



(12)发明专利

(10)授权公告号 CN 106372136 B

(45)授权公告日 2018.09.11

(21)申请号 201610741307.6

(22)申请日 2011.11.30

(65)同一申请的已公布的文献号

申请公布号 CN 106372136 A

(43)申请公布日 2017.02.01

(30)优先权数据

61/428,799 2010.12.30 US

13/227,381 2011.09.07 US

(62)分案原申请数据

201180068148.3 2011.11.30

(73)专利权人 脸谱公司

地址 美国加利福尼亚

(72)发明人 文卡特什瓦兰·文卡塔拉马尼

(74)专利代理机构 北京康信知识产权代理有限公司 11240

代理人 梁丽超 刘冀

(51)Int.Cl.

G06F 17/30(2006.01)

H04L 29/08(2006.01)

审查员 郝晓丽

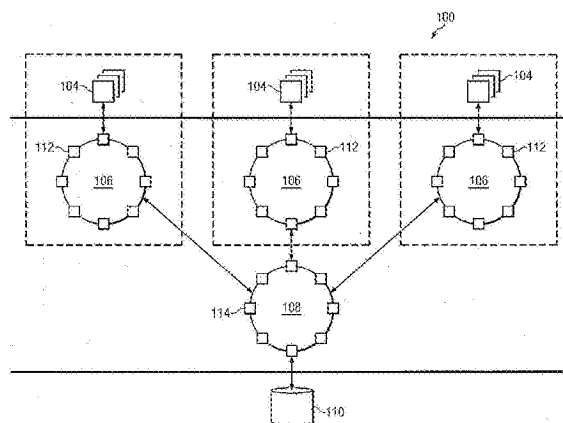
权利要求书3页 说明书16页 附图4页

(54)发明名称

分布式缓存系统、方法及存储介质

(57)摘要

本发明提供一种分布式缓存系统、方法及存储介质。所述系统包括：数据库；以及包括一个或多个缓存集群的缓存层。其中所述一个或多个缓存集群中的每一个操作为：在所述缓存集群的一个和多个缓存节点的内存中，维持社交图的至少一部分；处理从社交网络环境用户的一个或多个客户端系统接收的一个或多个查询；并且将响应于所接收的查询的搜索结果发送至所述一个或多个客户系统进行显示。



1. 一种分布式缓存系统,包括:数据库;以及包括多个缓存集群的缓存层,所述多个缓存集群中的每一个操作为:

在所述缓存集群的一个或多个缓存节点的内存中,维持社交图的一部分,所述社交图包括多个节点和连接所述节点的多个边,每个节点通过节点标识符唯一地识别,并且每个边指示节点之间的关联;

处理从社交网络环境的用户的一个或多个客户端系统接收的一个或多个查询,每个查询的所述处理包括:

如果所述查询是针对所述缓存集群中存储的所述社交图的所述一部分中的节点之间的关联,则通过搜索所述缓存集群的所述内存中存储的所述社交图的所述一部分来响应所述查询;以及

如果所述查询不是针对所述缓存集群中存储的所述社交图的所述一部分中的节点之间的关联,则将所述查询转发至所述数据库以进行处理;并且

将响应于所接收的查询的搜索结果发送至所述一个或多个客户端系统进行显示。

2. 根据权利要求1所述的系统,其中,所述一个或多个缓存节点进一步操作为:

接收改变所述内存中的关联信息的请求;

改变所述内存中的所述关联;

修改所述请求以便通过所述数据库实施;以及

将所修改的请求转发至所述数据库以进行处理。

3. 根据权利要求1所述的系统,其中,每个缓存集群包括多个缓存节点,每个缓存集群分配有一个或多个数据碎片,给定缓存集群的所述缓存节点的每一个存储关于与分配给所述给定缓存集群的数据碎片对应的节点的关联信息。

4. 根据权利要求3所述的系统,其中,所述多个缓存集群包括:

引导者缓存集群,包括多个引导者缓存节点;以及

一个或多个跟随者缓存集群,每个跟随者缓存集群包括多个跟随者缓存节点。

5. 根据权利要求1所述的系统,其中,所述一个或多个缓存节点进一步操作为:

对于与多个节点中的第一节点对应的每个关联集,在内存中维持第一索引和第二索引;其中,所述第一索引包括有序条目阵列,每个条目包括与所述第一节点相关联的第二节点的节点标识符以及分类属性;其中,所述第二索引包括哈希表,所述哈希表包括对应于与所述第一节点相关联的相应第二节点的所述节点标识符的条目;

接收在第一节点和第二节点之间添加多个关联类型中的第一关联类型的关联的命令,所述命令包括所述第一节点和所述第二节点的节点标识符;以及

就所述第一关联类型和所述第一节点的节点标识符访问所述内存,以向与所述第一关联类型和所述第一节点的节点标识符对应的第一索引和第二索引添加所述第二节点的节点标识符。

6. 根据权利要求5所述的系统,其中,所述一个或多个缓存节点进一步操作为:

针对每个关联集维持计数值;

响应于添加与相应关联集对应的关联的命令而递增计数值;以及

响应于删除与相应关联集对应的关联的命令而递减计数值。

7. 一种用于维持分布式缓存系统的方法,所述方法包括通过缓存层中的多个缓存集群

中的每一个执行的以下步骤：

在所述缓存集群的一个或多个缓存节点的内存中，维持社交图的一部分，所述社交图包括多个节点和连接所述节点的多个边，每个节点通过节点标识符唯一地识别，并且每个边指示节点之间的关联；

处理从社交网络环境的用户的一个或多个客户端系统接收的一个或多个查询，每个查询的所述处理包括：

如果所述查询是针对所述缓存集群中存储的所述社交图的所述一部分中的节点之间的关联，则通过搜索所述缓存集群的所述内存中存储的所述社交图的所述一部分来响应所述查询；以及

如果所述查询不是针对所述缓存集群中存储的所述社交图的所述一部分中的节点之间的关联，则将所述查询转发至数据库以进行处理；并且

将响应于所接收的查询的搜索结果发送至所述一个或多个客户端系统进行显示。

8. 根据权利要求7所述的方法，进一步包括通过每个缓存节点：

接收改变所述内存中的关联信息的请求；

改变所述内存中的所述关联；

修改所述请求以便通过所述数据库实施；以及

将所修改的请求转发至所述数据库以进行处理。

9. 根据权利要求7所述的方法，其中，每个缓存集群包括多个缓存节点，每个缓存集群分配有一个或多个数据碎片，给定缓存集群的所述缓存节点的每一个存储关于与分配给所述给定缓存集群的数据碎片对应的节点的关联信息。

10. 根据权利要求9所述的方法，其中，所述多个缓存集群包括：

引导者缓存集群，包括多个引导者缓存节点；以及

一个或多个跟随者缓存集群，每个跟随者缓存集群包括多个跟随者缓存节点。

11. 根据权利要求7所述的方法，进一步包括通过每个缓存节点：

对于与多个节点中的第一节点对应的每个关联集，在内存中维持第一索引和第二索引；其中，所述第一索引包括有序条目阵列，每个条目包括与所述第一节点相关联的第二节点的节点标识符以及分类属性；其中，所述第二索引包括哈希表，所述哈希表包括对应于与所述第一节点相关联的相应第二节点的所述节点标识符的条目；

接收在第一节点和第二节点之间添加第一关联类型的关联的命令，所述命令包括所述第一节点和所述第二节点的节点标识符；以及

就所述第一关联类型和所述第一节点的节点标识符访问所述内存，以向与所述第一关联类型和所述第一节点的节点标识符对应的第一索引和第二索引添加所述第二节点的节点标识符。

12. 根据权利要求11所述的方法，进一步包括通过每个缓存节点：

针对每个关联集维持计数值；

响应于添加与相应关联集对应的关联的命令而递增计数值；以及

响应于删除与相应关联集对应的关联的命令而递减计数值。

13. 一种存储用于维持分布式缓存系统的计算机可读指令的非暂时性存储介质，所述指令在被执行时操作用于使得一个或多个处理器操作为缓存层中的缓存集群，所述缓存层

包括多个缓存集群,所述多个缓存集群中的每一个操作为:

在所述缓存集群的一个或多个缓存节点的内存中,维持社交图的一部分,所述社交图包括多个节点和连接所述节点的多个边,每个节点通过节点标识符唯一地识别,并且每个边指示节点之间的关联;

处理从社交网络环境的用户的一个或多个客户端系统接收的一个或多个查询,每个查询的所述处理包括:

如果所述查询是针对所述缓存层中存储的所述社交图的所述一部分中的节点之间的关联,则通过搜索所述缓存集群的所述内存中存储的所述社交图的所述一部分来响应所述查询;以及

如果所述查询不是针对所述缓存集群中存储的所述社交图的所述一部分中的节点之间的关联,则将所述查询转发至数据库以进行处理;并且

将响应于所接收的查询的搜索结果发送至所述一个或多个客户端系统进行显示。

14. 根据权利要求13所述的存储介质,其中,所述一个或多个缓存节点进一步操作为:

接收改变所述内存中的关联信息的请求;

改变所述内存中的所述关联;

修改所述请求以便通过所述数据库实施;以及

将所修改的请求转发至所述数据库以进行处理。

15. 根据权利要求13所述的存储介质,其中,每个缓存集群包括多个缓存节点,每个缓存集群分配有一个或多个数据碎片,给定缓存集群的所述缓存节点的每一个存储关于与分配给所述给定缓存集群的数据碎片对应的节点的关联信息。

16. 根据权利要求13所述的存储介质,其中,所述多个缓存集群包括:

引导者缓存集群,包括多个引导者缓存节点;以及

一个或多个跟随者缓存集群,每个跟随者缓存集群包括多个跟随者缓存节点。

17. 根据权利要求13所述的存储介质,其中,所述一个或多个缓存节点进一步操作为:

对于与多个节点中的第一节点对应的每个关联集,在内存中维持第一索引和第二索引;其中,所述第一索引包括有序条目阵列,每个条目包括与所述第一节点相关联的第二节点的节点标识符以及分类属性;其中,所述第二索引包括哈希表,所述哈希表包括对应于与所述第一节点相关联的相应第二节点的所述节点标识符的条目;

接收在第一节点和第二节点之间添加第一关联类型的关联的命令,所述命令包括所述第一节点和所述第二节点的节点标识符;以及

就所述第一关联类型和所述第一节点的节点标识符访问所述内存,以向与所述第一关联类型和所述第一节点的节点标识符对应的第一索引和第二索引添加所述第二节点的节点标识符。

18. 根据权利要求13所述的存储介质,其中,所述一个或多个缓存节点进一步操作为:

针对每个关联集维持计数值;

响应于添加与相应关联集对应的关联的命令而递增计数值;以及

响应于删除与相应关联集对应的关联的命令而递减计数值。

分布式缓存系统、方法及存储介质

[0001] 本发明专利申请是中国专利申请CN201180068148.3 (发明名称为“用于图形数据的分布式缓存”)的分案申请。

技术领域

[0002] 本公开一般地涉及存储和供应图形数据,并且更特别地,涉及分布式缓存系统、方法及存储介质。

背景技术

[0003] 通过包括专有网络以及诸如因特网的公共网络的各种局域和广域计算机网络,计算机用户能够访问并共享大量信息。通常,用户的计算装置上安装的Web浏览器便于访问位于由关联的统一资源定位符(URL)识别的各种网络服务器的信息并与之互动。使得能够共享用户生成的内容的传统方法包括各种信息共享技术或诸如社交网络网站的平台。这样的网站可以包括、链接到、或提供使得用户能够查看由其他用户创建或定制的网页的应用程序的平台,其中,其他用户对这样的页面的可视性以及互动受规则的某些特性集支配。

[0004] 这样的社交网络信息,并且一般地大部分信息,通常存储在关系数据库中。一般地,关系数据库是关系(通常称为表)的集合。关系数据库使用一组数学术语,其可以使用结构化查询语言(SQL)数据库术语。例如,关系可以定义为具有相同属性的元组(tuple)集合。元组通常表示对象以及关于该对象的信息。关系通常被描述为组织成行和列的表。一般地,由属性引用的所有数据在同一域中并且遵循相同的约束。

[0005] 关系模型规定关系的元组没有具体顺序,并且元组也不反过来对属性施加顺序。应用程序通过规定查询来访问数据,该查询使用操作来识别元组、识别属性、以及组合关系。关系可以修改,并且新的元组可以提供明确的值或者从查询衍生。类似地,查询可以识别用于更新或删除的元组。关系的每个元组都需要通过其属性值的一些组合(一个或多个)可唯一地识别。该组合称为主键。在关系数据库中,经由关系来存储和访问所有数据。存储数据的关系通常用表实现或者称为表。

[0006] 诸如在关系数据库管理系统中实现的关系数据库已经成为用于例如财务记录、制造及物流信息、个人数据、以及其他应用程序的数据库中的信息存储的主要选择。随着计算机功率的增大,关系数据库对传统应用程序的易用性已经盖过了关系数据库效率低下,而效率低下使得关系数据库在较早时期不切实际。三个主导性的开源实施方式是MySQL、PostgreSQL、和SQLite。MySQL是关系数据库管理系统(RDBMS),其作为对多个数据库提供多用户接入的服务器运行。流行的LAMP软件堆栈缩写中的“M”是指MySQL。其用于网络应用程序的普及程度与PHP(LAMP中的“P”)的普及程度密切相关。几个高流量网站将MySQL用于数据存储和用户数据记录。

[0007] 由于与关系数据库的通信通常是速度瓶颈,因此许多网络利用缓存系统来供应特定信息查询。例如,Memcached(高性能缓存)是通用分布式内存缓存系统。其通常用于通过在RAM中缓存数据和对象来加速动态数据库驱动网站,以减少必须读取外部数据源(诸如数

数据库或API)的次数。Memcached的API提供了跨多个机器分布的巨型哈希表。当该表满时,后续的插入使得较早的数据以最近最少使用(LRU)的顺序被清除。在回落在较慢的备用存储区(诸如数据库)之前,使用Memcached的应用程序通常将请求和补充分层到内核中。

[0008] Memcached系统使用客服-服务器架构。服务器维持键-值关联阵列;客户端填充该阵列并查询该阵列。客户端使用客户端侧库与服务器联系。通常,每个客户端知道所有服务器,而服务器不彼此通信。如果服务器想要设置或读取与特定键对应的值,则客户端库首先计算该键的哈希,以确定将被使用的服务器。客户端然后与该服务器联系。服务器将计算该键的第二哈希,以确定在哪里存储或读取对应值。通常,服务器在RAM中保持该值;如果服务器耗尽了RAM,则其将最旧的值丢弃。因此,客户端必须将Memcached当作短暂缓存;它们不能假设在Memcached中存储的数据在它们需要时仍然在那里。

发明内容

[0009] 根据本发明的一个方面,提供一种系统,包括:数据库;以及包括一个或多个缓存集群的缓存层,其中所述一个或多个缓存集群中的每一个操作为:在所述缓存集群的一个和多个缓存节点的内存中,维持社交图的至少一部分,所述社交图包括多个节点和连接所述节点的多个边,每个节点通过节点标识符唯一地识别,并且每个边指示节点之间的关联;处理从社交网络环境用户的一个或多个客户端系统接收的一个或多个查询,每个查询的所述处理包括:如果所述查询是针对所述缓存集群中存储的所述社交图的所述部分中的节点之间的关联,则通过搜索所述缓存集群的所述内存中存储的所述社交图的所述部分来响应所述查询;以及如果所述查询不是针对所述缓存集群中存储的所述社交图的所述部分中的节点之间的关联,则将所述查询转发至所述数据库以进行处理;并且将响应于所接收的查询的搜索结果发送至所述一个或多个客户系统进行显示。

[0010] 根据本发明的又一个方面,提供一种方法,包括通过缓存层中的一个或多个缓存集群中的每一个执行:在所述缓存集群的一个和多个缓存节点的内存中,维持社交图的至少一部分,所述社交图包括多个节点和连接所述节点的多个边,每个节点通过节点标识符唯一地识别,并且每个边指示节点之间的关联;处理从社交网络环境用户的一个或多个客户端系统接收的一个或多个查询,每个查询的所述处理包括:如果所述查询是针对所述缓存集群中存储的所述社交图的所述部分中的节点之间的关联,则通过搜索所述缓存集群的所述内存中存储的所述社交图的所述部分来响应所述查询;以及如果所述查询不是针对所述缓存集群中存储的所述社交图的所述部分中的节点之间的关联,则将所述查询转发至数据库以进行处理;并且将响应于所接收的查询的搜索结果发送至所述一个或多个客户系统进行显示。

[0011] 根据本发明的又一个方面,提供一种存储计算机可读指令的非暂时性存储介质,所述指令在被执行时操作用于使得一个或多个处理器操作为缓存层中的缓存集群,所述缓存层包括一个或多个缓存集群,其中所述一个或多个缓存集群中的每一个操作为:在所述缓存集群的一个和多个缓存节点的内存中,维持社交图的至少一部分,所述社交图包括多个节点和连接所述节点的多个边,每个节点通过节点标识符唯一地识别,并且每个边指示节点之间的关联;处理从社交网络环境用户的一个或多个客户端系统接收的一个或多个查询,每个查询的所述处理包括:如果所述查询是针对所述缓存层中存储的所述社交图的所

述部分中的节点之间的关联,则通过搜索所述缓存集群的所述内存中存储的所述社交图的所述部分来响应所述查询;以及如果所述查询不是针对所述缓存集群中存储的所述社交图的所述部分中的节点之间的关联,则将所述查询转发至数据库以进行处理;并且将响应于所接收的查询的搜索结果发送至所述一个或多个客户系统进行显示。

附图说明

- [0012] 图1示出了根据本发明的一个实施方式的示例性缓存系统架构。
- [0013] 图2示出了示例性计算机系统架构。
- [0014] 图3提供了示例性网络环境。
- [0015] 图4示出了说明用于向图形添加新关联的示例性方法的流程图。
- [0016] 图5是说明缓存系统的各部件之间的示例性消息流的示意图。
- [0017] 图6示出了说明用于处理图形数据的变化示例性方法的流程图。
- [0018] 图7是说明缓存系统的各部件之间的示例性消息流的示意图。

具体实施方式

[0019] 特定实施方式涉及用于存储和提供建模为图形的信息的分布式缓存系统,该图形包括节点和边,边定义了该边在图形中连接的节点之间的关联或关系。在特定实施方式中,图形是或者包括社交图,并且分布式缓存系统是启用集成社交网络环境的较大网络系统、基础设置、或平台的一部分。在本公开中,可以就包括社交图信息的社交图来描述社交网络环境。实际上,本公开的特定实施方式依赖、采用、或利用由或为社交网络环境存储的大部分或所有数据可以表示为社交图这一事实。特定实施方式提供了成本有效的基础设施,就像本文中描述的那样,其能够有效地、智能地、以及成功地随着社交网络环境的用户数量的指数增加而扩展。

[0020] 在特定实施方式中,本文中描述的分布式缓存系统和后端基础设施提供了以下项中的一个或多个:扩展的低延迟、每个请求的更低成本、开发商容易使用的框架、支持多主装置(master)的基础设施、为客户端提供对以Hypertext Preprocessor (PHP) 以外的语言编写的存储数据的访问的基础设施、启用涉及本文中通过示例描述的社交图的关联(边)和对象(节点)的组合查询的基础设施、以及使得不同永久数据存储能够用于不同类型数据的基础设施。此外,特定实施方式提供了以下项中的一个或多个:使得能够从缓存+持久性+复制基础设施清楚地分离数据访问API的基础设施、支持写通/读通缓存的基础设施、移动使计算更接近数据的基础设施、使得能够对不同存储架构和后端透明移植的基础设施、以及提高数据对象访问的效率的基础设施。

[0021] 另外,如本文中所使用的,“或者”可以意味着“和”以及“或”,即,除非明确说明或暗指,否则“或者”并不必然排除“和”。

[0022] 特定实施方式可以在包括多个网络寻址系统的诸如因特网的广域网环境中操作。图3示出了可以操作各种示例性实施方式的示例性网络环境。网络云60一般表示本文中描述的系统 and 主机可以在上面通信的一个或多个互连网络。网络云60可以包括基于分组的广域网(诸如因特网)、私有网络、无线网络、卫星网络、蜂窝网络、寻呼网络等。如图3所示,特定实施方式可以在包括社交网络系统20以及一个或多个客户端装置30的网络环境中操作。

客户端装置30经由网络服务提供者、无线运营者、或其他合适的装置可操作地连接至网络环境。

[0023] 在一个示例性实施方式中,如本文中描述的,社交网络系统20包括计算系统,其允许用户彼此通信或另外地彼此互动并访问诸如用户档案的内容。社交网络系统20是可网络寻址系统,在各种示例性实施方式中,其包括一个或多个物理服务器22和数据存储24。一个或多个物理服务器22例如通过一组路由器和/或网络交换机26可操作地连接至计算机网络60。在示例性实施方式中,一个或多个物理服务器22承载的功能可以包括Web或HTTP服务器、FTP服务器、以及但不限于使用通用网关接口(CGI)脚本、PHP超文本预处理器(PHP)、有源服务器页面(ASP)、超文本标记语言(HTML)、可扩展标记语言(XML)、Java、JavaScript、Asynchronous JavaScript以及XML(AJAX)等实现的网页和应用程序。

[0024] 物理服务器22可以承载针对社交网络系统20的操作的功能。例如,社交网络系统20可以承载允许一个或多个用户在一个或多个客户端装置30处查看和发布信息以及经由网站彼此通信的网站。在下文中,虽然服务器22可以包括承载例如社交网络系统20的多个服务器以及其他内容分布服务器、数据存储器、和数据库,但是多个服务器22可以称为服务器22。数据存储24可以存储涉及并启用作为数字数据对象的社交网络系统的操作。在特定实施方式中,数据对象通常表现为存储或嵌入在数据文件、数据库、或记录中的数字信息。内容对象可以采取各种形式,包括:文本(例如,ASCII、SGML、HTML)、图像(例如,jpeg、tif、和gif)、图形(基于矢量的或位图)、音频、视频(例如,mpeg)、或其他多媒体、及其组合。内容对象数据还可以包括可执行代码对象(例如,可在浏览器窗口或帧中执行的游戏)、播客等。逻辑上,数据存储24对应于各种分立和集成的数据库中的一个或多个(诸如关系数据库和面向对象的数据库)它们将信息作为在一个或多个物理系统上存储的逻辑相关记录或文件的一体集合来维持。结构上,数据存储24一般地可以包括一大类的数据存储和管理系统中的一个或多个。在特定实施方式中,数据存储24可以通过任意适当物理系统来实现,该适当物理系统包括诸如一个或多个数据库服务器的部件、大容量介质、媒体库系统、存储区网络、数据存储云等。在一个示例性实施方式中,数据存储24包括一个或多个服务器、数据库(例如,MySQL)、和/或数据仓库。

[0025] 数据存储24可以包括与不同社交网络系统20的用户和/或客户端装置30相关联的数据。在特定实施方式中,社交网络系统20维持系统20的每个用户的用户档案。用户档案包括描述社交网络的用户的数据,例如,其可以包括:适当的名称(人的姓、中间名、和名、商业实体的商标名和/或公司名等)、履历、人口结构、和其他类型的描述性信息,诸如工作经历、教育背景、爱好或偏好、地理位置、以及额外的描述性数据。例如,用户档案可以包括用户的生日、关系状态、居住城市等。系统20可以进一步存储描述不同用户之间的一个或多个关系的数据。该关系信息可以指示具有类似或共同的工作经历、组成员身份、爱好、或教育背景的用户。用户档案还可以包括管理其他用户对该用户的信息的访问的隐私设置。

[0026] 客户端装置30一般是包括用于在计算机网络上进行通信(例如,远程地)的功能的计算机或计算装置。客户端装置30可以是台式计算机、膝上型计算机、平板电脑、个人数字助理(PDA)、车内或车外导航系统、蜂窝电话或其他智能或移动电话、或者移动游戏装置及其他适合的装置。客户端装置30可以执行一个或多个客户端应用程序,诸如Web浏览器(例如,Microsoft Windows Internet Explorer、Mozilla Firefox、Apple Safari、Google

Chrome、以及Opera等),以在计算机网络上访问和查看内容。在特定实施方式中,客户端应用程序允许客户端装置30的用户输入待检索的具体网络资源(诸如由社交网络系统20承载的资源)的地址。这些地址可以是唯一资源定位符或URL。另外,一旦已经检索了页面或其他资源,当用户“点击”到其他资源的超链接时,客户端应用程序可以提供到其他页面或记录的访问。例如,这样的超链接可以位于网页内部,并为用户提供输入另一页面的URL并检索该页面的自动化方法。

[0027] 图1示出了能够实现图3中示出的社交网络系统20的后端功能的网络系统、架构、或基础设施100(下文中,称为网络系统100)的示例性实施方式。在特定实施方式中,网络系统100使得网络系统100的用户能够经由网络系统100提供的社交网络服务彼此互动以及与第三方互动。例如,远程用户计算装置(例如,个人计算机、上网本、多媒体装置、蜂窝电话(尤其是智能电话)等)的用户可以经由Web浏览器或其他用户客户端应用程序访问网络系统100,以访问至少部分地由网络系统100承载或可访问的网站、网页、或网络应用程序,以查看信息、存储或更新信息、通信信息、或者与其他用户、第三方网站、网页、或网络应用程序、或者由网络系统100存储、承载、或可访问的其他信息互动。在特定实施方式中,如在下文中更详细地描述的,网络系统100维持包括表示用户、概念、主题、和其他信息(数据)的图形节点以及连接或限定图形节点之间的关系的图形边的图形。

[0028] 参照图1和图5,在特定实施方式中,网络系统100包括一个或多个数据中心102。例如,网络系统100可以包括战略性地位于各地理区域中用于为位于各区域中的用户服务的多个数据中心102。在特定实施方式中,每个数据中心都包括多个客户端或web服务器104(以下,称为客户端服务器104),其向/从网络系统100的用户进行信息通信。例如,经由网络和服务提供商的任意适当组合,在远程用户计算装置处的用户可以经由负载平衡器或其他适当系统与客户端服务器104通信。了检索数据以生成用于对用户请求进行响应的结构化文档,客户端服务器104可以查询本文中描述的缓存系统。

[0029] 每个客户端服务器104都与一个或多个跟随者分布式缓存集群或环106(以下称为跟随者缓存集群106)通信。在所示出的实施方式中,数据中心102包括各自服务于web服务器104的子集的三个跟随者缓存集群106。在特定实施方式中,跟随者缓存集群106以及跟随者缓存集群106所服务的客户端服务器104位置紧邻,诸如在建筑、房间、或其他集中的位置处,这降低了与基础设施(例如,有限或其他通信线路等)相关联的成本以及客户端服务器104和相应的服务跟随者缓存节点集群106之间的延迟。然而,在一些实施方式中,虽然跟随者缓存集群106中的每个及其分别服务的客户端服务器104可以位于集中位置,但是跟随者缓存集群106中的每个以及跟随者缓存集群106分别服务的相应客户端服务器104可以位于与给定数据中心的其他跟随者缓存集群106和相应客户端服务器104不同的位置;即,给定区域的给定数据中心的跟随者缓存集群106(以及该集群服务的相应客户端服务器104)可以遍布该区域内的不同位置分布。

[0030] 在特定实施方式中,每个数据中心102进一步包括引导者缓存集群108,其在给定数据中心102的跟随者缓存集群106和该给定数据中心102的永久存储数据库110之间通信信息。在特定实施方式中,数据库110是关系数据库。在特定实施方式中,引导者缓存集群108可以包括插件,其可操作为与数据库110的任何适当实现互操作。例如,数据库110可以实现为动态可变插件架构,并且可以使用MySQL和/或任何适当的关系数据库管理系统(诸

如,例如HAYSTACK、CASSANDRA及其他)。在一个实施方式中,插件执行各种转化操作,诸如将作为图形节点和边的缓存层中存储的数据转化成适于包括一个或多个表或平面文件的关系数据库的查询和命令。在特定实施方式中,引导者缓存集群108还协调来自跟随者缓存集群106的对数据库110的写请求,并且有时还协调来自跟随者缓存集群106的对于缓存在引导者缓存集群108或(如果没有缓存在引导者缓存集群108中的话)存储在数据库110中的信息的读请求。在特定实施方式中,引导者缓存集群108进一步协调相应数据中心102的跟随者缓存集群106中存储的信息的同步。即,在特定实施方式中,给定数据中心102的引导者缓存集群108被配置为维持数据中心102的跟随者缓存集群106之间的缓存一致性(例如,所缓存的信息),以维持跟随者缓存集群106和引导者缓存集群108之间的缓存一致性,以在数据库110中存储引导者缓存集群108中缓存的信息。在一个实施方式中,引导者缓存集群108和跟随者缓存集群106可以看作是客户端服务器104和数据库110之间的缓存层。

[0031] 在一个实施方式中,缓存层是所有读和写都遍历缓存层的写通/读通缓存层。在一个实施方式中,缓存层维持相关信息,并因此可以处理对这样的信息的查询。其他查询被传递到数据库110以用于执行。数据库110一般地意味着其本身可以包括用于处理其他查询类型的其他缓存层的数据库系统。

[0032] 每个跟随者缓存集群106可以包括均可以在单独的计算机、计算系统、或服务器上运行的多个跟随者缓存节点112。然而,如上所述,给定的跟随者缓存集群106的每个跟随者缓存节点112都可以位于中心位置。类似地,每个引导者缓存集群108可以包括均都可以在单独的计算机、计算系统、或服务器上运行的多个引导者缓存节点114。与给定跟随者缓存集群106的跟随者缓存节点112类似,给定引导者缓存集群108的每个引导者缓存节点114可以位于中心位置。例如,每个数据中心102可以包括数十个、数百个、或数千个客户端服务器104,而每个跟随者缓存集群可以包括数十个、数百个、或数千个服务于客户端服务器104的子集的跟随者缓存节点112。类似地,每个引导者缓存集群108可以包括数十个、上百个、或上千个引导者缓存节点114。在特定实施方式中,给定跟随者缓存集群106中的每个跟随者缓存节点112可以仅与特定跟随者缓存集群106中的其他跟随者缓存节点112、由特定跟随者缓存集群106服务的客户端服务器104、以及引导者缓存集群108中的引导者缓存节点114通信。

[0033] 在特定实施方式中,由网络系统100存储的信息分别存储在每个数据中心102中(两者都在数据库110中)以及跟随者缓存集群106和引导者缓存集群108中的每一个中。在特定实施方式中,每个数据库110中存储的信息被相关地存储(例如,经由MySQL存储为对象和表),而在由跟随者缓存集群106和引导者缓存集群108中的每个存储的多个数据碎片中,相同信息以包括图形节点以及节点之间的关联或关系(文中称为图形边)的图形的形式存储在跟随者缓存集群106和引导者缓存集群108中的每个中。在特定实施方式中,跟随者缓存集群106和引导者缓存集群108中的每个的数据碎片在相应的缓存集群的缓存节点112或114中被分桶(bucketized)或划分。即,相应缓存集群中的每个缓存节点112或114都存储由集群存储的碎片的子集(并且因为引导者缓存集群将由给定数据中心102(在一些实施方式中,数据中心102之间)的每个缓存集群存储的碎片同步,所以由跟随者缓存集群106和引导者缓存集群108中的每个分别存储的每个碎片集都存储相同的信息)。

[0034] 在特定实施方式中,每个图形节点都被分配唯一标识符(ID)(在下文中,称为节点

ID),其唯一地识别分别由跟随者缓存集群106和引导者缓存集群108中的每个以及数据库110存储的图形中的图形节点。在一个实施方式中,每个节点ID都是64比特标识符。在一个实施方式中,碎片分配了节点ID空间的片段。在特定实施方式中,每个节点ID都映射到(例如,算术地或经由某些数学函数)唯一对应碎片ID;即,每个碎片ID也是全局唯一的,并引用分别由跟随者缓存集群106和引导者缓存集群108中的每个存储的每个碎片集中的同一数据对象。换句话说,所有数据对象都存储为具有唯一节点ID的图形节点,并且分别存储在跟随者缓存集群106和引导者缓存集群108中的每个的数据碎片的图形中的所有信息使用相同的对应唯一碎片ID分别存储在跟随者缓存集群106和引导者缓存集群108中的每个的数据碎片中。

[0035] 如刚才描述的,在特定实施方式中,碎片ID空间(由每个缓存集群的所有碎片存储并在所有其他跟随者缓存集群106和引导者缓存集群108中被复制的碎片ID以及相关信息的集合)分别在跟随者缓存集群106和引导者缓存集群108中的跟随者缓存节点112或引导者缓存节点114中被划分。例如,给定跟随者缓存集群106中的每个跟随者缓存节点112可以存储由相应跟随者缓存集群106存储的碎片的子集(例如,数十个、上百个、或上千个碎片),并且每个碎片都被分配为其存储信息的节点ID范围,该信息包括关于相应节点ID映射到由特定碎片存储的碎片ID范围中的碎片ID的节点的信息。类似地,引导者缓存集群108中的每个引导者缓存节点114可以存储由相应引导者缓存集群108存储的碎片的子集(例如,数十个、上百个、或上千个碎片),并且每个碎片都被分配为其存储信息的节点ID范围,上述信息包括关于其对应节点ID映射到由特定碎片存储的碎片ID范围中的碎片ID的节点的信息。

[0036] 然而,如上所述,给定碎片ID分别对应于跟随者缓存集群106和引导者缓存集群108存储的同一数据对象。由于每个跟随者缓存集群106中的跟随者缓存节点的数量以及引导者缓存集群108中的引导者缓存节点114的数量可能会静态地(例如,跟随者缓存集群106和引导者缓存集群108一般地可以分别包括不同数量的跟随者缓存节点112和引导者缓存节点114)或动态地(例如,给定缓存集群中的缓存节点可以因为各种原因而周期性地或者在需要修复、更新、或维持时关闭)改变,因此在每个缓存集群中以及缓存集群之间,由跟随者缓存节点112和引导者缓存节点114中的每个存储的碎片的数量可以静态或动态地改变。此外,分配给每个碎片的碎片ID范围也可以静态或动态地改变。

[0037] 在特定实施方式中,跟随者缓存节点112和引导者缓存节点114中的每个都包括图形管理软件,其管理相应缓存节点中缓存的信息的存储和供应。在特定实施方式中,在给定缓存集群的每个缓存节点上运行的图形管理软件可以进行通信,以确定各缓存集群中的每个缓存节点都存储了哪些碎片(以及对应的碎片ID)。因此,如果缓存节点是跟随者缓存节点112,则在跟随者缓存节点112上运行的图形管理软件接收来自客户端服务器104的请求(读请求或写请求),通过在跟随者缓存节点的适当碎片中检索、更新、删除、或存储信息来服务请求,并且管理或促进在跟随者缓存节点112以及相应的跟随者缓存集群106的其他跟随者缓存节点112之间的通信以及在跟随者缓存节点112和引导者缓存集群108的引导者缓存节点114之间的通信。类似地,如果缓存节点是引导者缓存节点114,则在引导者缓存节点114上运行的图形管理软件管理引导者缓存节点114、和跟随者缓存集群106的跟随者缓存节点112、以及引导者缓存集群108的其他引导者缓存节点114之间的通信以及引导者缓存节点114和数据库110之间的通信。在缓存节点112和114中的每个上运行的图形管理软件明

白其以图形的形式管理和供应信息。

[0038] 在特定实施方式中,每个跟随者缓存节点112上的图形管理软件还负责维持其与相应的跟随者缓存集群106的其他缓存节点112、引导者缓存集群108的引导者缓存节点114、以及相应跟随者缓存集群106服务的客户端服务器104共享的表。该表提供了每个碎片ID到给定跟随者缓存集群106中的特定缓存节点112的映射,其中,上述给定缓存集群106存储碎片ID以及与该碎片ID相关联的信息。以这种方式,由特定跟随者缓存集群106服务的客户端服务器104知道跟随者缓存集群106中的哪个跟随者缓存节点112维持与客户端服务器104试图访问、添加、或更新的信息相关联的碎片ID(例如,客户端服务器104可以向特定跟随者缓存节点112发送写请求或读请求,该特定跟随者缓存节点112在使用映射表确定分配哪个跟随者缓存节点112之后存储或者将要存储与特定碎片ID相关联的信息并且存储碎片ID)。类似地,在特定实施方式中,每个引导者缓存节点114上的图形管理软件还负责维持其与对应引导者缓存集群108的其他缓存节点114以及该引导者缓存集群108管理的跟随者缓存集群106的跟随者缓存节点112共享的表。此外,以这种方式,给定跟随者缓存集群106中的每个跟随者缓存节点112都知道引导者缓存集群108中的哪个其他引导者缓存节点114存储了由引导者缓存集群108存储的哪个碎片ID。此外,给定跟随者缓存集群106中的每个跟随者缓存节点112知道引导者缓存集群108中的哪个引导者缓存节点114存储了哪个碎片ID。类似地,引导者缓存集群108中的每个引导者缓存节点114知道每个跟随者缓存集群106中的哪个跟随者缓存节点112存储了哪个碎片ID。

[0039] 在特定实施方式中,关于图形(在特定示例性实施方式中,社交图形)中的每个节点的信息基于其碎片ID存储在跟随者缓存集群106和引导者缓存集群108中的每个的相应碎片中。如上所述,图形中的每个节点都具有节点ID。连同碎片ID一起,相应缓存节点112或114还可以存储识别节点类型的节点类型参数以及一个或多个名称值对(诸如内容(例如,文本、媒体、或到媒体或其他资源的URL))和元数据(例如,创建或修改节点时的时间戳)。在特定实施方式中,图形(在特定示例性实施方式中,社交图形)中的每个边都连同该边连接到的每个节点被一起存储。例如,大多数边是双向的;即,大多数边中的每个都连接图形中的两个节点。在特定实施方式中,每个边都与该边连接的每个节点存储在同一碎片中。例如,将节点ID1连接至节点ID2的边可以与对应于节点ID1的碎片ID(例如,碎片ID1)和对应于节点ID2的碎片ID(例如,碎片ID2)一起存储,它们可以在不同碎片中,或者甚至在给定缓存集群的不同缓存节点中。例如,边可以以{node ID1, edge type, node ID2}的形式和碎片ID1一起存储,其中,边类型表示边的类型。边还可以包括元数据(例如,指示边何时被创建或修改的时间戳)。边还可以以{node ID1, edge type, node ID2}的形式与碎片ID2一起缓存。例如,当社交网络系统100的用户建立了与另一用户的联系关系或与概念或用户的爱好者关系时,类型为“朋友”或“爱好者”的边可以存储在与用户标识符映射到的碎片对应的第一碎片以及其他用户或概念的对象标识符映射到的第二碎片的两个碎片中。

[0040] 网络系统100,特别是在跟随者缓存集群106的跟随者缓存节点112和引导者缓存集群108的引导者缓存节点114上运行的图形管理软件,支持从客户端服务器104接收的多个查询以及分别去往或来自其他跟随者缓存节点112或引导者缓存节点114的查询。例如,query object_add{ID1,node type 1,metadata(并不总是指定),payload(并不总是指定)}使得接收缓存节点在节点ID1对应的碎片中存储具有在指定节点类型1的查询中指定

的节点ID1的新节点。如果指定了,则接收缓存节点还将元数据(例如,时间戳)和有效载荷(例如,名称值对和/或诸如文本、媒体、资源、或到资源的参考)与节点ID1一起存储。另一示例是,query object_update{ID1,node type 1(并不总是指定),metadata(并不总是指定),payload(并不总是指定)}使得接收缓存节点在对应碎片中更新由查询中指定的节点ID1识别的节点(例如,将节点类型改为查询中指定的节点类型1,用查询中指定的元数据更新元数据,或者更新与查询中指定的有效载荷一起存储的内容)。另一示例是,query object_delete{node ID1}使得接收缓存节点删除由查询中指定的节点ID1识别的节点。另一示例是,query object_get{node ID1}使得接收缓存节点检索与在查询中指定的节点ID1识别的节点一起存储的内容。

[0041] 现在参照边查询(与刚才描述的节点查询相对),query assoc_add{ID1,edge type 1,ID2,metadata(并不总是指定)}使得接收缓存节点(其存储节点ID1)在由节点ID1识别的节点和由节点ID2识别的节点之间创建边的类型为边类型1的边,并且如果指定了,则将具有由节点ID1识别的节点的边连同元数据(例如,指示何时请求该边的时间戳)一起存储。另一示例是,query assoc_update{node ID1,edge type 1,node ID2,metadata(并不总是指定)}使得接收缓存节点(其存储节点ID1)更新由节点ID1识别的节点和由节点ID2识别的节点之间的边。另一示例是,query assoc_delete{node ID1,edge type 1(并不总是指定),node ID2}使得接收缓存节点(其存储节点ID1)删除由节点ID1识别的节点和由节点ID2识别的节点之间的边。另一示例是,query assoc_get{node ID1,edge type 1、sortkey(并不总是指定)、start(并不总是指定)、limit(并不总是指定)}使得接收缓存节点(其存储节点ID1)通过节点类型1的边返回与由节点ID1指定的节点连接的节点的节点ID。另外,如果指定了,则分类键(sortkey)指定筛选器。例如,如果分类键指定了时间戳,则接收缓存节点(其存储节点ID1)返回通过边类型1的边连接至由节点ID1识别的节点的节点的节点ID,上述边在由开始参数指定的时间值和由限制参数指定的时间值之间创建。另一示例是,query assoc_exists{node ID1,edge type1,list of other node IDs,sortkey(并不总是指定),start(并不总是指定),limit(并不总是指定)}使得接收缓存节点(其存储节点ID1),返回通过边类型1的边连接至由碎片ID1识别的节点的其他节点ID列表中的指定节点的节点ID。另外,上述查询可以以所描述的形式发送,并用于更新引导者缓存节点114。

[0042] 在一个实施方式中,由跟随者缓存集群108和引导者缓存集群106实现的缓存层以支持一个或多个查询类型的高查询速率的方式在一个或多个索引中维持关联数据。在一些实施方式中,本发明有助于针对图形中的节点之间的关联的高效交集、会员资格和筛选查询。例如,在一个实施方式中,缓存层以被优化为对于处理节点之间的各种关联来处理点查找、范围和计数查询的方式来缓存信息。例如,在构造页面时,客户端服务器104可以发出对给定用户的所有朋友的查询。客户端服务器104可以发出识别了用户和“朋友”边类型的assoc_get查询。为了便于处理该查询,缓存层中的缓存节点可以存储第一节点(例如,与用户对应的节点)和与用户的联系人或朋友对应的节点之间的给定类型(诸如“朋友”、“爱好者”、“会员”、“爱好”等)的关联。另外,为了构造该页面的另一方,通过发出识别用户或用户档案、“帖子(wallpost)”边类型、和限制值的assoc_get查询,客户端服务器104可以发出对档案上的末尾N组帖子的查询。类似地,可以以类似方式检索对于特定贴子的评论。

[0043] 在一个实施方式中,由跟随者缓存集群106和引导者缓存集群实现的缓存层为图形中的节点(id1,id2)之间的关联维持便于快速搜索和处理高查询速率的一组内存中结构。例如,对于每个(id1,type)关联集(始于id1并具有给定类型的所有关联的集),缓存层维持两个内存中索引。如上所述,这些关联集由基于id1落入的碎片的每个集群中的缓存节点维持。进一步地,给出以下讨论的结构,可以在各自针对关联的相应节点的两个关联集中存储两个节点之间的给定关联。第一索引基于时间属性(例如,时间戳)并支持范围查询。id2的第二索引不支持范围查询,但是支持更好的插入和查找的时间复杂度。在一个实施方式中,第一索引是循环缓冲区中存储的关联条目的有序动态序列。循环缓冲区中的每个条目都描述或对应一个关联,并含有以下字段:a)\$flags(1字节)(指示关联的可见性);b)\$id2(8字节);c)\$time(4字节);d)\$data(8字节)(\$data是固定大小的8字节字段(当\$data需要8个以上字节时,这成为到保持全\$data值的另一内存块的指针;对于给定assoc类型,\$data是可选的);以及e)\$link(8字节),同一id2索引桶中的下一条目和前一条目的偏移(见下文)。在一个实施方式中,按照\$time属性升序对阵列分类。索引中的条目数量是有上限的(诸如10,000),并且可以通过关联类型来配置。当达到极限时,阵列回绕。由于阵列是\$time分类的,因此大部分新条目将附加在末尾,而不会移动任何现有元素。

[0044] 在一个实施方式中,主索引可以存储在可以通过全局高性能缓存哈希表用姓名(“assoc:<id1>:<type>”)查找的单个高性能缓存键中。该阵列的前面是含有以下字段的头:a)count(4字节):(id1,type)关联集(永久存储,不仅仅是索引中缓存的条目)中的可见关联的计数;b)head(4字节):循环缓冲区中的阵列头(分类最高的元素)的字节偏移;c)tail(4字节):循环缓冲区中的阵列尾(分类最低的元素)的字节偏移;以及d)id2索引指针(8字节):到含有id2哈希表的块的指针。

[0045] 在一个实施方式中,第二(\$id2)索引实现为哈希表,并支持给定(\$id1,\$type,\$id2)关联的快速插入和查找。在一个实施方式中,哈希表本身可以存储在与高性能缓存的内存分配器相关联的单独块中。表是到主索引中的偏移的阵列,其每个都识别对应哈希块中的第一元素。元素通过其\$link字段链接到块。在单独块中存储哈希表允许实施者单独地调整表和主索引的大小,从而减小随着关联集增大而复制的内存量。将关联条目就地链接到块中还提高了内存效率。当从索引擦除标记为隐藏或删除的条目时,可能需要重建哈希表(以及块列表),但是这很少进行。

[0046] 因此,随着添加了同一<type>的新关联,缓存节点112、114将最新关联的对象添加到哈希表和循环缓冲区,并且从循环缓冲区去除最旧的条目。如上所述,<sortkey>值可以用于基于诸如时间戳的属性将匹配条目分类。另外,<limit>值限制了向前N个值返回结构的数量,其中,N=<limit>。该配置允许以非常高的查询速率供应关于节点之间的关联的查询。例如,第一查询可能要求在网页的一部分中显示一组朋友。通过访问主索引并且检索循环缓冲区中的前N(其中,N=limit)个id2条目来查找对应于id1的关联集,缓存节点可以快速响应get_assoco(id1,type,sortkey,limit)。另外,二级索引的哈希表便于指针查找。进一步地,由缓存层维持的计数值便于对给定关联集(id1,type)的计数的快速响应。

[0047] 现在将描述分类和供应数据的一些一般示例(涉及社交图的特定示例实现方式的更具体的示例将在描述了社交图的特定示例实施方式之后描述)。例如,当客户端服务器104诸如从网络系统100的用户或者从网络系统100的另一服务器、部件、应用程序、或处理

接收网页的请求时(例如,响应于用户请求),客户端服务器104可能需要发出一个或多个查询,以生成所请求的网页。另外,随着用户与网络系统100互动,客户端服务器104可以接收建立或修改对象节点和/或对象节点的关联的请求。在一些情形中,客户端服务器104接收的请求一般包括表示代表其向客户端服务器104作出请求的用户的节点ID。另外,或者可选地,请求还可以包括与用户想要查看、更新、删除、或者链接或(与边)关联的对象对应的一个或多个其他节点ID。

[0048] 例如,请求可以是访问与用户想要查看的一个或多个对象(例如,用于服务网页的一个或多个对象)相关联的信息的读请求。例如,读请求可以是对于为特定节点存储的内容的请求。例如,用户档案上的帖子可以表示为具有“帖子”边类型的节点。对于该帖子的评论也可以表示为具有与帖子的“评论”边类型的关联的图形中的节点。在这样的示例中,在特定实施方式中,客户端服务器104确定与包括所请求的内容或其他信息的对象(节点)的节点ID对应的碎片ID,使用映射表来确定哪个跟随者缓存节点112(在服务客户端服务器104的跟随者缓存集群106中)存储碎片ID,并将包括碎片ID的查询发送到一个特定的跟随者缓存节点112,该特定的跟随者缓存节点112存储了与碎片ID相关并与碎片ID存储在一起的信息。该特定的缓存节点112然后检索所请求的信息(如果其缓存在对应碎片中)并将信息传输到请求客户端服务器104,客户端服务器104然后将该信息供应到请求用户(例如,以HTML或者web浏览器或用户的计算装置上运行的其他文档呈现应用程序可呈现的其他结构化文档的形式)。如果所请求的信息没有存储/缓存在跟随者缓存节点112中,则跟随者缓存节点112可以使用映射表确定哪个引导者缓存节点114存储了存储碎片ID的碎片,并将该查询转达到存储了该碎片ID的特定引导者缓存节点114。如果所请求的信息缓存在特定引导者缓存节点114中,则引导者缓存节点114可以随后检索所请求的信息,并将其转发到跟随者缓存节点112,跟随者缓存节点112然后更新跟随者缓存节点112中的特定碎片,从而所请求的信息和碎片ID一起存储,并且如刚才描述的那样将查询供应到客户端服务器104,客户端服务器104然后将信息供应到请求用户。如果所请求的信息没有缓存在引导者缓存节点114中,则引导者缓存节点114可以随后将查询转化成数据库110的语言,并将新的查询发送到数据库110,数据库110然后检索所请求的信息,并将所请求的信息发送到特定引导者缓存节点114。引导者缓存节点114然后可以将所检索的信息转化回图形管理软件理解的图形语言,在引导者缓存节点114中更新特定碎片,以将所请求的信息和碎片ID一起存储,并且将所检索的信息发送到特定跟随者缓存节点112,特定跟随者缓存节点112然后更新跟随者缓存节点112中的特定碎片,以将所请求的信息和碎片ID一起存储,并且如刚才描述的那样将查询供应到客户端服务器104,客户端服务器104然后将信息供应到请求用户。

[0049] 另一示例是,用户请求可以是更新节点的现有信息或存储额外信息或者创建或修改节点之间的边的写请求。在第一种情形中,如果待存储的信息是不存在的节点的,则接收用户请求的客户端服务器104将新节点的节点ID的请求发送到服务客户端服务器104的相应跟随者缓存集群106。在一些情形或实施方式中,客户端服务器104可以指定其中将存储新节点的特定碎片(例如,将新节点与另一节点共同定位)。在这样的情形中,客户端服务器104从存储了指定碎片的特定跟随者缓存节点112请求新节点ID。可选地,客户端服务器104可以将现有节点的节点ID连同对新节点ID的请求传递到存储碎片(该碎片存储所传递的节点ID)的跟随者缓存节点112,以使得跟随者缓存节点112以在碎片中存储的节点ID范围中

的新节点的节点ID对客户端服务器104进行响应。在其他情形或实施方式中,客户端服务器104可以(例如,随机地或基于同一函数)选择特定跟随者缓存节点112或特定碎片,以向其发送新节点ID请求。无论是哪种情形,特定缓存节点112,或者更特别地,在跟随者缓存节点112上运行的图形管理软件,将新节点ID发送到客户端服务器104。客户端服务器104然后可以将包括新节点ID的写请求公式化(formulate)到对应的跟随者缓存节点112。写请求还可以指定新节点的节点类型,并且包括将与节点ID一起存储的有效载荷(例如,将与新节点一起存储的内容)和/或元数据(例如,作出请求的用户的节点ID,指示客户端服务器104何时接收请求的时间戳,以及其他数据)。例如,发送到跟随者缓存节点112的写请求可以是以下形式:object_add{node ID,node type,payload,metadata}。类似地,为了更新节点,客户端服务器104可以向存储了其中存储节点ID的碎片的跟随者缓存节点112发送以下形式的写请求:object_modify{node ID,node type,payload,metadata}。类似地,为了删除节点,客户端服务器104可以向存储其中存储了节点ID的碎片的跟随者缓存节点112发送以下形式的写请求:object_delete{node ID}。

[0050] 在特定实施方式中,跟随者缓存节点然后将请求发送到存储了存储对应节点ID的碎片的引导者缓存节点114,使得引导者缓存节点114然后可以更新碎片。引导者缓存节点114然后将请求转化成数据库110的语言,并将转化的请求发送到数据库110,使得然后可以更新数据库。

[0051] 图4示出了用于处理添加两个节点之间的关联的请求(assoc_add)的示例性方法。如图4所示,当跟随者缓存节点112接收assoc_add请求时(例如,assoc_add(id1,type,id2,metadata)),其访问索引,以识别与id1和type对应的关联集对象(402)。跟随者缓存节点112将id2添加到关联集对象的哈希表和循环缓冲区,并且递增关联集对象的计数值(404)。关联集对象现在维持节点id1和节点id2之间的给定类型的新关联。为了便于搜索与id2相关的关联,跟随者缓存节点112识别与节点标识符id2对应的碎片ID,并且将assoc_add请求转发到处理所识别的碎片的集群中的跟随者缓存节点112(406)。如果瞬时跟随者缓存节点112处理碎片,则其处理assoc_add请求。在一个实施方式中,转发跟随者缓存节点112可以发送修改的assoc_add请求,其通知这是在缓存层中建立双向关联所需的更新。跟随者缓存节点112还将该assoc_add请求转发到与id1落入的碎片对应的引导者缓存节点114(408)。引导者缓存节点114可以执行类似处理,以在引导者缓存集群中建立双向关联。引导者缓存节点114还使得新关联持续存在于数据库110中。以这种方式,可以分开地参照id1和type以及id2和type在索引中搜索节点id1和节点id2之间的关联。

[0052] 在特定实施方式中,图形可以维持各种不同节点类型,诸如用户、页面、事件、贴子(wall posts)、评论、照片、视频、背景信息、概念、兴趣、以及将会对表示为节点有用的任何其他元素。边类型对应于节点之间的关联,并且可以包括朋友、追随者、用户、爱好者、喜好(或兴趣的其他表示)、贴子、评论、链接、建议、推荐、以及节点之间的其他关联类型。在一个实施方式中,图形的一部分可以是社交图,其包括每个都对应于社交网络环境的对应用户的用户节点。社交图还可以包括其他节点,诸如每个都专门用于或针对特定概念的概念节点以及主题节点,主题节点可能是或者可能不是短暂的,其每个都专门用于或针对社交网络环境用户的当前兴趣的特定主题。在特定实施方式中,每个节点都具有、表示在社交网络环境中承载或可访问的对应网页(“档案页面”),或者由该对应网页表示。例如,用户节点可

以具有对应用户档案页面,其中,对应用户可以添加内容,作出声明,或者表达他或她自己。例如,如将在下文中描述的,在社交网络环境中承载或可访问的各种网页(例如,用户档案页面、概念档案页面、或者主题档案页面)使得用户能够发布内容、发布状态更新、发布消息、发布包括对由该用户或其他用户提交的其他帖子的评论的评论、声明兴趣、声明对任意上述帖子以及页面和具体内容的“喜好”(在下文中描述)、或者表达自我或执行各种动作(下文中,这些和其他用户动作可以统称为“帖子”或“用户动作”)。在一些实施方式中,发布可以包括经由其对应档案页面、其他用户档案页面、概念档案页面、主题页面、或者其他网页或网络应用程序链接到或引用额外信息,诸如媒体内容(例如,相片、视频、音乐、文本等)、唯一资源定位符(URL)、以及其他节点。这样的帖子、声明、或动作对于认证用户和其他用户可见。在特定实施方式中,社交图进一步包括多个边,每个边都限定或表示社交图中的一对对应节点之间的连接。如上所述,内容的每个项目(item)都可以是链接到其他节点的图形中的节点。

[0053] 如上所述,在各示例性实施方式中,所描述的一个或多个网页或网络应用程序与社交网络环境或社交网络服务相关联。如本文中所使用的,“用户”可以是使用这样的社交网络环境或者在这样的社交网络环境上互动或通信的个人(真人用户)、实体(例如,企业、商业、或第三方应用程序)、或组(例如,个人或实体的组)。如文本中所使用的,“注册用户”是指在社交网络环境中正式注册的用户(一般地,本文中描述的用户和用户节点仅指注册用户,虽然这在其他实施方式中并不一定要求;即,在其他实施方式中,本文中描述的用户和用户节点可以指还没有在本文中描述的社交网络环境注册的用户)。在特定实施方式中,每个用户都具有对应的“档案”页面,该页面由社交网络环境存储、承载、或者可由社交网络环境访问,并且对所有或选择的其他用户的子集可见。一般地,用户对他或她自己的相应档案页面的全部或部分具有管理权,并且潜在地对由或为该特定用户创建的其他页面(例如,主页、承载网络应用程序的页面)具有管理权。如本文中所使用的,“认证用户”是指已经被社交网络环境认证为在该用户具有管理权的对应档案页面中被宣称的用户、或者所宣称的用户的适当可信代表的用户。

[0054] 两个用户或概念之间的连接可以表示社交网络环境的用户或概念之间的限定关系,并且可以在社交网络环境的适当数据结构中逻辑地定义为对应于用户、概念、事件的节点、或已经作出关联的社交网络环境的其他节点之间的边。如本文中所使用的,“友谊”表示社交网络环境的一对用户之间的关联,诸如定义的社交关系。如本文中所使用的,“朋友”可以指以用户已经与其形成连接、友谊、关联、或关系使得将在两个用户之间生成边的社交网络环境的任意用户。例如,通过两个用户之一选择与另一个的友谊(作为向可以接受或拒绝请求的另一用户传输友谊请求或使得友谊请求被传输的结果),两个注册用户可以彼此明确地成为朋友。可选地,可以自动建立友谊关系或其他连接。这样的社交友谊关系对其他用户可见,特别是对于其本身与注册用户中的一个或两个是朋友的用户可见。注册用户的朋友对注册用户的档案或其他页面上的内容,特别是用户生成或声明的内容,具有增大的访问权限。然而,应当指出的是,在社交图中建立了二者之间的朋友连接的两个用户在现实生活(社交网络环境外部)中不一定是朋友(传统意义上)。例如,在一些实现中,用户可以是商业或其他非真人实体,因此,在该词的传统意义上,其不能与真人用户成为朋友。

[0055] 如本文中所使用的,“爱好者”可以指是特定用户、网页、网络应用程序、或可在社

交网络环境中访问的其他Web内容的支持者或追随着的用户。在特定实施方式中,当用户是特定网页的爱好者时(“爱好”特定网页),该用户可以作为爱好者列在该页面上,以供其他注册用户或一般公众查看。另外,该用户的头像或档案图片可以显示在页面上(或者在以下描述的任意页面中/上)。如本文中所使用的,“喜欢”可以指某物,诸如,例如但不限于,除了其他可能性,用户(特别是注册或认证用户)已经声明或表明他或她喜欢、是其爱好者、支持、欣赏、或另外有正面看法的帖子、评论、兴趣、链接、媒体(例如,照片、相册、视频、歌曲等)、概念、实体、页面及其他可能(在一些实现中,用户可以表示或声明对社交网络系统或环境承载或可访问的任意页面上的几乎任何事物的喜欢)。在一个实施方式中,表示或声明“喜欢”或者表示或声明该用户是某物的“爱好者”在社交网络环境中可以被等效地处理和定义,并且可以互换地使用;类似地,声明自己是诸如概念或概念档案页面的某物的“爱好者”,或者声明自己“喜欢”该事物,可以在社交网络环境中等效地定义并且可以在本文中互换地使用。另外,如本文中所使用的,“兴趣”可以指用户声明的兴趣,诸如在用户档案页面中呈现的用户声明的兴趣。如本文中所使用的,“想要”可以指用户想要的几乎任何事物。如上所述,“概念”可以指用户可以声明或表明兴趣、喜欢、或关系的几乎任何事物,例如,体育、体育组、音乐流派、音乐作曲家、爱要、商业(企业)、实体、组、名人、不是注册用户的个人、或者甚至是事件、在一些实施方式中,另一用户(例如,非认证用户)等。例如,可以有由多个用户中的一个或多个(例如,Jerry Rice以外的用户)创建和管理的著名的专业足球运动员“Jerry Rice”的概念节点和概念档案页面,而社交图形额外包括由Jerry Rice自己(或者Jerry Rice的信任或授权代表)创建和管理的Jerry Rice的用户节点和用户档案页面。

[0056] 图5示出了分布式冗余系统。在所示出的实施方式中,分布式冗余系统至少包括第一数据中心102a和第二数据中心102b。数据中心102a和102b中的每个都包括一个或多个跟随者缓存集群106和引导者缓存集群108a、108b。在一个实施方式中,引导者缓存集群108a充当主要(主)缓存集群,而引导者缓存集群108b是次要(从)缓存集群。在一个实施方式中,数据中心102a、102b在用于实现数据库110的复制性副本的同步功能这个意义上是冗余的。在一个实施方式中,数据中心102a可以物理地位于一个地理区域(诸如美国西海岸)以服务来自该区域的流量,而数据中心102b可以物理地位于另一地理区域(诸如美国东海岸)。已知来自这些区域中的任一个的用户都可以访问相同的数据和关联,因此需要有效的同步机制。

[0057] 图6示出了引导者缓存节点114如何处理写命令的示例性方法。如上所述,并且参照图5,跟随者缓存节点112可以从客户端服务器104接收添加/更新对象或关联的写命令(图5,标号1)。跟随者缓存节点112将写命令转发到对应引导者缓存节点114(图5,标号2)。当引导者缓存节点114接收到来自跟随者缓存节点的写命令时(602),其处理写命令,以更新由引导者缓存集群108a维持的缓存中的一个或多个条目(604),并将更新写入永久数据库110a(606)(图5,标号3)。引导者缓存节点114还应答到跟随者缓存节点112的写命令(ACK),并且将向数据中心102a的其他跟随者缓存集群106(图5,标号4a)和次要引导者缓存集群108b广播更新,次要引导者缓存集群108b将更新转发到其跟随者缓存集群106(图5,标号4b)(608)。如图6所示,引导者缓存节点114还将更新添加到复制日志(610)。数据库110a、110b实现诸如MySQL复制的同步机制,以同步永久数据库。

[0058] 图7示出了根据本发明的一个实施方式的消息流。当在不与主引导者缓存集群108a直接相关联的环106中的跟随者缓存节点112接收到写命令时(图7,标号1),跟随者缓存节点112将写消息转发到主引导者缓存集群108a以用于处理(图7,标号2)。主引导者缓存集群108a中的引导者缓存节点114然后可以向其跟随者缓存集群106广播该更新(图7,标号3)并且将变化写入数据库110a。如图7所示,接收写命令的跟随者缓存节点112还可以将写命令转发到其次要引导者缓存集群108b(图7,标号5),其向其跟随者缓存集群106广播该更新(图7,标号5)。上述架构允许跨数据中心快速地复制缓存层的变化,而数据库110a、110b之间的分开复制允许数据安全性。

[0059] 本文中描述的应用程序或处理可以实施为嵌入或编码在有形数据存储介质上/中的一系列计算机可读指令,当该指令被执行时,操作为使得一个或多个处理器实施上述操作。虽然已经通过广泛类别的物理系统在广泛类别的网络和计算环境中实施了上述处理和机制,但是以下描述的计算系统提供了上述服务器和客户端系统的示例性计算系统架构,用于教学而非限制性目的。

[0060] 图2示出了可以用于实现服务器22a、22b的示例性计算系统架构。在一个实施方式中,硬件系统1000包括处理器1002、缓存内存1004、以及在有形计算机可读介质上存储的针对本文中描述的功能的一个或多个可执行模块和驱动。另外,硬件系统1000包括高性能输入/输出(I/O)总线1006和标准I/O总线1008。主桥1010将处理器1002耦接至高性能I/O总线1006,而I/O总线桥1012将两个总线1006和1008彼此耦接。系统内存1014以及一个或多个网络/通信接口1016耦接至总线1006。硬件系统1000可以进一步包括视频内存(未示出)以及耦接至视频内存的显示装置。大容量存储器1018以及I/O端口1020耦接至总线1008。硬件系统1000可以可选地包括键盘和指针装置、以及耦接至总线1008的显示装置(未示出)。这些元件旨在共同地表示广泛类别的计算机硬件系统,包括但不限于基于由Intel Corporation of Santa Clara, California制造的x86-兼容处理器以及由Advanced Micro Devices (AMD), Inc., of Sunnyvale, California制造的x86-兼容处理器以及任何其他适当处理器的通用计算机系统。

[0061] 以下更详细地描述了硬件系统1000的元件。特别地,网络接口1016提供了硬件系统1000和诸如以太网(例如,IEEE 802.3)网络、背板等的任何各种各样的网络之间的通信。大容量存储器1018为数据和编程指令提供了永久存储,以执行在服务器22a、22b中实现的上述功能,而当由处理器1002执行时,系统内存1014(例如,DRAM)为数据和编程指令提供了临时存储。I/O端口620是在可以耦接至硬件系统1000的额外外围装置之间提供通信的一个或多个串行和/或并行通信端口。

[0062] 硬件系统1000可以包括各种系统架构;并且可以重新安排硬件系统1000的各部件。例如,缓存1004可以与处理器1002集成。可选地,缓存1004和处理器1002可以封装在一起作为“处理器模块”,处理器1002称为“处理器核”。此外,本发明的特定实施方式可能不要求包括所有上述部件。例如,示出为耦接至标准I/O总线1008的外围装置可以耦接至高性能I/O总线1006。另外,在一些实施方式中,可以只存在单个总线,硬件系统1000的部件耦接至该单个总线。此外,硬件系统1000可以包括额外的部件,诸如额外的处理器、存储装置、或内存。

[0063] 在一个实施方式中,本文中所描述的实施方式的操作实施为由硬件系统1000单独

或集中在分布式计算环境中运行的一系列可执行模块。在特定实施方式中，一组软件模块和/或驱动实施网络通信协议栈、浏览和其他计算功能、优化处理等。上述功能模块可以由硬件、计算机可读介质上存储的可执行模块、或者二者的组合来实现。例如，功能模块可以包括将要由硬件系统中的处理器（诸如处理器1002）执行的多个或一系列指令。最初，该一系列指令可以存储在诸如大容量存储器1018的存储装置上。然而，该一系列指令可以切实存储在任何适当的存储介质上，诸如软盘、CD-ROM、ROM、EEPROM等。此外，该一系列指令不需要本地存储，并且可以经由网络/通信接口1016从诸如网络上的服务器的远程存储装置接收。该指令从诸如大容量存储器1018的存储装置复制到内存1014中，并且然后由处理器1002访问并执行。

[0064] 操作系统管理并控制硬件系统1000的操作，包括到/从软件应用程序（未示出）的数据输入和输出。操作系统在系统上执行的软件应用程序和系统的硬件部件之间提供了接口。可以使用任何适当的操作系统，诸如Linux操作系统、可从Apple Computer Inc. of Cupertino, Calif. 获得的Apple Macintosh操作系统、UNIX操作系统、Microsoft (r) Windows (r) 操作系统、BSD操作系统等。当然，其他实施方式也是可能的。例如，可以在固件中或者专用集成电路上实施本文中描述的昵称生成功能。

[0065] 此外，上述元件和操作可以包括存储介质上存储的指令。该指令可以由处理系统检索并执行。指令的一些示例有软件、程序代码、以及固件。存储介质的一些示例有存储装置、磁带、磁盘、集成电路、以及服务器。当由处理系统执行时，指令操作为指示处理系统根据本发明操作。术语“处理系统”是指单个处理装置或者一组内部操作的处理装置。处理装置的一些示例有集成电路和逻辑电路。本领域普通技术人员熟悉指令、计算机、以及存储介质。

[0066] 本领域普通技术人员将会理解，本公开包括对本文的示例性实施方式的所有改变、替换、变形、变更和修改。类似地，本领域普通技术人员将会理解，适当时，所附权利要求包括对本文的示例性实施方式的所有改变、替换、变形、变更和修改。例如，虽然已经结合社交网站描述了本发明的实施方式的操作，但是本发明可以用于任何支持网络应用程序并将数据建模为关联图的通信设施。此外，在一些实施方式中，术语“网络服务”和“网站”可以互换地使用，并且另外可以指在使API直接调用服务器的诸如移动装置（例如，蜂窝电话、智能电话、个人GPS、个人数字助理、个人游戏装置等）的装置上定制或生成的API。

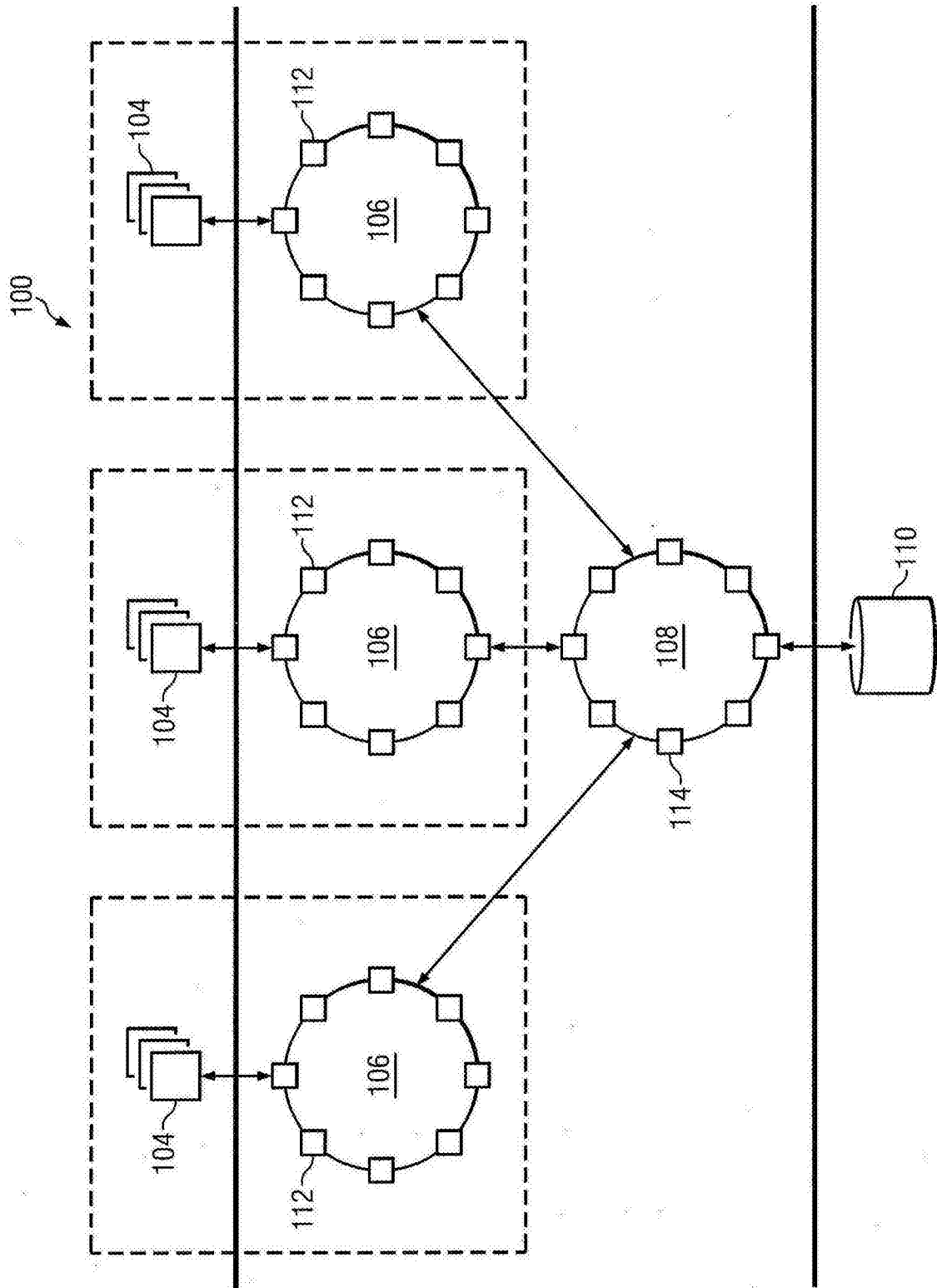


图1

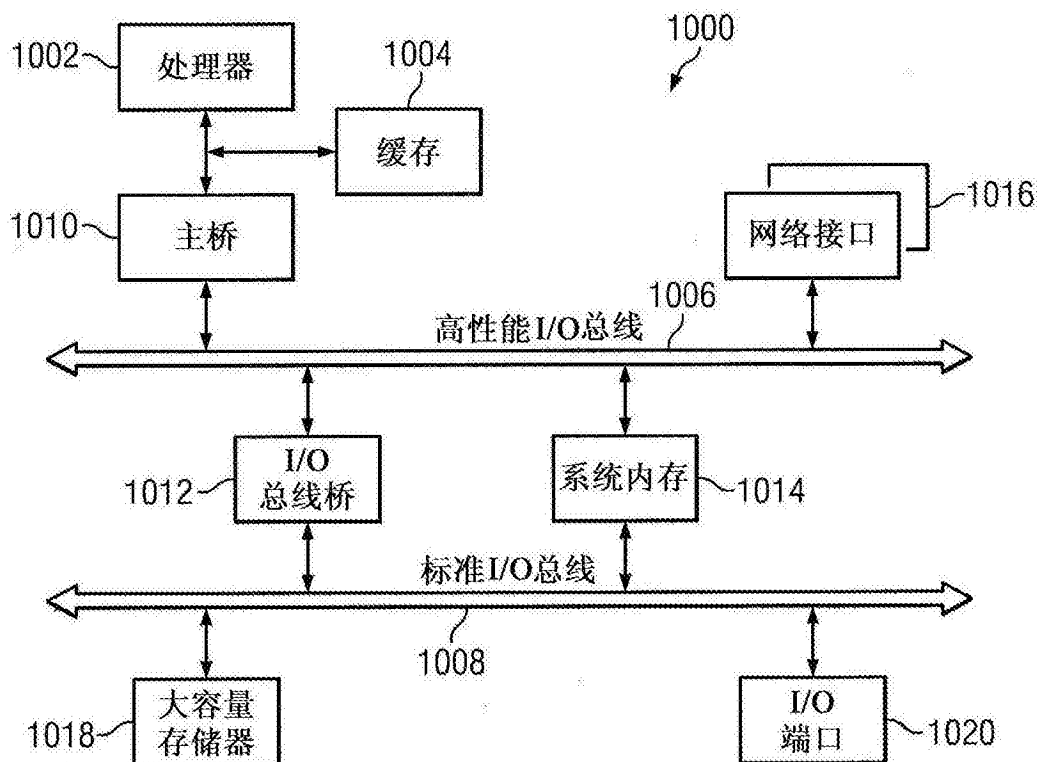


图2

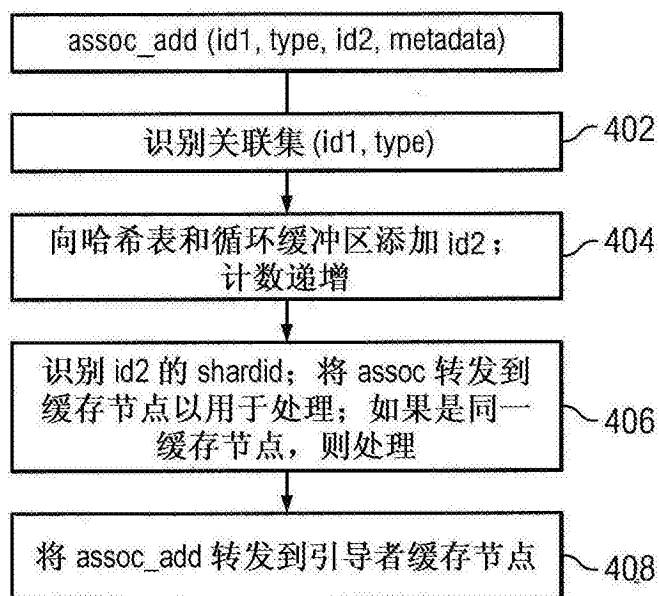


图4

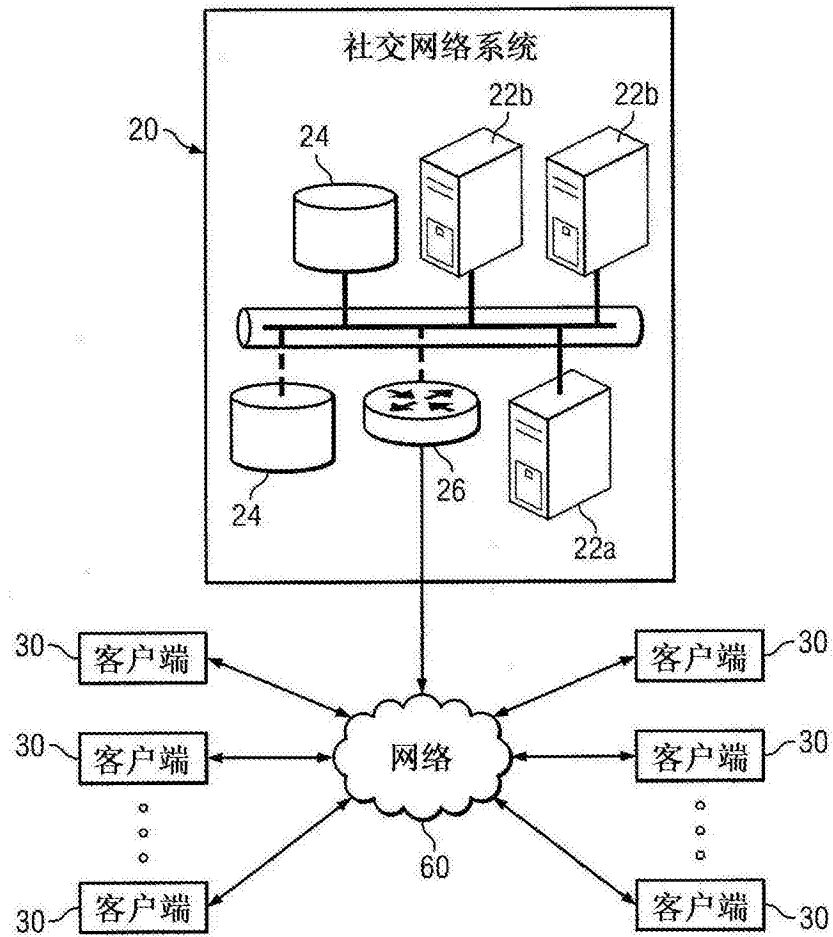


图3

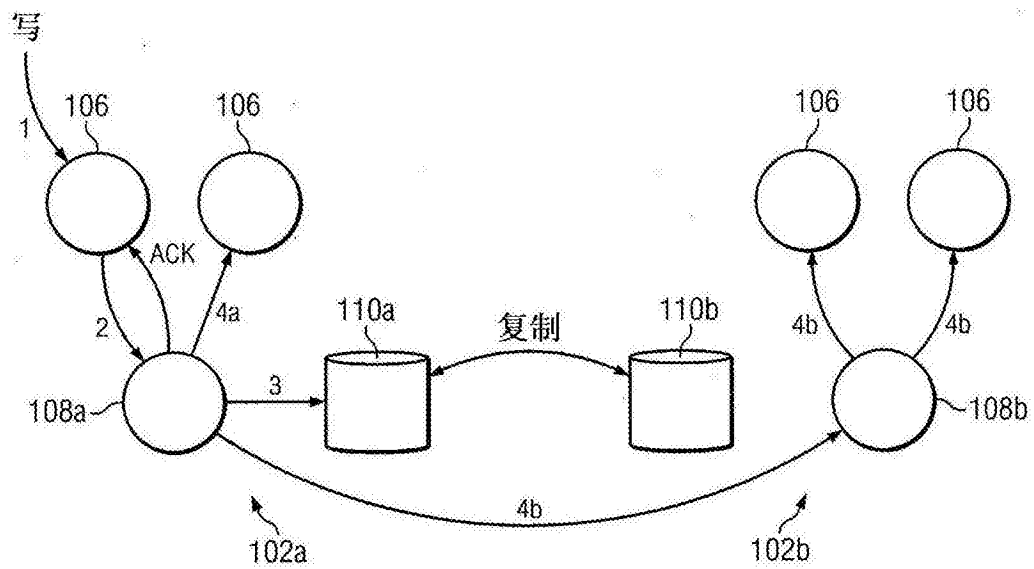


图5

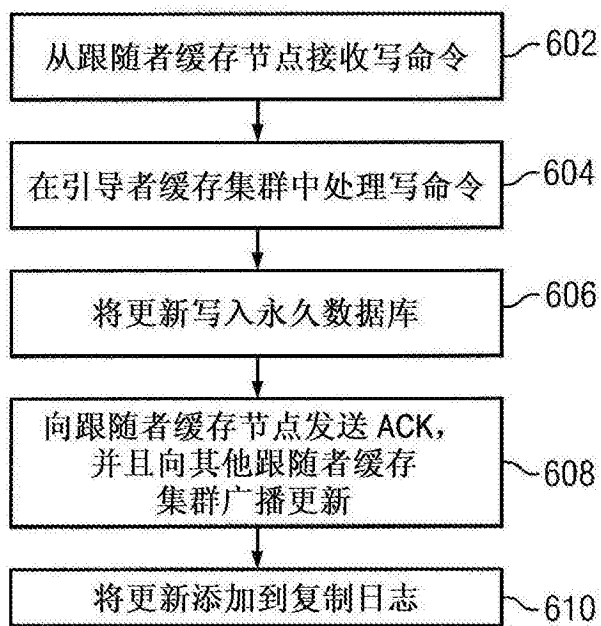


图6

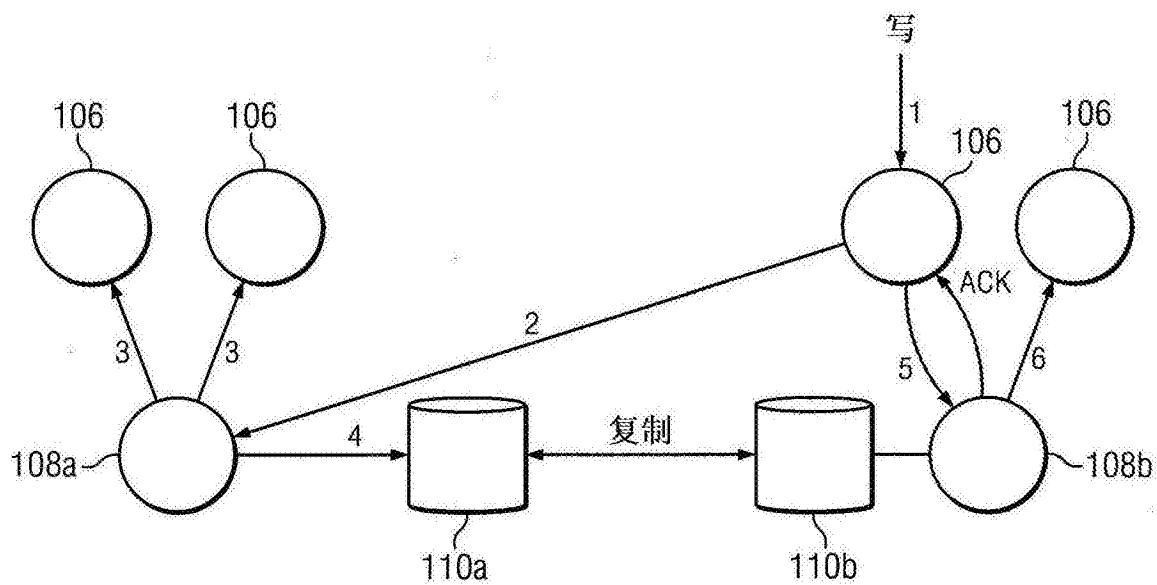


图7