



(12)发明专利申请

(10)申请公布号 CN 111666302 A
(43)申请公布日 2020.09.15

(21)申请号 202010469645.5

(22)申请日 2020.05.28

(71)申请人 广州虎牙科技有限公司
地址 511400 广东省广州市番禺区钟村街
(汉溪商业中心)泽溪街13号1301

(72)发明人 黄玉书

(74)专利代理机构 北京品源专利代理有限公司
11332
代理人 孟金喆

(51)Int.Cl.
G06F 16/2455(2019.01)

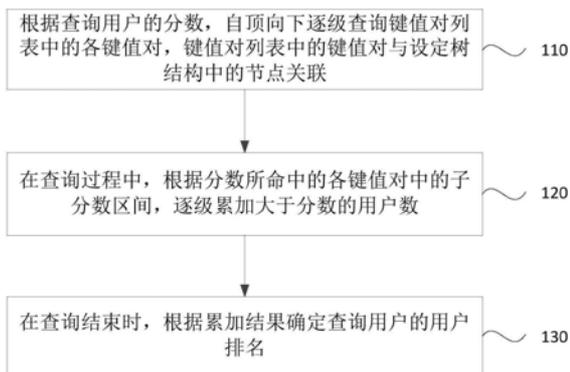
权利要求书3页 说明书14页 附图3页

(54)发明名称

用户排名的查询方法、装置、设备及存储介质

(57)摘要

本发明实施例公开了一种用户排名的查询方法、装置、设备及存储介质。该方法包括：根据查询用户的分数，自顶向下逐级查询键值对列表中的各键值对，键值对列表中的键值对与设定树结构中的节点关联；其中，键值对的键名为指定分数区间，键值为指定分数区间下的各子分数区间分别包含的用户数；同一父节点下全部子节点对应的键值对中的指定分数区间的总和为所述父节点对应的键值对中的指定分数区间；在查询过程中，根据分数所命中的各键值对中的子分数区间，逐级累加大于分数的用户数；在查询结束时，根据累加结果确定查询用户的用户排名。本发明实施例的技术方案，在大规模数据量下可以实时获取用户的具体排名，并节省数据存储空间。



1. 一种用户排名的查询方法,其特征在于,包括:

根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,所述键值对列表中的键值对与设定树结构中的节点关联;

其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为所述父节点对应的键值对中的指定分数区间;

在查询过程中,根据所述分数所命中的各键值对中的子分数区间,逐级累加大于所述分数的用户数;

在查询结束时,根据累加结果确定所述查询用户的用户排名。

2. 根据权利要求1所述的方法,其特征在于,在查询过程中,根据所述分数所命中的各键值对中的子分数区间,逐级累加大于所述分数的用户数,包括:

在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

确定所述分数在目标键值对的键值中命中的第一子分数区间,并确定大于第一子分数区间的全部第二子分数区间;

将与各第二子分数区间分别对应的用户数与设定累加值进行累加,累加值具有设定初始值;

在所述键值对列表中,获取键名与所述第一子分数区间匹配的键值对作为新的目标键值对后,返回执行获取所述分数在目标键值对的键值中命中的第一子分数区间的操作,直至处理至与叶子节点对应的键值对。

3. 根据权利要求1所述的方法,其特征在于,在根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对之前,还包括:

获取查询用户的目标用户标识;

根据用户标识与分数之间的映射关系,获取与所述目标用户标识对应的分数作为所述查询用户的分数。

4. 根据权利要求1所述的方法,其特征在于,还包括:

响应于与键值对列表匹配的分数调整请求,获取与所述分数调整请求匹配的至少一个待调整分数;

根据所述至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整。

5. 根据权利要求4所述的方法,其特征在于,所述分数调整请求包括新增分数请求,与所述新增分数请求匹配的待调整分数为新增分数;

根据所述至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,包括:

在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

确定所述新增分数在目标键值对的键值中命中的第一目标子分数区间,并将与所述第一目标子分数区间对应的用户数加一;

在所述键值对列表中,获取键名与所述第一目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定所述新增分数在目标键值对的键值中命中的第一目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

6. 根据权利要求4所述的方法,其特征在于,所述分数调整请求包括删除分数请求,与所述删除分数请求匹配的待调整分数为删除分数;

根据所述至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,包括:

在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

确定所述删除分数在目标键值对的键值中命中的第二目标子分数区间,并将与所述第二目标子分数区间对应的用户数减一;

在所述键值对列表中,获取键名与所述第二目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定所述删除分数在目标键值对的键值中命中的第二目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

7. 根据权利要求4所述的方法,其特征在于,所述分数调整请求包括调整分数请求,与所述调整分数请求匹配的待调整分数为调整前分数和调整后分数;

根据所述至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,包括:

在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

确定所述调整前分数在目标键值对的键值中命中的第三目标子分数区间,并将与所述第三目标子分数区间对应的用户数减一;

在所述键值对列表中,获取键名与所述第三目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定所述调整前分数在目标键值对的键值中命中的第三目标子分数区间的操作,直至处理至与叶子节点对应的键值对;

在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

确定所述调整后分数在目标键值对的键值中命中的第四目标子分数区间,并将与所述第四目标子分数区间对应的用户数加一;

在所述键值对列表中,获取键名与所述第四目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定所述调整后分数在目标键值对的键值中命中的第四目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

8. 根据权利要求1-7任一项所述的方法,其特征在于,所述树结构为16阶的平衡多叉树;

其中,按照键值对中的指定分数区间从小到大的顺序以及树结构从左子节点到右子节点的顺序,将各所述键值对与对应的树节点关联。

9. 根据权利要求1-7任一项所述的方法,其特征在于,所述键值对列表存储于RocksDB中。

10. 一种用户排名的查询装置,其特征在于,包括:

查询模块,用于根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,所述键值对列表中的键值对与设定树结构中的节点关联;

其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为所述父节点对应的键值对中的指定分数区间;

累加模块,用于在查询过程中,根据所述分数所命中的各键值对中的子分数区间,逐级

累加大于所述分数的用户数；

排名确定模块，用于在查询结束时，根据累加结果确定所述查询用户的用户排名。

11. 一种电子设备，其特征在于，所述电子设备包括：

一个或多个处理器；

存储装置，用于存储一个或多个程序，

当所述一个或多个程序被所述一个或多个处理器执行，使得所述一个或多个处理器实现如权利要求1-9中任一所述的用户排名的查询方法。

12. 一种计算机可读存储介质，其上存储有计算机程序，其特征在于，该程序被处理器执行时实现如权利要求1-9中任一所述的用户排名的查询方法。

用户排名的查询方法、装置、设备及存储介质

技术领域

[0001] 本发明实施例涉及计算机技术领域,尤其涉及一种用户排名的查询方法、装置、设备及存储介质。

背景技术

[0002] 目前,显示排行榜是各种应用程序的常见需求,一般除了展示排行榜中TOPN的用户榜单之外,还需要显示当前用户的具体排名。

[0003] 现有技术中,MySQL等常见的数据库或存储引擎通常使用内存排序,在大规模数据量下并不能很好很快地获取到用户排名。并且,如果榜单应用太多超出内存限制,排名获取速度会明显下降,有数据落盘需要时,系统重启也会成为负担。

发明内容

[0004] 本发明实施例提供一种用户排名的查询方法、装置、设备及存储介质,实现了在大规模数据量下可以实时获取用户的排名,并节省数据存储空间。

[0005] 第一方面,本发明实施例提供了一种用户排名的查询方法,包括:

[0006] 根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;

[0007] 其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;

[0008] 在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;

[0009] 在查询结束时,根据累加结果确定查询用户的用户排名。

[0010] 可选的,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数,包括:

[0011] 在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

[0012] 确定分数在目标键值对的键值中命中的第一子分数区间,并确定大于第一子分数区间的全部第二子分数区间;

[0013] 将与各第二子分数区间分别对应的用户数与设定累加值进行累加,累加值具有设定初始值;

[0014] 在键值对列表中,获取键名与第一子分数区间匹配的键值对作为新的目标键值对后,返回执行获取分数在目标键值对的键值中命中的第一子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0015] 可选的,在根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对之前,还包括:

[0016] 获取查询用户的目标用户标识;

[0017] 根据用户标识与分数之间的映射关系,获取与目标用户标识对应的分数作为查询用户的分数。

[0018] 可选的,还包括:

[0019] 响应于与键值对列表匹配的分数调整请求,获取与分数调整请求匹配的至少一个待调整分数;

[0020] 根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整。

[0021] 可选的,分数调整请求包括新增分数请求,与新增分数请求匹配的待调整分数为新增分数;

[0022] 根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,包括:

[0023] 在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

[0024] 确定新增分数在目标键值对的键值中命中的第一目标子分数区间,并将与第一目标子分数区间对应的用户数加一;

[0025] 在键值对列表中,获取键名与第一目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定新增分数在目标键值对的键值中命中的第一目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0026] 可选的,分数调整请求包括删除分数请求,与删除分数请求匹配的待调整分数为删除分数;

[0027] 根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,包括:

[0028] 在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

[0029] 确定删除分数在目标键值对的键值中命中的第二目标子分数区间,并将与第二目标子分数区间对应的用户数减一;

[0030] 在键值对列表中,获取键名与第二目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定删除分数在目标键值对的键值中命中的第二目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0031] 可选的,分数调整请求包括调整分数请求,与调整分数请求匹配的待调整分数为调整前分数和调整后的分数;

[0032] 根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,包括:

[0033] 在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

[0034] 确定调整前分数在目标键值对的键值中命中的第三目标子分数区间,并将与第三目标子分数区间对应的用户数减一;

[0035] 在键值对列表中,获取键名与第三目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定调整前分数在目标键值对的键值中命中的第三目标子分数区间的操作,直至处理至与叶子节点对应的键值对;

[0036] 在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;

[0037] 确定调整后分数在目标键值对的键值中命中的第四目标子分数区间,并将与第四

目标子分数区间对应的用户数加一；

[0038] 在键值对列表中,获取键名与第四目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定调整后分数在目标键值对的键值中命中的第四目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0039] 可选的,树结构为16阶的平衡多叉树;

[0040] 其中,按照键值对中的指定分数区间从小到大的顺序以及树结构从左子节点到右子节点的顺序,将各键值对与对应的树节点关联。

[0041] 可选的,键值对列表存储于RocksDB中。

[0042] 第二方面,本发明实施例还提供了一种用户排名的查询装置,包括:

[0043] 查询模块,用于根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;

[0044] 其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;

[0045] 累加模块,用于在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;

[0046] 排名确定模块,用于在查询结束时,根据累加结果确定查询用户的用户排名。

[0047] 第三方面,本发明实施例还提供了一种电子设备,电子设备包括:

[0048] 一个或多个处理器;

[0049] 存储装置,用于存储一个或多个程序,

[0050] 当一个或多个程序被一个或多个处理器执行,使得一个或多个处理器实现本发明任意实施例提供的用户排名的查询方法。

[0051] 第四方面,本发明实施例还提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现本发明任意实施例提供的用户排名的查询方法。

[0052] 本发明实施例中,根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;在查询结束时,根据累加结果确定查询用户的用户排名,解决了现有技术中在大规模数据量下无法快速获取用户排名的问题,实现了通过设置与分数区间对应的键值对,节省数据存储空间,在大规模数据量下可以实时获取用户的具体排名。

附图说明

[0053] 图1a是本发明实施例一中的一种用户排名的查询方法的流程图;

[0054] 图1b是本发明实施例一中的一种树结构示意图;

[0055] 图2是本发明实施例二中的一种用户排名的查询方法的流程图;

[0056] 图3是本发明实施例三中的一种用户排名的查询装置的结构示意图;

[0057] 图4是本发明实施例四中的一种电子设备的结构示意图。

具体实施方式

[0058] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明,而非对本发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明相关的部分而非全部结构。

[0059] 实施例一

[0060] 图1a是本发明实施例一中的一种用户排名的查询方法的流程图,本实施例可适用于在大规模数据量下实时获取用户排名的情况,该方法可以由用户排名的查询装置来执行,该装置可以由硬件和/或软件来实现,并一般可以集成在提供用户排名查询服务的电子设备中,例如服务器中。如图1a所示,该方法包括:

[0061] 步骤110、根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联。

[0062] 其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间。

[0063] 本实施例中,查询用户是指待查询在某应用程序中的排名的用户,键值对列表用于查询任意分数对应的排名。键值对列表中包括有多个键值对,每个键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数。每个键值对与设定树结构中的一个节点关联,同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间,根节点对应的键值对中的指定分数区间是最大的,所有用户的分数都会落在根节点对应的指定分数区间中。

[0064] 示例性的,假设指定分数区间为 $[0, 5]$,该指定分数区间下的子分数区间包括 $[0, 2]$ 、 $[3, 4]$ 和 $[5, 5]$,并且,子分数区间 $[0, 2]$ 中包含的用户数为2,子分数区间 $[3, 4]$ 中包含的用户数为4,子分数区间 $[5, 5]$ 中包含的用户数为0,则对应的键值对中的键名为 $[0, 5]$,键值为 $[2, 4, 0]$ 。

[0065] 本实施例中,在确定查询用户的分数之后,可以根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,直至找到与叶子节点对应的键值对。由于本实施例采用一种新的数据结构表示键值对,即,键值对中的键名为一个指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数,并且各键值对之间存在树状结构关系,因此,使得通过很少次数的分数区间缩小就可以找到与叶子节点对应的键值对。也就是说,本实施例通过对分数进行分段查询,可以提高获取分数对应的排名的效率。

[0066] 可选的,在根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对之前,还可以包括:获取查询用户的目标用户标识;根据用户标识与分数之间的映射关系,获取与目标用户标识对应的分数作为查询用户的分数。

[0067] 本实施例中,用户标识可以是用户在当前应用程序中的注册账号、联系方式或者昵称等信息,用于唯一标识用户。响应于查询用户的排名查询请求,先获取查询用户的目标用户标识,根据目标用户标识查询用户标识与分数之间的映射关系,得到与目标用户标识对应的分数,将该分数作为查询用户的分数。其中用户标志与分数之间的映射关系,可以通过键值对存储,例如,键名为用户A,对应的键值为分数B。

[0068] 步骤120、在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加

大于分数的用户数。

[0069] 本实施例中,由于各键值对的键值中存储有指定分数区间下的各子分数区间包含的用户数,因此,在根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对过程中,可以根据分数所命中的各键值对中的子分数区间,在键值中找到并累加高于命中的子分数区间的其他子分数区间包含的用户数,以得到分数对应的用户排名。

[0070] 可选的,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数,可以包括:在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定分数在目标键值对的键值中命中的第一子分数区间,并确定大于第一子分数区间的全部第二子分数区间;将与各第二子分数区间分别对应的用户数与设定累加值进行累加,累加值具有设定初始值;在键值对列表中,获取键名与第一子分数区间匹配的键值对作为新的目标键值对后,返回执行获取分数在目标键值对的键值中命中的第一子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0071] 本实施例中,由于是通过自顶向下逐级查询键值对列表中的各键值对的方式,来实现由大分数区间到小分数区间的层层缩小,找到分数的位置,因此,开始查询时,先在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对。然后根据预先确定的子分数区间计算方式,确定目标键值对的指定分数区间下的各个子分数区间,确定分数在目标键值对的键值中命中的第一子分数区间,以及大于第一子分数区间的全部第二子分数区间。由于落在第二子分数区间内的分数对应的用户排名必然在查询用户之前,因此,将与各第二子分数区间分别对应的用户数与设定累加值进行累加,以通过累加值来记录分数高于查询用户的用户总数。其中,累加值的设定初始值为排名的初始值,例如,如果排名从0开始,则累加值的设定初始值为0,如果排名从1开始,则累加值的设定初始值为1。判断此时的第一子分数区间中是否只包含一个分数,若是,则第一子分数区间是叶子结点对应的键值对中的指定分数区间,否则,在键值对列表中,获取键名与第一子分数区间匹配的键值对作为新的目标键值对后,返回执行获取分数在目标键值对的键值中命中的第一子分数区间的操作,继续在第一子分数区间的各子分数区间中寻找与分数匹配的子分数区间,直至处理至与叶子节点对应的键值对。

[0072] 本实施例中,对父节点对应的键值对中的指定分数区间进行均分后得到各个子节点对应的键值对中的指定分数区间,例如,父节点对应的指定分数区间为 $[0, 9]$,则父节点的5个子节点对应的指定分数区间分别为 $[0, 1]$ 、 $[2, 3]$ 、 $[4, 5]$ 、 $[6, 7]$ 、 $[8, 9]$ 。如果指定分数区间不能刚好均分,则将多余的数据按顺序均匀的分配到各个子节点,例如,父节点对应的指定分数区间为 $[0, 9]$,则父节点的4个子节点对应的指定分数区间分别为 $[0, 2]$ 、 $[3, 5]$ 、 $[6, 7]$ 、 $[8, 9]$ 。

[0073] 本实施例中,当分数为double类型的值时,区间的划分将难以确定,考虑到使用uint64类型的数据作为分数时,在排行榜的应用中使用整型是足够业务使用的,因此可以将double类型的分数乘以一个固定大小的大整数,将分数从double类型转为整型,再进行区间划分。

[0074] 示例性的,如图1b所示,键值对列表中包括键值对:键名 $[0, 9]$ -键值 $[5, 3, 2]$,键名 $[0, 3]$ -键值 $[2, 2, 1]$,键名 $[4, 6]$ -键值 $[3, 0, 0]$,键名 $[7, 9]$ -键值 $[2, 0, 0]$,键名 $[0, 1]$ -键值 $[1, 1, 0]$,假设查询用户的分数为3,累加值的初始值为1,则获取3对应的用户排名的过程如

下:将键名[0,9]-键值[5,3,2]作为目标键值对,对区间[0,9]进行等分计算出子区间[0,3]、[4,6]以及[7,9],确定3命中的子区间[0,3]为第一子分数区间,子区间[4,6]和[7,9]为第二子分数区间,将累加值与子区间[4,6]和[7,9]包含的用户数3和2进行累加。此时子区间[0,3]中包含0-3共4个分数,需要进一步缩小区间范围,因此,将键名[0,3]-键值[2,2,1]作为新的目标键值对,重复上述过程,确定只包含一个分数的第一子分数区间[3,3],此时没有确定第二子区间,不需要对累加值进行累加,此时已经可以确定分数3对应的用户排名。

[0075] 本实施例中,通过将同一父节点下的各子节点对应的指定子分数区间包含的用户数都存储在父节点对应的键值对的键值中,使得在计算用户排名时,只需读取当前搜索到的父节点就可以得到最终结果,减少对父节点的子节点的读取。同时,为了减少数据冗余,父节点对应的键值中不存储父节点对应的指定分数区间包含的用户总数,以及父节点的各子节点对应的指定子分数区间。其中,父节点对应的指定分数区间包含的用户总数可以通过对父节点对应的键值中存储的各子分数区间分别包含的用户数进行累加得到,各子节点对应的指定子分数区间可以由父节点对应的键值计算出来。

[0076] 步骤130、在查询结束时,根据累加结果确定查询用户的用户排名。

[0077] 本实施例中,当查询到的第一子分数区间只包含一个分数时,则在与各第二子分数区间分别对应的用户数与设定累加值进行累加之后,当前的累加值就是查询用户的用户排名。例如,步骤120的示例中,查询用户的分数为3时,用户排名为6。

[0078] 本实施例中,由于节点的键名对应一个分数区间,而不是一个分数,因此在父节点对应的键值中存储各子节点的信息不会引入嵌套问题。并且,相同的分数对应同一个叶子节点,也对应同一个键值对,可以节省数据存储空间,而且相同分数得到的用户排名是相同的,更符合实际。

[0079] 本发明实施例中,根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;在查询结束时,根据累加结果确定查询用户的用户排名,解决了现有技术中在大规模数据量下无法快速获取用户排名的问题,实现了通过设置与分数区间对应的键值对,节省数据存储空间,在大规模数据量下可以实时获取用户的具体排名。

[0080] 实施例二

[0081] 图2是本发明实施例二中的一种用户排名的查询方法的流程图,本实施例在上述实施例的基础上进一步细化,提供对分数调整请求和待调整分数的描述,以及在不同描述下,根据待调整分数,对键值对列表中的各键值对中的子分数区间的用户数进行调整的具体步骤。下面结合图2对本申请第二实施例提供的一种用户排名的查询方法进行说明,包括以下步骤:

[0082] 步骤210、获取查询用户的目标用户标识,以及与目标用户标识对应的分数作为查询用户的分数。

[0083] 步骤220、根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键

值对列表中的键值对与设定树结构中的节点关联。

[0084] 可选的,键值对列表存储于RocksDB中。

[0085] 本实施例中,RocksDB是一个采用基于日志结构的合并树(Log-Structured Merge-Tree,LSM树)存储引擎的键值对数据库,具有完善的持久化机制,同时保证性能和安全性。RocksDB中的键值对是按照键名顺序存储的,因此可以良好的支持范围查询。LSM树的原理是把一棵大树拆分成N棵小树,它首先将数据写入到内存中,在内存中构建一颗有序小树,随着小树越来越大,内存的小树会输出到磁盘上。磁盘中的树定期可以做合并操作,生成一棵大树,以优化读性能。

[0086] 本实施例通过将键值对列表存储到RocksDB中,利用RocksDB或者LSM树的存储特性,实现实时获取用户排名功能不受内存大小限制,并且可以实现数据落盘永久保存。

[0087] 可选的,树结构为16阶的平衡多叉树;其中,按照键值对中的指定分数区间从小到大的顺序以及树结构从左子节点到右子节点的顺序,将各键值对与对应的树节点关联。

[0088] 本实施例中,考虑到RocksDB的读取性能受到与键值对列表对应的树的层数影响,层数太多时读取或更新时性能较低的特点,在RocksDB上构建了B树的结构,并对其进行优化。B树是多路平衡查找树,描述一颗B树时需要指定它的阶数,阶数表示一个节点最多有多少子节点。考虑到实际使用到的分数空间与最终生成的层次,为了使与键值对列表对应的树的层数不大于10层,将树设置为16阶的平衡多叉树,这样在uint32_t的空间中与键值对列表对应的树只有8层,每次计算用户排名时,只需要进行8次RocksDB的查找操作,可以大幅度提高查找用户排名的每秒查询率。

[0089] 本实施例中,由于RocksDB中的键值对是按照键名顺序存储的,因此,可以按照键值对中的指定分数区间从小到大的顺序以及树结构从左子节点到右子节点的顺序,将各键值对与对应的树节点关联,使得左子节点是低分区间,右子节点是高分区间,从而在键值中,很容易确定分数大于某个子分数区间的所有用户数,即便于计算用户排名。

[0090] 步骤230、在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数。

[0091] 步骤240、在查询结束时,根据累加结果确定查询用户的用户排名。

[0092] 本发明实施例中,根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;在查询结束时,根据累加结果确定查询用户的用户排名,解决了现有技术中在大规模数据量下无法快速获取用户排名的问题,实现了通过设置与分数区间对应的键值对,节省数据存储空间,在大规模数据量下可以实时获取用户的具体排名。

[0093] 需要说明的是,在上述实施例的基础上,还可以包括:响应于与键值对列表匹配的分数调整请求,获取与分数调整请求匹配的至少一个待调整分数;根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整。

[0094] 本实施例中,用户的分数发生变化时,会生成与键值对列表匹配的分数调整请求,

以根据用户的分数变化情况,对键值对列表中的键值对进行相应的更新调整。当分数变化时,理论上可以用Delete接口和Insert接口来实现,但实际上分数变化量一般较小,往往只需要更新几个分数区间甚至不需要更新分数区间,就可以达到目的。总体上,所需要更新的分数区间数量与分数变化量 det 是相关的,两者存在 $\log(\text{det})$ 的关系。

[0095] 可选的,分数调整请求包括新增分数请求,与新增分数请求匹配的待调整分数为新增分数。

[0096] 本实施例中,分数调整请求可以是新增分数请求,与新增分数请求匹配的待调整分数可以是新增分数,例如,当应用程序中增加了新用户,产生新用户的分数时,需要将新用户的分数增加到键值对列表中。

[0097] 可选的,根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,可以包括:在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定新增分数在目标键值对的键值中命中的第一目标子分数区间,并将与第一目标子分数区间对应的用户数加一;在键值对列表中,获取键名与第一目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定新增分数在目标键值对的键值中命中的第一目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0098] 本实施例中,当根据新增分数对键值对列表中的各键值对进行调整时,先在键值对列表中,获取与根节点对应的键值对,即与最大的指定分数区间对应的键值对作为目标键值对,新增分数必然在目标键值对的指定分数区间内,如果不在,则认为出错。然后确定目标键值对的指定分数区间下的各个子分数区间,确定新增分数在目标键值对的键值中命中的第一目标子分数区间,并将与第一目标子分数区间对应的用户数加一,判断第一目标子分数区间中是否只包含一个分数,如果是,则认为完成对键值对列表中的键值对的调整,否则,在键值对列表中,获取键名与第一目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定新增分数在目标键值对的键值中命中的第一目标子分数区间的操作,继续在第一目标子分数区间的各子分数区间中寻找与新增分数匹配的子分数区间,直至处理至与叶子节点对应的键值对。

[0099] 示例性的,假设键值对列表与图1b中的树结构对应,新增分数等于3,则根据新增分数对键值对列表中的各键值对进行调整的过程如下:将键名[0,9]-键值[5,3,2]作为目标键值对,确定分数3命中的子区间[0,3]为第一目标子分数区间,将目标键值对中的键值中子区间[0,3]包含的用户数加1,将键名[0,3]-键值[2,2,1]作为新的目标键值对,重复上述过程,确定只包含一个分数的第一目标子分数区间[3,3],将键值中子区间[3,3]包含的用户数加1,完成对键值对列表中的键值对的调整。

[0100] 可选的,分数调整请求包括删除分数请求,与删除分数请求匹配的待调整分数为删除分数。

[0101] 本实施例中,分数调整请求可以是删除分数请求,与删除分数请求匹配的待调整分数可以是删除分数,例如,当某用户注销了应用程序账号不再使用时,需要将该用户的分数从键值对列表中删除。

[0102] 可选的,根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,可以包括:在键值

对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定删除分数在目标键值对的键值中命中的第二目标子分数区间,并将与第二目标子分数区间对应的用户数减一;在键值对列表中,获取键名与第二目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定删除分数在目标键值对的键值中命中的第二目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0103] 本实施例中,当根据删除分数对键值对列表中的各键值对进行调整时,先在键值对列表中,获取与最大的指定分数区间对应的键值对作为目标键值对,删除分数必然在目标键值对的指定分数区间内,如果不在,则认为出错。然后确定目标键值对的指定分数区间下的各个子分数区间,确定删除分数在目标键值对的键值中命中的第二目标子分数区间,并将与第二目标子分数区间对应的用户数减一,判断第二目标子分数区间中是否只包含一个分数,如果是,则认为完成对键值对列表中的键值对的调整,否则,在键值对列表中,获取键名与第二目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定删除分数在目标键值对的键值中命中的第二目标子分数区间的操作,继续在第二目标子分数区间的各子分数区间中寻找与删除分数匹配的子分数区间,直至处理至与叶子节点对应的键值对。

[0104] 其中,如果将与第二目标子分数区间对应的用户数减一之后,用户数变为-1,则认为键值对列表中不存在删除分数,返回错误。如果将与第二目标子分数区间对应的用户数减一之后,目标键值对的键值中的所有数据都是0,则认为该父节点下没有子节点,可以删除目标键值对。

[0105] 示例性的,假设键值对列表与图1b中的树结构对应,删除分数等于3,则根据删除分数对键值对列表中的各键值对进行调整的过程如下:将键名[0,9]-键值[5,3,2]作为目标键值对,确定分数3命中的子区间[0,3]为第二目标子分数区间,将目标键值对中的键值中子区间[0,3]包含的用户数减1,将键名[0,3]-键值[2,2,1]作为新的目标键值对,重复上述过程,确定只包含一个分数的第二目标子分数区间[3,3],将键值中子区间[3,3]包含的用户数减1,完成对键值对列表中的键值对的调整。

[0106] 可选的,分数调整请求包括调整分数请求,与调整分数请求匹配的待调整分数为调整前分数和调整后分数。

[0107] 本实施例中,分数调整请求可以是调整分数请求,与调整分数请求匹配的待调整分数可以是调整前分数和调整后分数,例如,当某用户的分数上升时,需要在键值对列表中将该用户的分数从调整前分数替换为调整后分数。

[0108] 可选的,根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整,可以包括:在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定调整前分数在目标键值对的键值中命中的第三目标子分数区间,并将与第三目标子分数区间对应的用户数减一;在键值对列表中,获取键名与第三目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定调整前分数在目标键值对的键值中命中的第三目标子分数区间的操作,直至处理至与叶子节点对应的键值对;

[0109] 在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定调整后分数在目标键值对的键值中命中的第四目标子分数区间,并将与第四目标子分数区间

对应的用户数加一；在键值对列表中，获取键名与第四目标子分数区间匹配的键值对作为新的目标键值对后，返回执行确定调整后分数在目标键值对的键值中命中的第四目标子分数区间的操作，直至处理至与叶子节点对应的键值对。

[0110] 本实施例中，根据调整前分数和调整后分数对键值对列表中的各键值对进行调整的过程，实际上就是将调整前分数从键值对列表中删除，并将调整后分数新增到键值对列表中的过程。其中，将调整前分数从键值对列表中删除的过程与根据删除分数对键值对列表中的各键值对进行调整的过程相同，将调整后分数新增到键值对列表中的过程与根据新增分数对键值对列表中的各键值对进行调整的过程相同，此处不再详细描述。

[0111] 本实施例中，根据调整前分数和调整后分数对键值对列表中的各键值对进行调整的过程也可以如下所述：在键值对列表中，获取与树结构的根节点对应的键值对作为目标键值对；确定调整前分数在目标键值对的键值中命中的第三目标子分数区间，以及调整后分数在目标键值对的键值中命中的第四目标子分数区间；如果第三目标子分数区间与第四目标子分数区间相等，则该子分数区间中的用户数不变，在获取键名与第三目标子分数区间匹配的键值对作为新的目标键值对后，返回执行确定调整前分数在目标键值对的键值中命中的第三目标子分数区间的操作，直至第三目标子分数区间与第四目标子分数区间不相等，此时，将与第三目标子分数区间对应的用户数减一，将与第四目标子分数区间对应的用户数加一。

[0112] 然后，获取键名与第三目标子分数区间匹配的键值对作为新的目标键值对，确定调整前分数在目标键值对的键值中命中的第二目标子分数区间，并将与第二目标子分数区间对应的用户数减一；在键值对列表中，获取键名与第二目标子分数区间匹配的键值对作为新的目标键值对后，返回执行确定调整前分数在目标键值对的键值中命中的第二目标子分数区间的操作，直至处理至与叶子节点对应的键值对。

[0113] 然后，获取键名与第四目标子分数区间匹配的键值对作为新的目标键值对，确定调整后分数在目标键值对的键值中命中的第一目标子分数区间，并将与第一目标子分数区间对应的用户数加一；在键值对列表中，获取键名与第一目标子分数区间匹配的键值对作为新的目标键值对后，返回执行确定调整后分数在目标键值对的键值中命中的第一目标子分数区间的操作，直至处理至与叶子节点对应的键值对。

[0114] 示例性的，假设键值对列表与图1b中的树结构对应，调整前分数等于3，调整后分数等于6，则根据调整前分数和调整后分数对键值对列表中的各键值对进行调整的过程如下：将键名[0,9]-键值[5,3,2]作为目标键值对，确定分数3命中的子区间[0,3]为第三目标子分数区间，确定分数6命中的子区间[4,6]为第四目标子分数区间，第三目标子分数区间与第四目标子分数区间不相等，因此，将目标键值对中的键值中子区间[0,3]包含的用户数减1，将键值中子区间[4,6]包含的用户数加1。然后，将键名[0,3]-键值[2,2,1]作为新的目标键值对，确定分数3命中的子区间[3,3]中只包含分数3，因此将目标键值对中的键值中子区间[3,3]包含的用户数减1。然后，将键名[4,6]-键值[3,0,0]作为新的目标键值对，确定分数6命中的子区间[6,6]中只包含分数6，因此将目标键值对中的键值中子区间[6,6]包含的用户数加1，完成对键值对列表中的各键值对的调整。

[0115] 实施例三

[0116] 图3是本发明实施例三中的一种用户排名的查询装置的结构示意图，本实施例可

适用于在大规模数据量下实时获取用户排名的情况,该装置可以由硬件和/或软件来实现,并一般可以集成在提供用户排名查询服务的电子设备中,例如服务器中。如图3所示,该用户排名的查询装置包括:

[0117] 查询模块310,用于根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;

[0118] 其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;

[0119] 累加模块320,用于在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;

[0120] 排名确定模块330,用于在查询结束时,根据累加结果确定查询用户的用户排名。

[0121] 本发明实施例中,根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;在查询结束时,根据累加结果确定查询用户的用户排名,解决了现有技术中在大规模数据量下无法快速获取用户排名的问题,实现了通过设置与分数区间对应的键值对,节省数据存储空间,在大规模数据量下可以实时获取用户的具体排名。

[0122] 可选的,累加模块320,具体用于:在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定分数在目标键值对的键值中命中的第一子分数区间,并确定大于第一子分数区间的全部第二子分数区间;将与各第二子分数区间分别对应的用户数与设定累加值进行累加,累加值具有设定初始值;在键值对列表中,获取键名与第一子分数区间匹配的键值对作为新的目标键值对后,返回执行获取分数在目标键值对的键值中命中的第一子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0123] 可选的,还包括:分数获取模块,用于在根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对之前,获取查询用户的目标用户标识;根据用户标识与分数之间的映射关系,获取与目标用户标识对应的分数作为查询用户的分数。

[0124] 可选的,还包括:调整模块,用于响应于与键值对列表匹配的分数调整请求,获取与分数调整请求匹配的至少一个待调整分数;根据至少一个待调整分数,自顶向下逐级查询键值对列表中的各键值对,并对待调整分数所命中的各键值对中的子分数区间的用户数进行调整。

[0125] 可选的,分数调整请求包括新增分数请求,与新增分数请求匹配的待调整分数为新增分数;

[0126] 调整模块,包括新增单元,用于:在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定新增分数在目标键值对的键值中命中的第一目标子分数区间,并将与第一目标子分数区间对应的用户数加一;在键值对列表中,获取键名与第一目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定新增分数在目标键值对的键值中命中的第一目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0127] 可选的,分数调整请求包括删除分数请求,与删除分数请求匹配的待调整分数为删除分数;

[0128] 调整模块,包括删除单元,用于:在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定删除分数在目标键值对的键值中命中的第二目标子分数区间,并将与第二目标子分数区间对应的用户数减一;在键值对列表中,获取键名与第二目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定删除分数在目标键值对的键值中命中的第二目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0129] 可选的,分数调整请求包括调整分数请求,与调整分数请求匹配的待调整分数为调整前分数和调整后的分数;

[0130] 调整模块,包括分数调整单元,用于:在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定调整前分数在目标键值对的键值中命中的第三目标子分数区间,并将与第三目标子分数区间对应的用户数减一;在键值对列表中,获取键名与第三目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定调整前分数在目标键值对的键值中命中的第三目标子分数区间的操作,直至处理至与叶子节点对应的键值对;

[0131] 在键值对列表中,获取与树结构的根节点对应的键值对作为目标键值对;确定调整后分数在目标键值对的键值中命中的第四目标子分数区间,并将与第四目标子分数区间对应的用户数加一;在键值对列表中,获取键名与第四目标子分数区间匹配的键值对作为新的目标键值对后,返回执行确定调整后分数在目标键值对的键值中命中的第四目标子分数区间的操作,直至处理至与叶子节点对应的键值对。

[0132] 可选的,树结构为16阶的平衡多叉树;其中,按照键值对中的指定分数区间从小到大的顺序以及树结构从左子节点到右子节点的顺序,将各键值对与对应的树节点关联。

[0133] 可选的,键值对列表存储于RocksDB中。

[0134] 本发明实施例所提供的用户排名的查询装置可执行本发明任意实施例所提供的用户排名的查询方法,具备执行方法相应的功能模块和有益效果。

[0135] 实施例四

[0136] 图4是本发明实施例四公开的一种电子设备的结构示意图。图4示出了适于用来实现本发明实施方式的示例性设备12的框图。图4显示的设备12仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0137] 如图4所示,设备12以通用计算设备的形式表现。设备12的组件可以包括但不限于:一个或者多个处理器或者处理单元16,系统存储器28,连接不同系统组件(包括系统存储器28和处理单元16)的总线18。

[0138] 总线18表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构(ISA)总线,微通道体系结构(MAC)总线,增强型ISA总线、视频电子标准协会(VESA)局域总线以及外围组件互连(PCI)总线。

[0139] 设备12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被设备12访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0140] 系统存储器28可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)30和/或高速缓存存储器32。设备12可以进一步包括其它可移动/不可移动的、

易失性/非易失性计算机系统存储介质。仅作为举例,存储系统34可以用于读写不可移动的、非易失性磁介质(图4未显示,通常称为“硬盘驱动器”)。尽管图4中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如CD-ROM,DVD-ROM或者其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线18相连。存储器28可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0141] 具有一组(至少一个)程序模块42的程序/实用工具40,可以存储在例如存储器28中,这样的程序模块42包括但不限于操作系统、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。程序模块42通常执行本发明所描述的实施例中的功能和/或方法。

[0142] 设备12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信,还可与一个或者多个使得用户能与该设备12交互的设备通信,和/或与使得该设备12能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口22进行。并且,设备12还可以通过网络适配器20与一个或多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。如图所示,网络适配器20通过总线18与设备12的其它模块通信。应当明白,尽管图中未示出,可以结合设备12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系统、磁带驱动器以及数据备份存储系统等。

[0143] 处理单元16通过运行存储在系统存储器28中的程序,从而执行各种功能应用以及数据处理,例如实现本发明实施例所提供的用户排名的查询方法。

[0144] 也即:实现一种用户排名的查询方法,包括:根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;

[0145] 其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;

[0146] 在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;

[0147] 在查询结束时,根据累加结果确定查询用户的用户排名。

[0148] 实施例五

[0149] 本发明实施例五还公开了一种计算机存储介质,其上存储有计算机程序,该程序被处理器执行时实现一种用户排名的查询方法,包括:

[0150] 根据查询用户的分数,自顶向下逐级查询键值对列表中的各键值对,键值对列表中的键值对与设定树结构中的节点关联;

[0151] 其中,键值对的键名为指定分数区间,键值为指定分数区间下的各子分数区间分别包含的用户数;同一父节点下全部子节点对应的键值对中的指定分数区间的总和为父节点对应的键值对中的指定分数区间;

[0152] 在查询过程中,根据分数所命中的各键值对中的子分数区间,逐级累加大于分数的用户数;

[0153] 在查询结束时,根据累加结果确定查询用户的用户排名。

[0154] 本发明实施例的计算机存储介质,可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机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、便携式计算机磁盘、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、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0155] 计算机可读的信号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,其中承载了计算机可读的程序代码。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。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0156] 计算机可读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、电线、光缆、RF等等,或者上述的任意合适的组合。

[0157] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码,程序设计语言包括面向对象的程序设计语言——诸如Java、Smalltalk、C++,还包括常规的过程式程序设计语言——诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0158] 注意,上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解,本发明不限于这里所述的特定实施例,对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此,虽然通过以上实施例对本发明进行了较为详细的说明,但是本发明不仅仅限于以上实施例,在不脱离本发明构思的情况下,还可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。

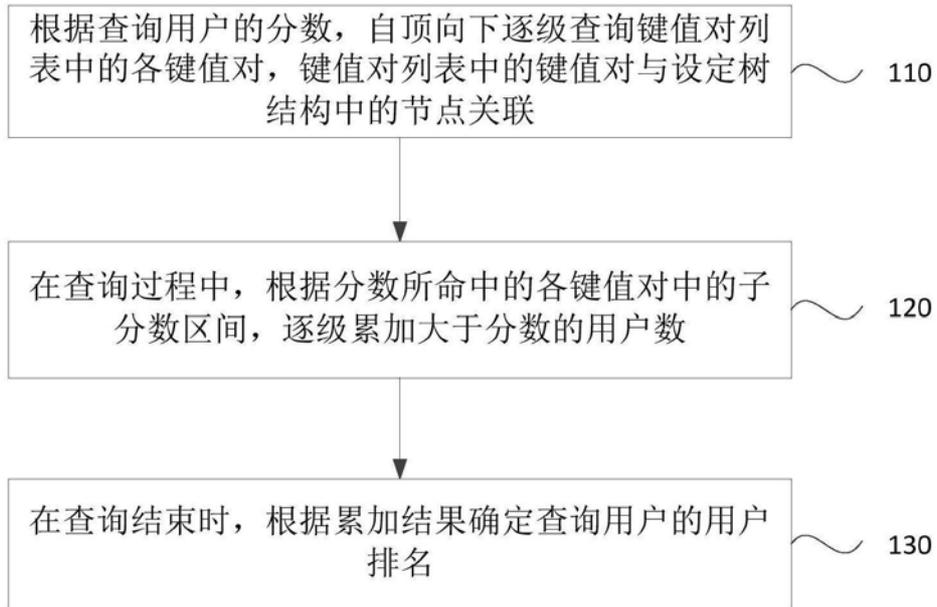


图1a

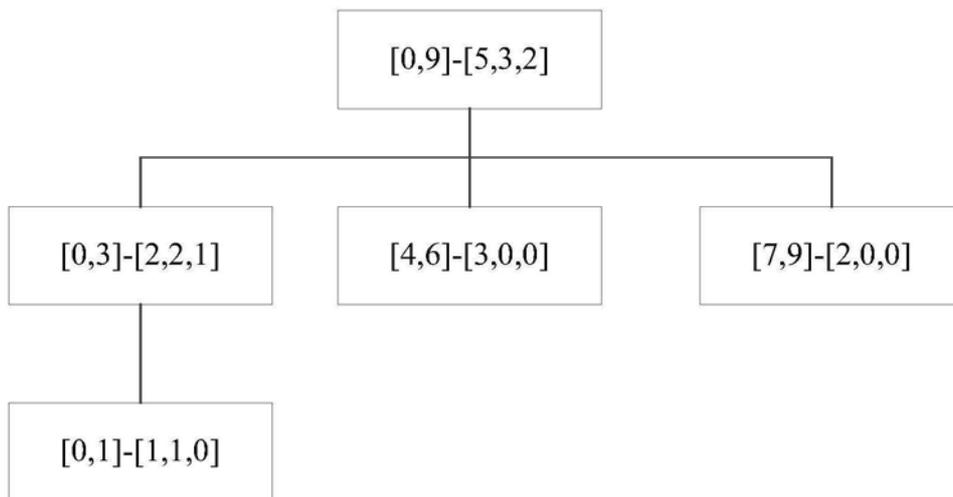


图1b

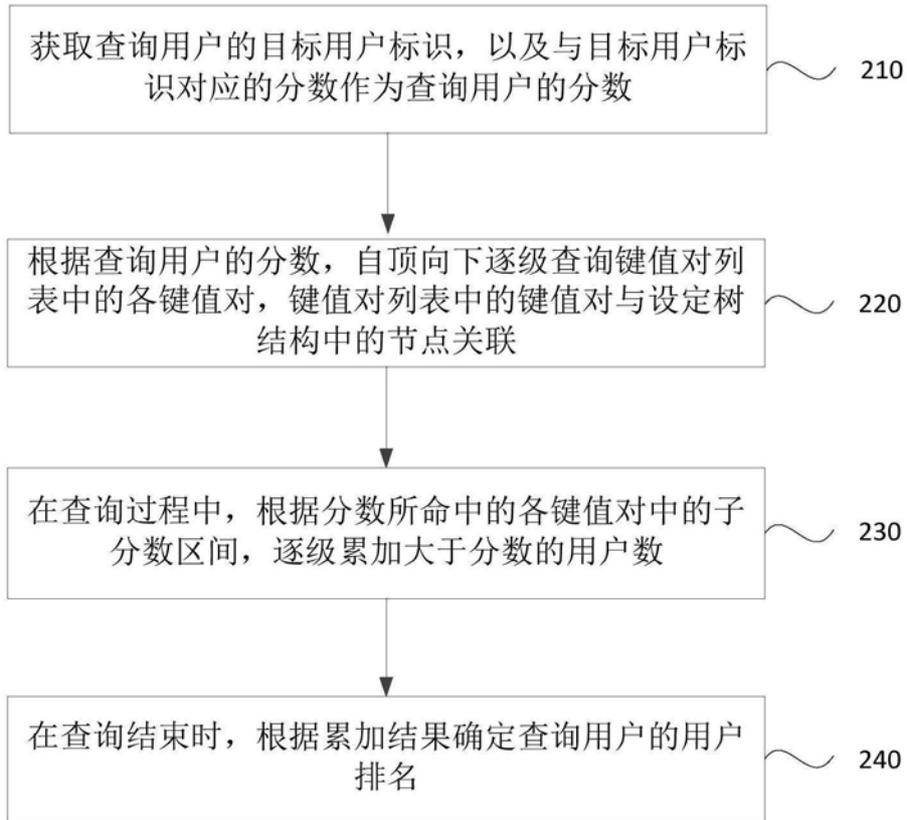


图2



图3

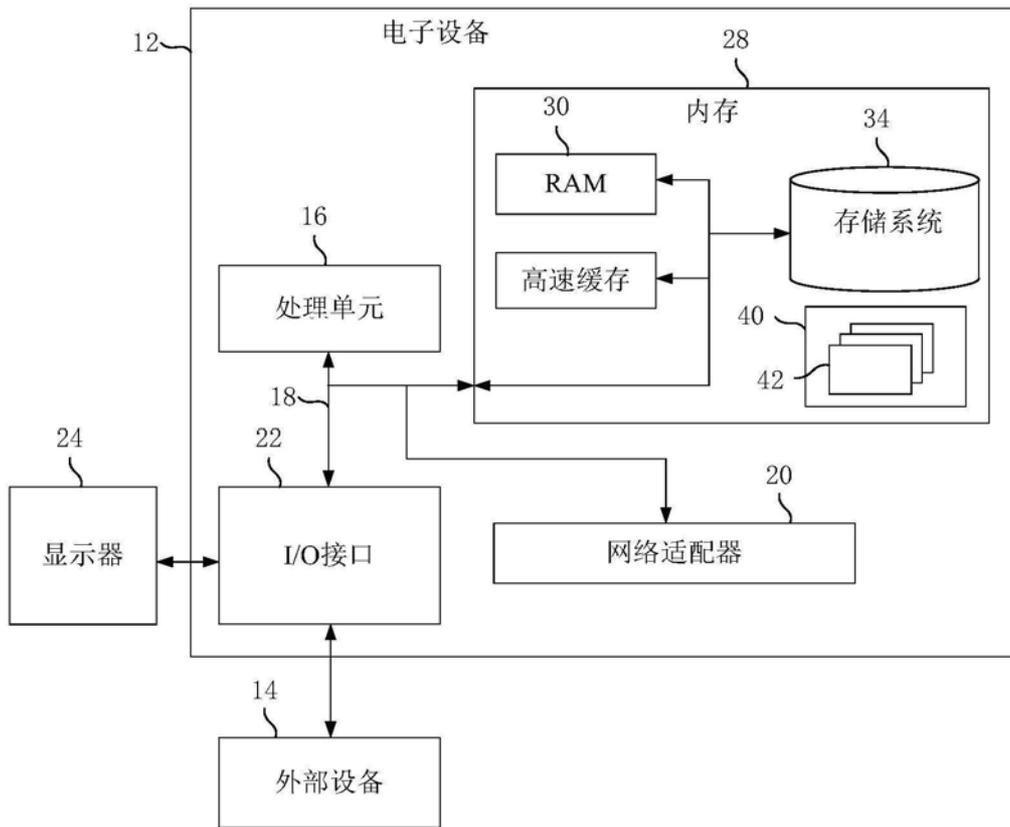


图4