时候需要更新缓存，所以上述既会使得用户操作非常不方便（需要多次重新登录），同时也无法保证缓存数据的及时更新。

[0007] 在现有技术的另一解决方案中，在每个客户端提供了一个单独的“缓存刷新”功能，用户触发该功能，就可以直接实现手动的缓存更新，这个方案虽然比上述的重新登录方式要好一些，但是仍然存在不方便的问题，并且没有从根本上解决问题，因为手动操作仍然会带来效率缺陷。

[0008] 总之，需要本领域技术人员迫切解决的一个技术问题就是：如何能够实现在分布式系统中及时方便的更新客户端的缓存数据。

发明内容

[0009] 本发明所要解决的技术问题是提供一种在分布式系统中更新客户端的缓存数据的方法和系统，能够及时方便的实现各个客户端缓存数据的更新，确保连接到同一服务端的所有在线客户端的缓存数据一致性。

[0010] 为了解决上述问题，本发明公开了一种更新分布式系统中客户端缓存数据的方法，包括：当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；每隔一定时间，从第一集合中识别出写操作集合，并将写操作集合所涉及的对象类型置于所有登录用户的服务端上下文中；接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型；客户端清空缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

[0011] 依据本发明的另一实施例，还公开了一种更新分布式系统中客户端缓存数据的方法，包括：当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型以及第二集合中的用户标识置于所有登录用户的服务端上下文中；接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型和用户标识；如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端清空缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

[0012] 优选的，当所述分布式系统采用服务器集群架构时，该方法还包括：通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

[0013] 依据本发明的另一实施例，还公开了一种更新分布式系统中客户端缓存数据的方法，包括：当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；每隔一定时间，从第一集合中识别出写操作集合，将写操作集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识；客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

[0014] 依据本发明的另一实施例，还公开了一种更新分布式系统中客户端缓存数据的方法，包括：当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识以及第二集合中的用户标识置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识和用户标识；如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

[0015] 优选的，当所述分布式系统采用服务器集群架构时，该方法还包括：通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

[0016] 依据本发明的另一实施例，还公开了一种更新分布式系统中客户端缓存数据的装

置，包括：

[0017] 位于服务端的操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

[0018] 位于服务端的写操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，并将写操作集合所涉及的对象类型置于所有登录用户的服务端上下文中；

[0019] 位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

[0020] 位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型；

[0021] 位于客户端的清空模块，用于清空客户端缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

[0022] 依据本发明的另一实施例，还公开了一种更新分布式系统中客户端缓存数据的装置，包括：

[0023] 位于服务端的普通操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

[0024] 位于服务端的用户权限操作记录模块，用于当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

[0025] 位于服务端的操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型以及第二集合中的用户标识置于所有登录用户的服务端上下文中；

[0026] 位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

[0027] 位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回写操作集合所涉及的对象类型和用户标识；

[0028] 位于客户端的清空模块，用于当客户端的登录用户标识存在于所收到的用户标识中时，则清空该客户端本地全部缓存数据；如果不存在，则客户端清空缓存中与所述对象类型相关的数据，所述客户端的缓存数据按照对象类型进行分区管理。

[0029] 优选的，当所述分布式系统采用服务器集群架构时，该装置还包括：位于服务端的信息通知模块，用于通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

[0030] 依据本发明的另一实施例，还公开了一种更新分布式系统中客户端缓存数据的装置，包括：

[0031] 位于服务端的操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

[0032] 位于服务端的写操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，将写操作集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；

[0033] 位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

[0034] 位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识；

[0035] 位于客户端的清空模块，用于依据所获得的缓存区标识，清空该客户端相应缓存区中的数据。

[0036] 依据本发明的另一实施例，还公开了一种更新分布式系统中客户端缓存数据的装置，包括：

[0037] 位于服务端的普通操作记录模块，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及该操作所涉及的对象类型置于第一集合中；

[0038] 位于服务端的用户权限操作记录模块，用于当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

[0039] 位于服务端的操作识别模块，用于每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识以及第二集合中的用户标识置于所有登录用户的服务端上下文中，所述客户端的缓存数据按照对象类型进行分区管理；

[0040] 位于客户端的轮询模块，用于每隔一定时间发起缓存更新的轮询请求；

[0041] 位于服务端的上下文模块，用于接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识和用户标识；

[0042] 位于客户端的清空模块，用于当客户端的登录用户标识存在于所收到的用户标识中时，则清空该客户端本地全部缓存数据；如果不存在，则客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

[0043] 优选的，当所述分布式系统采用服务器集群架构时，该装置还包括：位于服务端的信息通知模块，用于通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

[0044] 与现有技术相比，本发明具有以下优点：

[0045] 本发明通过在服务端对写方法进行监控、并由客户端发起轮询以及将客户端的缓存数据进行分区管理，这些措施的协同工作，可以确保连接到同一服务端的所有在线客户端的数据一致性。并且，本发明所公开的缓存更新机制，对具体的客户端和服务端业务逻辑是透明的；也就是说，远程方法调用发起者和服务者，都不需要维护缓存的更新，非常的简单方便。

[0046] 另外，针对由多个应用服务器组成集群架构的分布式系统而言，本发明也可以通过 JGroups 组播技术，以确保连接到集群中不同服务端的在线客户端的数据一致性。

附图说明

[0047] 图 1 是本发明一种更新分布式系统中客户端缓存数据的方法实施例 1 的步骤流程图；

[0048] 图 2 是本发明一种更新分布式系统中客户端缓存数据的方法实施例 2 的步骤流程图；

[0049] 图 3 是本发明一种更新分布式系统中客户端缓存数据的方法实施例 3 的步骤流程图；

[0050] 图 4 是实施例 3 中两个线程的协作示意图；

[0051] 图 5 是本发明一种更新分布式系统中客户端缓存数据的方法实施例 4 的步骤流程

图；

[0052] 图 6 是本发明一种更新分布式系统中客户端缓存数据的装置实施例 1 的结构框图；

[0053] 图 7 是本发明一种更新分布式系统中客户端缓存数据的装置实施例 2 的结构框图；

[0054] 图 8 是本发明一种更新分布式系统中客户端缓存数据的装置实施例 3 的结构框图；

[0055] 图 9 是本发明一种更新分布式系统中客户端缓存数据的装置实施例 4 的结构框图。

具体实施方式

[0056] 为使本发明的上述目的、特征和优点能够更加明显易懂，下面结合附图和具体实施方式对本发明作进一步详细的说明。

[0057] 本发明可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本发明，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0058] 需要说明的是，所述远程方法调用是一种计算机之间对象互相调用对方函数，启动对方进程的一种机制，使用这种机制，某一台计算机上的对象在调用另外一台计算机上的方法时，使用的程序语法规则和在本地机上对象间的方法调用的语法规则一样。这种方式允许程序可以利用分布式计算将工作量分散到多个虚拟机上。远程方法调用使用户能访问在另一主机上的对象，并远程调用其方法。总之，这种机制给分布计算的系统设计、编程都带来了极大的方便。其中的“方法”一词属于本领域的通用术语，在本发明中，为了避免和传统意义上的“方法”一词产生混淆，因此，在本发明的权利要求中将其转换为“数据处理操作”或者“操作”一词加以描述，实际上，本领域技术人员应该知悉，二者仅仅是文字上的转换，其内涵和外延都是一样的。在说明书的描述中，为了适合本领域技术人员阅读，仍然采用“方法”一词进行描述。

[0059] 本发明的核心思想之一是：在服务端对所有的方法调用进行监控，如果是写方法，则把服务端对象类型写入当前所有登录用户的服务端上下文中，等待客户端轮询时取回类型，并清空对应类型的客户端缓存区。如果服务端存在集群部署，还可以利用 JGroups 将对象类型通知集群中的所有服务器节点。此方案对具体的服务端业务逻辑而言是透明的，因而无论是远程调用的发起者还是服务者，都不需要维护缓存的更新。

[0060] 参照图 1，示出了本发明一种更新分布式系统中客户端缓存数据的方法实施例 1，可以包括以下步骤：

[0061] 步骤 101、当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中。

[0062] 在后面的详细说明中，将会直接采用“方法”一词代替“操作”进行描述。步骤 101 中的操作标识信息一般的可以是方法名称或者序号等等。步骤 101 可以针对服务端的任意

一个接口方法的成功执行。

[0063] 步骤 102、每隔一定时间，从第一集合中识别出写操作集合，并将其所涉及的对象类型置于所有登录用户的服务端上下文中；对于识别写操作的方式可以有很多，简单而言，就可以通过名称加以确定。

[0064] 步骤 103、接受客户端的轮询请求，并从服务端上下文中返回相应的对象类型。

[0065] 步骤 104、客户端清空缓存中与所述对象类型相关的数据；所述客户端的缓存数据按照对象类型进行分区管理。

[0066] 缓存 (Cache) 是一种在软硬件系统中广泛使用的提升响应性能的技术，主要原理是，从响应较慢的存储容器读取数据后，临时保存在响应较快的存储容器中，下次再访问相同的数据时，优先从较快的容器中加载。为了能够仅仅依据返回的对象类型就完成相应缓存数据的清除，本发明中客户端的缓存数据是按照对象类型进行分区管理的。根据远程对象的类型，客户端缓存划分为多个缓存区 (Cache Region)，例如期间、公司、客户、供应商等等多个缓存区。某个类型的远程对象是否参与客户端缓存，可以在运行期由配置文件指定。

[0067] 需要更新的缓存数据被清空后，当客户端需要调用该部分数据时，本地缓存中没有，就可以从服务端获取最新的数据了，从而实现了缓存数据的更新。

[0068] 其中，步骤 102 可以作为服务端的一个线程循环执行，而客户端发起的轮询请求和步骤 104 则可以作为客户端的一线程循环执行。

[0069] 本发明中的“对象”一词也是本领域的通用术语，在本说明书中就不对其进行详细解释了。一般的，在 Java 中，一个类，一个方法，一个变量都可以作为对象，其中，有些对象可以直接去用（比如基本变量类型，或一些静态的类、方法、变量等），而有些对象不可以直接去用，需要创建这个对象的实例（这样，既能实现对象的功能，又不会直接破坏对象的构造）。

[0070] 步骤 102 和 103 中的采用的“上下文”也是本领域的一个通用术语，在此也不准备详细描述。上下文其实是一个抽象的概念，上下文就是一个对象所处的环境，上下文服务就是对象在特定环境下所能得到的服务和它所能提供的服务。可以理解为，上下文屏蔽了服务端线程和客户端线程的差异，使得业务逻辑的执行好像是在一个线程中执行的。

[0071] 例如，访问 JNDI 的 Context（上下文），其上层是 JNDI 服务器（可能是远程的），下层是客户的应用程序，其作用就是建立一个通道让你能访问 JNDI 服务器，同时也让 JNDI 服务器接受客户端的请求，起到交互（或者说通道）作用。

[0072] 在本发明的另一优选实施例中，所述分布式系统采用了服务器集群架构，则该方法还可以包括步骤 105：通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。JGroups 是一个开源的纯 Java 的可靠的群组通讯工具。其工作模式基于 IP 多播，但可以在可靠性和群组成员管理上进行扩展。

[0073] 集群是一种现有技术：它将多个系统连接到一起，使多个系统能够像一个系统那样工作。采用集群通常是为了提高系统的稳定性和网络中心的数据处理能力及服务能力，能够提供高可用性和可伸缩性。本发明中的集群，主要指多个应用服务器组成的集群。

[0074] 在本实施例中，和现有技术相比，增加了客户端远程方法调用轮询，虽然增加了总的网络通讯次数，但是由于这些通讯一般不在用户的等待时间内（例如用户输入信息或者浏览信息时，在后台执行），而且通讯数据量非常小（仅仅传输标识或者类型即可），所以在

不影响用户性能感受的基础上，还可以及时（例如，设置时间间隔为 30 秒）方便的实现客户端缓存的更新。

[0075] 参照图 2，示出了一种更新分布式系统中客户端缓存数据的方法实施例 2，可以包括以下步骤：

[0076] 步骤 201、当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；

[0077] 步骤 202、当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

[0078] 明显的，步骤 201 和 202 之间并没有必然的先后顺序。

[0079] 步骤 203、每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型以及第二集合中的用户标识置于所有登录用户的服务端上下文中；

[0080] 步骤 204、接受客户端的轮询请求，并从服务端上下文中返回相应的对象类型和用户标识；

[0081] 步骤 205、如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端清空缓存中与所述对象类型相关的数据；所述客户端的缓存数据按照对象类型进行分区管理。

[0082] 本发明的实施例 2 和实施例 1 的区别在于，实施例 2 除了针对写方法的缓存更新，还进一步包括了在用户权限发生变化的情况下，对客户端相应缓存数据的更新。即如果某个客户端登录用户的权限发生了变化，则需要及时清空该客户端中的所有缓存数据；而对于多个在线客户端轮询而言，实际上就能够及时清空权限发生变化的用户登录的所有客户端中的缓存数据。

[0083] 进一步，当所述分布式系统采用服务器集群架构时，该方法实施例还可以包括：通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。需要说明的是，步骤 205 与其他步骤之间可以采用并行的方式，并不具有必然的先后顺序。

[0084] 下面通过一个具体例子，并将上面的整个流程分为在服务端执行的步骤和在客户端执行的步骤两部分进行描述，以更清楚的说明本发明解决方案的实现。

[0085] 服务端执行步骤：

[0086] (1) 启动应用服务器时，启动一个缓存清除通告线程 CacheCleanSender，并加载缓存配置文件。所述配置文件主要用于标志哪些实体的哪些方法需要参与缓存，以及这些方法的意图是读还是写等。

[0087] (2) 服务端任意一个接口方法成功执行后，将服务端对象类型及方法名注册到 CacheCleanSender 的“未处理方法集合”。

[0088] (3) 服务端执行了修改用户权限、组织范围、启用禁用用户等操作后，需要将用户 ID 注册到 CacheCleanSender 的“需清空缓存用户 ID 集合”；

[0089] (4)CacheCleanSender 每隔一段时间（默认 30 秒），根据缓存配置文件，处理“需清空缓存用户 id 集合”和“未处理方法集合”。将待清空类型集合和待清空缓存用户 ID 写到所有登录用户的服务端上下文中。

[0090] (5)CacheCleanSender 接收到集群中其它服务器的待清空类型集合或用户 ID 集合消息后，需要做的工作类似于上述周期性工作。

[0091] (6) 服务端对象 CacheCleanFacade，负责处理客户端的轮询，从服务端上下文中返回缓存更新信息给客户端，然后由客户端清空相关信息。

[0092] 客户端执行的步骤：

[0093] (7) 启动客户端时，启动一个缓存清除轮询线程 CacheCleanReceiver，并载入缓存配置文件。

[0094] (8)CacheCleanReceiver 每隔一段时间（默认 30 秒），远程调用服务端的 CacheCleanFacade 方法。如果是用户权限变化，则清空本地全部缓存区；如果是类型集合，则清空对应类型的缓存区。

[0095] 参照图 3，示出了一种更新分布式系统中客户端缓存数据的方法实施例 3，包括以下步骤：

[0096] 步骤 301、当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；

[0097] 步骤 302、每隔一定时间，从第一集合中识别出写操作集合，将其所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识置于所有登录用户的服务端上下文中；所述客户端的缓存数据按照对象类型进行分区管理；

[0098] 步骤 303、接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识；

[0099] 步骤 304、客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

[0100] 实施例 3 和实施例 1 的区别在于，在实施例 3 中，服务端上下文中注册的是缓存区 ID，客户端可以直接清空相应缓存区即可，而不需要像实施例 1 中还需要执行对象类型和缓存区地址 ID 之间的匹配过程。

[0101] 为了更清晰的说明本实施例，下面同时结合图 4 进行说明，图 4 示出了两个线程之间的协作图。本实施例主要可以涉及以下两个线程。

[0102] 服务端清空通告线程：根据缓存配置文件，从未处理方法集合中区分出写方法集合，并进一步解释为待清空的客户端缓存区 id，并注册到当前所有在线客户端的服务端上下文中。优选的，还需要通过 JGroups，将待清空的客户端缓存区 id 传递到集群中的其他应用服务器节点。

[0103] 客户端轮询线程：周期性地从服务端上下文取回待清空缓存区 id 集合，清空本地客户端对应的缓存区。

[0104] 下面的描述是顺着图 4 中的箭头方向进行描述的，首先，图 4 中的客户端存根 401 向服务端框架 402 进行远程方法调用，然后服务端框架 402 将其注册到未处理方法集合中；进而，服务端清空通告线程 403 循环执行，不断的从未处理方法集合中识别出写方法，然后将相应的缓存区 id 注册到待清空缓存区集合，该集合位于服务端上下文 404 中；接着，当客户端轮询线程 405 发起轮询请求时，从服务端上下文 404 获取相应的缓存区 id，然后清空客户端缓存 406 对应的缓存区，即完成了整个流程。实际上，为了保证持续更新，上面的两个线程 403 和 405 是每隔一定时间循环执行的。

[0105] 存根 (Stub) 与框架 (Skeleton) 属于本领域的术语，简单介绍如下：

[0106] 在远程方法调用中，可以把远程对象像本地对象一样使用，应用程序并不知道一个对象是远程的还是本地的。远程方法调用时，系统通过远程代理自动拦截方法调用，找到服务端远程对象并调用它的方法，这一机制就是通过存根 (Stub) 与框架 (Skeleton) 实现

的。Stub 是客户端对象，是服务端对象的远程代理；Skeleton 是服务端对象，接收 Stub 的远程调用请求，实际地调用服务端对象方法，最后把方法返回值写回给 Stub。

[0107] 参照图 5，示出了一种更新分布式系统中客户端缓存数据的方法实施例 4，可以包括以下步骤：

[0108] 步骤 501、当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；

[0109] 步骤 502、当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

[0110] 步骤 503、每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识以及第二集合中的用户标识置于所有登录用户的服务端上下文中；所述客户端的缓存数据按照对象类型进行分区管理；

[0111] 步骤 504、接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识和用户标识；

[0112] 步骤 505、如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

[0113] 当然，如果从服务端上下文中没有收到用户标识，则说明在当前循环时间段之内，没有用户的权限发生变化；实际上，该情况属于客户端的登录用户标识不存在于所接收的信息中的特殊情形。进一步查看所接收的缓存区标识，如果也没有，则说明在当前循环时间段之内，没有写方法执行，不需要更新缓存。本段的描述也适用于前述的各个实施例。

[0114] 实施例 4 和实施例 2 的区别在于，在实施例 4 中，服务端上下文中注册的是缓存区 ID，客户端可以直接清空相应缓存区即可，而不需要像实施例 2 中还需要执行对象类型和缓存区地址 ID 之间的匹配过程。

[0115] 进一步，当所述分布式系统采用服务器集群架构时，该实施例还可以包括：通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

[0116] 参照图 6，示出了一种更新分布式系统中客户端缓存数据的装置实施例 1，包括以下模块：

[0117] 位于服务端的操作记录模块 601，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；

[0118] 位于服务端的写操作识别模块 602，用于每隔一定时间，从第一集合中识别出写操作集合，并将其所涉及的对象类型置于所有登录用户的服务端上下文中；

[0119] 位于客户端的轮询模块 603，用于每隔一定时间发起缓存更新的轮询请求；

[0120] 位于服务端的上下文模块 604，用于接受客户端的轮询请求，并从服务端上下文中返回相应的对象类型；

[0121] 位于客户端的清空模块 605，用于清空客户端缓存中与所述对象类型相关的数据；所述客户端的缓存数据按照对象类型进行分区管理。

[0122] 由于装置实施例 1 是按照方法实施例 1 的步骤流程基本对应的方式描述的，因此不再赘述。

[0123] 参照图 7，示出了一种更新分布式系统中客户端缓存数据的装置实施例 2，包括以下模块：

[0124] 位于服务端的普通操作记录模块 701，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；

[0125] 位于服务端的用户权限操作记录模块 702，用于当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

[0126] 位于服务端的操作识别模块 703，用于每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型以及第二集合中的用户标识置于所有登录用户的服务端上下文中；

[0127] 位于客户端的轮询模块 704，用于每隔一定时间发起缓存更新的轮询请求；

[0128] 位于服务端的上下文模块 705，用于接受客户端的轮询请求，并从服务端上下文中返回相应的对象类型和用户标识；

[0129] 位于客户端的清空模块 706，用于当客户端的登录用户标识存在于所收到的用户标识中时，则清空该客户端本地全部缓存数据；如果不存在，则客户端清空缓存中与所述对象类型相关的数据；所述客户端的缓存数据按照对象类型进行分区管理。

[0130] 进一步，当所述分布式系统采用服务器集群架构时，该装置还包括：位于服务端的信息通知模块 707，用于通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。实际上，所述信息通知模块 707 可以应用在本发明的各个装置实施例中。

[0131] 由于装置实施例 2 是按照方法实施例 2 的步骤流程基本对应的方式描述的，因此不再赘述。

[0132] 参照图 8，示出了一种更新分布式系统中客户端缓存数据的装置实施例 3，包括以下模块：

[0133] 位于服务端的操作记录模块 801，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；

[0134] 位于服务端的写操作识别模块 802，用于每隔一定时间，从第一集合中识别出写操作集合，将其所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识置于所有登录用户的服务端上下文中；所述客户端的缓存数据按照对象类型进行分区管理；

[0135] 位于客户端的轮询模块 803，用于每隔一定时间发起缓存更新的轮询请求；

[0136] 位于服务端的上下文模块 804，用于接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识；

[0137] 位于客户端的清空模块 805，用于依据所获得的缓存区标识，清空该客户端相应缓存区中的数据。

[0138] 进一步，当所述分布式系统采用服务器集群架构时，该装置还包括：位于服务端的信息通知模块 806，用于通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

[0139] 由于装置实施例 3 是按照方法实施例 3 的步骤流程基本对应的方式描述的，因此不再赘述。

[0140] 参照图 9，示出了一种更新分布式系统中客户端缓存数据的装置实施例 4，包括以下模块：

[0141] 位于服务端的普通操作记录模块 901，用于当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中；

[0142] 位于服务端的用户权限操作记录模块 902，用于当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中；

[0143] 位于服务端的操作识别模块 903，用于每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识以及第二集合中的用户标识置于所有登录用户的服务端上下文中；

[0144] 位于客户端的轮询模块 904，用于每隔一定时间发起缓存更新的轮询请求；

[0145] 位于服务端的上下文模块 905，用于接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识和用户标识；

[0146] 位于客户端的清空模块 906，用于当客户端的登录用户标识存在于所收到的用户标识中时，则清空该客户端本地全部缓存数据；如果不存在，则客户端依据所获得的缓存区标识，清空相应缓存区中的数据。

[0147] 优选的，当所述分布式系统采用服务器集群架构时，该装置实施例还可以包括：位于服务端的信息通知模块，用于通过 JGroups 技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点。

[0148] 由于装置实施例 4 是按照方法实施例 4 的步骤流程基本对应的方式描述的，因此不再赘述。

[0149] 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。对于系统实施例而言，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。

[0150] 以上对本发明所提供的一种更新分布式系统中客户端缓存数据的方法和装置，进行了详细介绍，本文中应用了具体个例对本发明的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本发明的方法及其核心思想；同时，对于本领域的一般技术人员，依据本发明的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本发明的限制。

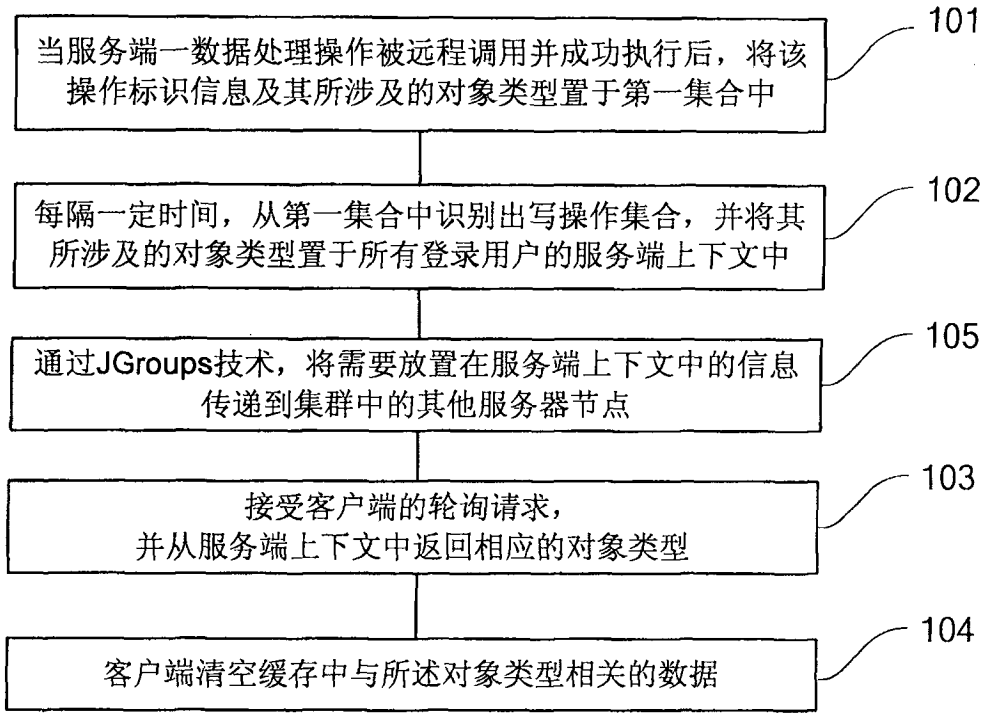

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中
101
每隔一定时间，从第一集合中识别出写操作集合，并将其所涉及的对象类型置于所有登录用户的服务端上下文中
102
通过JGroups技术，将需要放置在服务端上下文中的信息传递到集群中的其他服务器节点
105
接受客户端的轮询请求，并从服务端上下文中返回相应的对象类型
103
客户端清空缓存中与所述对象类型相关的数据
104

图 1

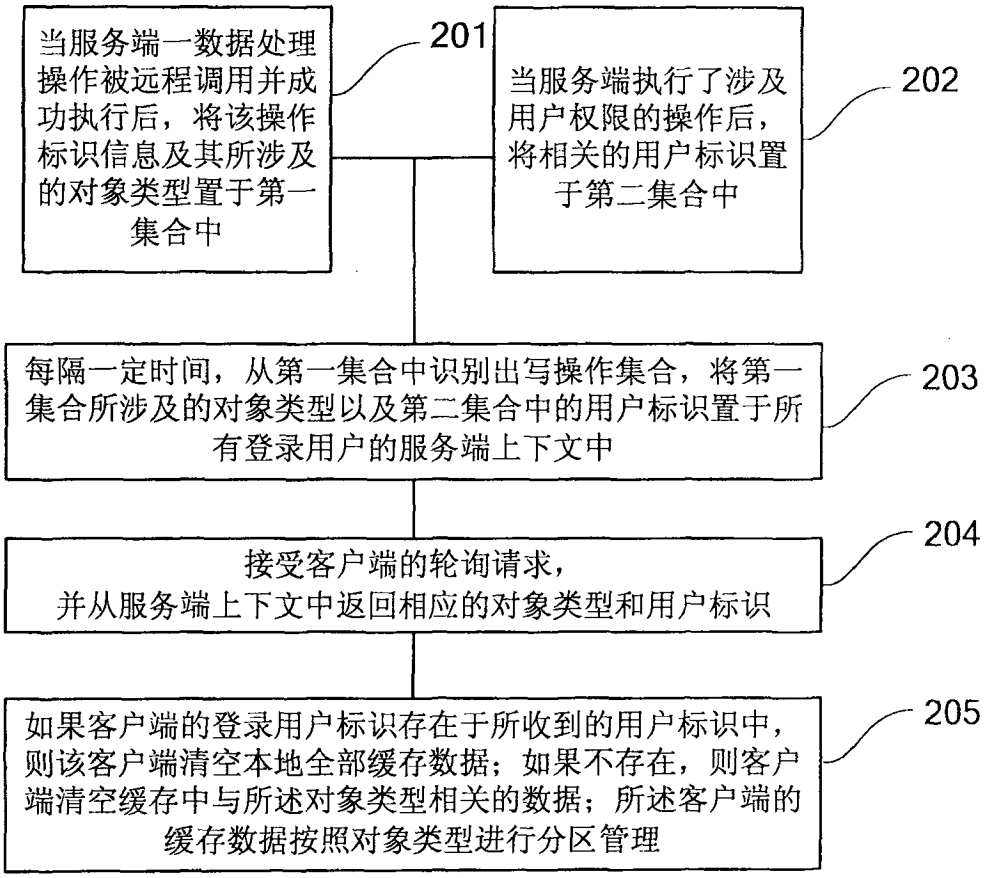

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中
201
当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中
202
每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型以及第二集合中的用户标识置于所有登录用户的服务端上下文中
203
接受客户端的轮询请求，并从服务端上下文中返回相应的对象类型和用户标识
204
如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端清空缓存中与所述对象类型相关的数据；所述客户端的缓存数据按照对象类型进行分区管理
205

图 2

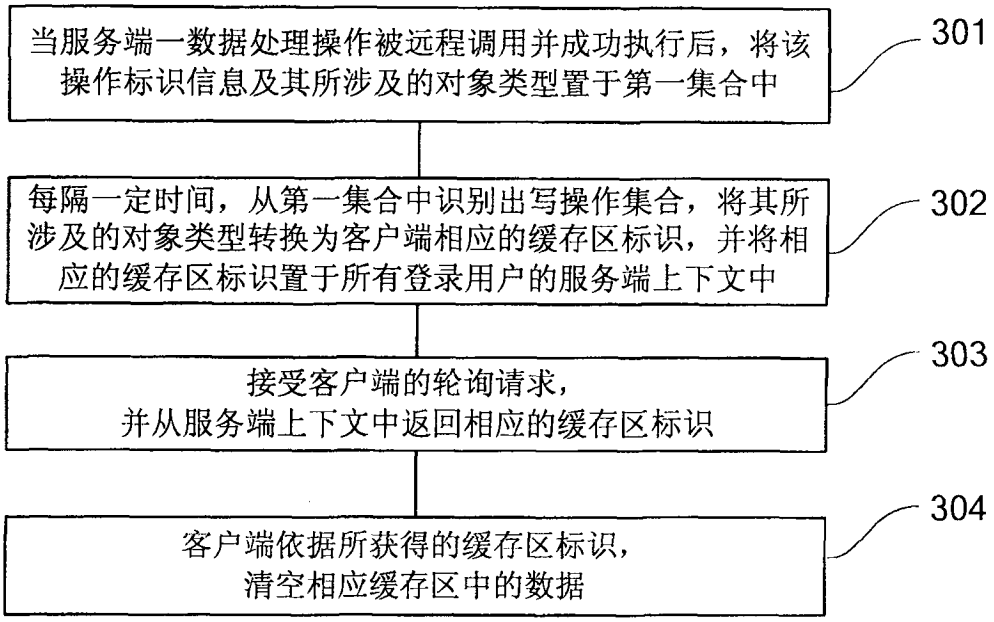

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中
301
每隔一定时间，从第一集合中识别出写操作集合，将其所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识置于所有登录用户的服务端上下文中
302
接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识
303
客户端依据所获得的缓存区标识，清空相应缓存区中的数据
304

图 3

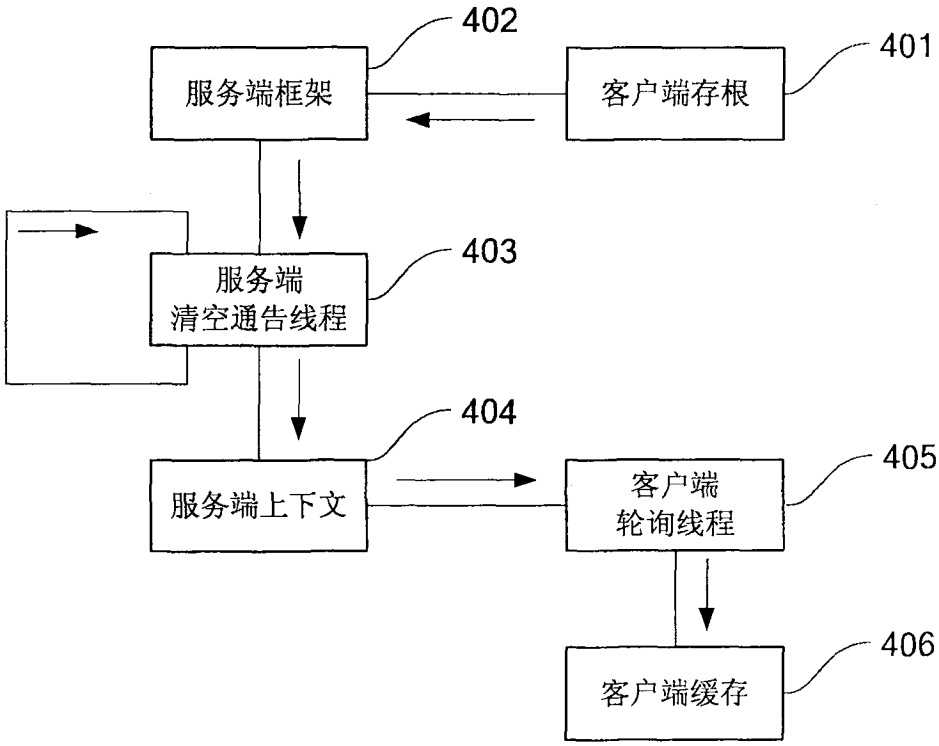

402
服务端框架
客户端存根
401
服务端
清空通告线程
403
404
服务端上下文
客户端
轮询线程
405
客户端缓存
406

图 4

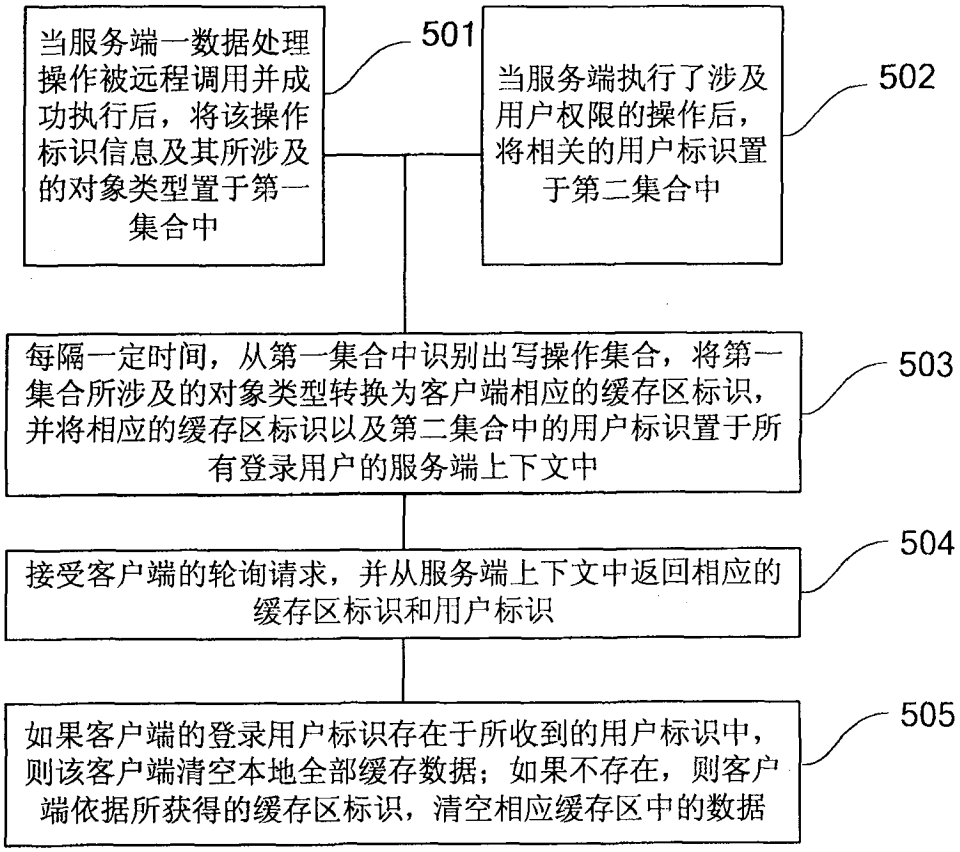

当服务端一数据处理操作被远程调用并成功执行后，将该操作标识信息及其所涉及的对象类型置于第一集合中
501
当服务端执行了涉及用户权限的操作后，将相关的用户标识置于第二集合中
502
每隔一定时间，从第一集合中识别出写操作集合，将第一集合所涉及的对象类型转换为客户端相应的缓存区标识，并将相应的缓存区标识以及第二集合中的用户标识置于所有登录用户的服务端上下文中
503
接受客户端的轮询请求，并从服务端上下文中返回相应的缓存区标识和用户标识
504
如果客户端的登录用户标识存在于所收到的用户标识中，则该客户端清空本地全部缓存数据；如果不存在，则客户端依据所获得的缓存区标识，清空相应缓存区中的数据
505

图 5

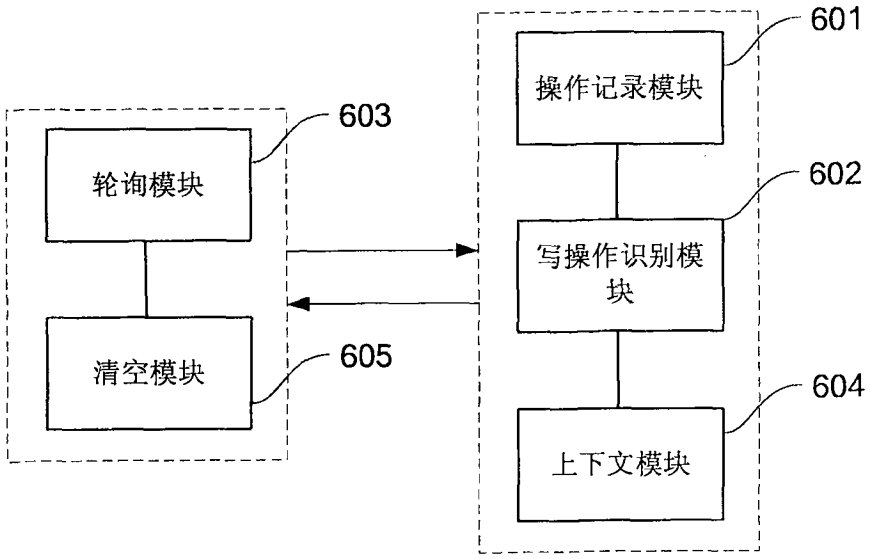

601
操作记录模块
602
写操作识别模块
603
轮询模块
604
上下文模块
605
清空模块

图 6

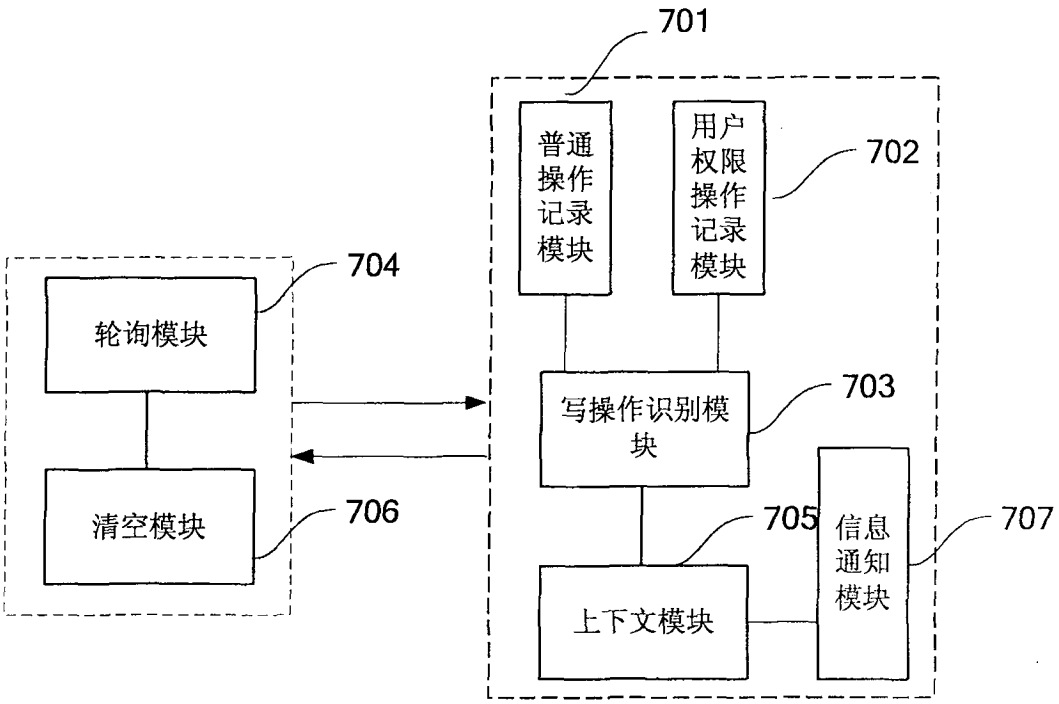

701
普通操作记录模块
用户权限操作记录模块
702
704
轮询模块
703
写操作识别模块
清空模块
706
705
信息通知模块
707
上下文模块

图 7

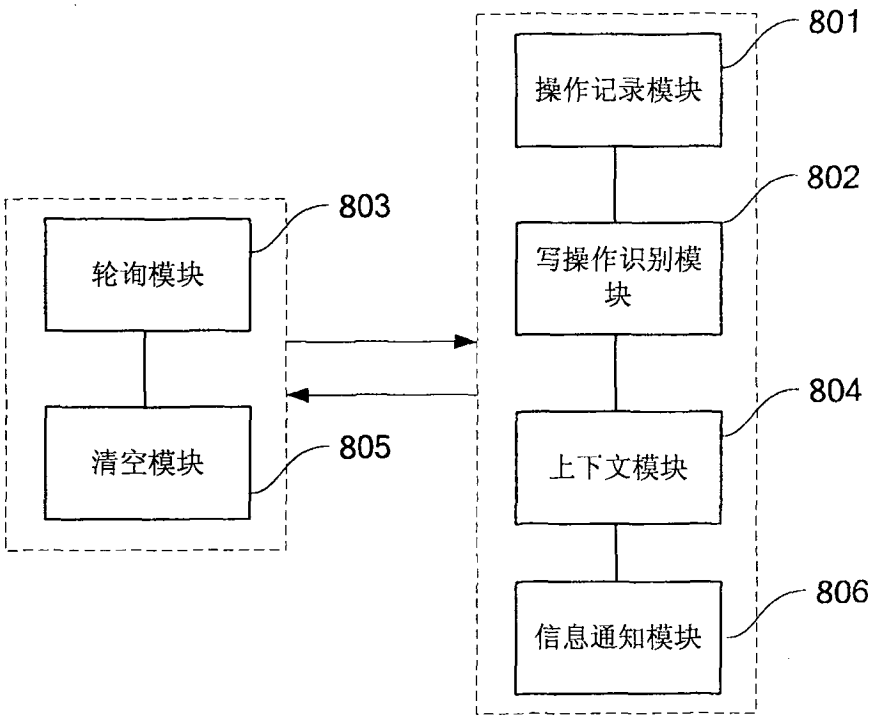

801
操作记录模块
802
803
轮询模块
写操作识别模块
804
清空模块
805
上下文模块
806
信息通知模块

图 8

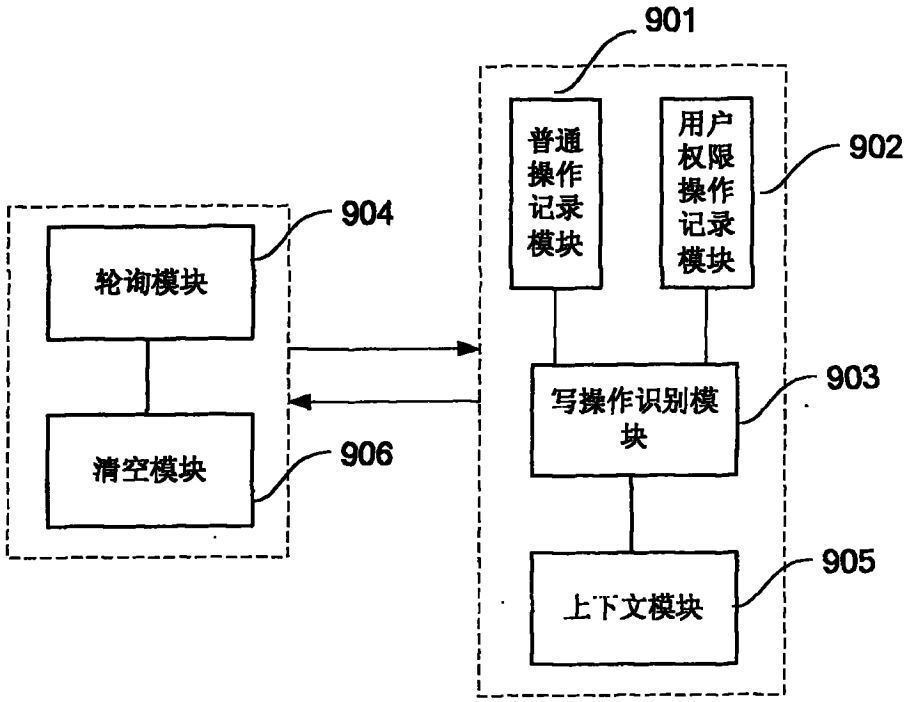

901
普通操作记录模块
用户权限操作记录模块
902
904
轮询模块
903
写操作识别模块
清空模块
906
905
上下文模块

图 9