(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 115242727 B
(45)授权公告日 2023.08.08

(21)申请号 202210831645.4

(22)申请日 2022.07.15

(65)同一申请的已公布的文献号
申请公布号 CN 115242727 A

(43)申请公布日 2022.10.25

(73)专利权人 深圳市腾讯计算机系统有限公司
地址 518057 广东省深圳市南山区高新区科技中一路腾讯大厦35层
专利权人 华中科技大学

(72)发明人 邢广杰 王桦 周可 杨光超 范龙隆

(74)专利代理机构 北京志霖恒远知识产权代理有限公司 11435
专利代理师 宿慧妮

(51)Int.Cl.
*H04L 47/52*(2022.01)
*H04L 47/6275*(2022.01)
*H04L 67/60*(2022.01)

(56)对比文件
CN 104079501 A,2014.10.01
CN 110209343 A,2019.09.06
US 2008189350 A1,2008.08.07
US 2021058337 A1,2021.02.25
CN 113961334 A,2022.01.21
CN 108848037 A,2018.11.20
CN 109756428 A,2019.05.14

审查员 陈欢

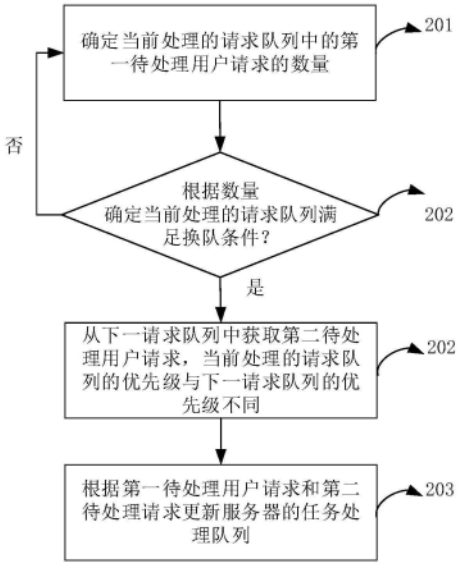
权利要求书2页 说明书16页 附图5页

(54)发明名称

用户请求处理方法、装置、设备和介质

(57)摘要

本申请公开了一种用户请求处理方法、装置、设备和介质，涉及通信技术领域，具体涉及数据处理技术领域。该方法包括：确定当前处理的请求队列中的第一待处理用户请求的数量；请求队列用于存储优先级相同的用户请求；若根据数量确定当前处理的请求队列满足换队条件，则从下一请求队列中获取第二待处理用户请求；当前处理的请求队列的优先级与下一请求队列的优先级不同；根据第一待处理用户请求和第二待处理请求更新服务器的任务处理队列；任务处理队列用于存储服务器待响应的用户请求。能够合理分配带宽，提高整体带宽的利用率。




CN 115242727 B

1.一种用户请求处理方法,其特征在于,所述方法包括:

从当前处理的请求队列中获取第一待处理用户请求,并确定所述当前处理的请求队列中所述第一待处理用户请求的数量;请求队列用于存储优先级相同的用户请求;所述第一待处理用户请求为第一阈值限定下,在所述当前处理的请求队列中可获取的用户请求;所述第一阈值用于表征一个处理周期中,在所述当前处理的请求队列中允许获取的用户请求的上限数量;所述一个处理周期用于按照优先级顺序遍历访问所有请求队列;

若根据所述第一待处理用户请求的数量确定所述当前处理的请求队列满足换队条件,则从下一请求队列中获取第二待处理用户请求;所述第二待处理用户请求为第二阈值限定下,在所述下一请求队列中可获取的用户请求;所述第二阈值用于表征所述一个处理周期中,在所述下一请求队列中允许获取的用户请求的上限数量;所述当前处理的请求队列的优先级高于所述下一请求队列的优先级;所述换队条件包括所述第一待处理用户请求的数量为所述第一阈值限定下,在所述当前处理的请求队列中可获取到用户请求的最大数量;所述最大数量是由所述第一阈值与所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量综合决定;其中,若所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量大于或等于所述第一阈值,则所述最大数量为所述第一阈值;若所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量小于所述第一阈值,则所述最大数量为所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量;

根据所述第一待处理用户请求和所述第二待处理用户请求存储服务器的任务处理队列;所述任务处理队列用于存储所述服务器待响应的用户请求。

2.根据权利要求1所述的用户请求处理方法,其特征在于,所述方法还包括:

根据所述当前处理的请求队列的优先级在所有优先级中所占的权重,确定所述第一阈值。

3.根据权利要求1或2所述的用户请求处理方法,其特征在于,所述根据所述第一待处理用户请求和所述第二待处理用户请求存储服务器的任务处理队列,包括:

若从所述第一待处理用户请求对应的第一令牌集合中获取到所述第一待处理用户请求所需的令牌,则将所述第一待处理用户请求添加至所述任务处理队列;

若从所述第二待处理用户请求对应的第二令牌集合中获取到所述第二待处理用户请求所需的令牌,则将所述第二待处理用户请求添加至所述任务处理队列。

4.根据权利要求3所述的用户请求处理方法,其特征在于,所述方法还包括:

根据各请求队列的优先级,确定各所述请求队列对应的令牌集合的令牌投放速率。

5.根据权利要求1或2所述的用户请求处理方法,其特征在于,所述方法还包括:

接收用户端发送的用户请求,确定所述用户请求的优先级;

根据所述用户请求的优先级将所述用户请求添加至所述优先级的请求队列。

6.根据权利要求5所述的用户请求处理方法,其特征在于,所述确定所述第一待处理用户请求的优先级,包括:

根据所述用户请求的源端口,确定所述用户请求的优先级。

7.根据权利要求6所述的用户请求处理方法,其特征在于,所述根据所述用户请求的优先级将所述用户请求添加至所述当前处理的请求队列,包括:

基于所述用户请求的优先级，确定所述请求队列的标识；

根据所述请求队列的标识，将所述用户请求添加至所述请求队列所对应的数据链表中。

8.一种用户请求处理装置，其特征在于，所述装置包括：

确定单元，用于从当前处理的请求队列中获取第一待处理用户请求，并确定所述当前处理的请求队列中所述第一待处理用户请求的数量；请求队列用于存储优先级相同的用户请求；所述第一待处理用户请求为第一阈值限定下，在所述当前处理的请求队列中可获取的用户请求；所述第一阈值用于表征一个处理周期中，在所述当前处理的请求队列中允许获取的用户请求的上限数量；所述一个处理周期用于按照优先级顺序遍历访问所有请求队列；

处理单元，用于若根据所述第一待处理用户请求的数量确定所述当前处理的请求队列满足换队条件，则从下一请求队列中获取第二待处理用户请求；所述第二待处理用户请求为第二阈值限定下，在所述下一请求队列中可获取的用户请求；所述第二阈值用于表征所述一个处理周期中，在所述下一请求队列中允许获取的用户请求的上限数量；所述当前处理的请求队列的优先级高于所述下一请求队列的优先级；所述换队条件包括所述第一待处理用户请求的数量为所述第一阈值限定下，在所述当前处理的请求队列中可获取到用户请求的最大数量；所述最大数量是由所述第一阈值与所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量综合决定；其中，若所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量大于或等于所述第一阈值，则所述最大数量为所述第一阈值；若所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量小于所述第一阈值，则所述最大数量为所述当前处理的请求队列在所述一个处理周期中所存储的用户请求的数量；

更新单元，用于根据所述第一待处理用户请求和所述第二待处理用户请求存储服务器的任务处理队列；所述任务处理队列用于存储所述服务器待响应的用户请求。

9.一种计算机设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，其特征在于，所述处理器执行所述程序时，实现如权利要求1至7任一项所述的方法。

10.一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至7中任一项所述的方法。

# 用户请求处理方法、装置、设备和介质

## 技术领域

[0001] 本申请一般涉及通信技术领域，具体涉及数据处理技术领域，尤其涉及一种用户请求处理方法、装置、设备和介质。

## 背景技术

[0002] 随着网络通信技术的快速发展，用户对数据传输速率的要求也越来越高，当用户请求出现拥塞时，如何更好的对流量进行分配也显得越来越重要。

[0003] 现如今的用户请求处理技术在通过优先级的高低来区分不同的用户时，会为高优先级的用户提前预留部分带宽来应对这些用户请求的突发情况。但是当这些高优先级的用户请求没有突发情况时，这些预留带宽就处于空闲状态，如此导致整个带宽的利用率不高。

## 发明内容

[0004] 鉴于现有技术中的上述缺陷或不足，期望提供一种用户请求处理方法、装置、设备和介质，能够合理分配带宽，提高整个带宽的利用率。

[0005] 第一方面，本申请提供了一种用户请求处理方法，该方法包括：确定当前处理的请求队列中的第一待处理用户请求的数量；请求队列用于存储优先级相同的用户请求；若根据数量确定当前处理的请求队列满足换队条件，则从下一请求队列中获取第二待处理用户请求；当前处理的请求队列的优先级与下一请求队列的优先级不同；根据第一待处理用户请求和第二待处理用户请求更新服务器的任务处理队列；任务处理队列用于存储服务器待响应的用户请求。

[0006] 第二方面，本申请提供了一种用户请求处理装置，该用户请求处理装置包括：确定单元，用于确定当前处理的请求队列中的第一待处理用户请求的数量；请求队列用于存储优先级相同的用户请求；处理单元，用于若根据数量确定当前处理的请求队列满足换队条件，则从下一请求队列中获取第二待处理用户请求；当前处理的请求队列的优先级与下一请求队列的优先级不同；更新单元，用于根据第一待处理用户请求和第二待处理用户请求更新服务器的任务处理队列；任务处理队列用于存储服务器待响应的用户请求。

[0007] 在本申请的另一实施例中，第一待处理用户请求为第一阈值限定下，在当前处理的请求队列中可获取的用户请求；第一阈值用于表征一个处理周期中，在当前处理的请求队列中允许获取的用户请求的上限数量，处理周期用于遍历所有请求队列；换队条件，包括：

[0008] 数量为第一阈值限定下，在当前处理的请求队列中可获取到用户请求的最大数量。

[0009] 在本申请的另一实施例中，处理单元，具体用于根据当前处理的请求队列的优先级在所有优先级中所占的权重，确定第一阈值。

[0010] 在本申请的另一实施例中，更新单元，具体用于，

[0011] 若从第一待处理用户请求对应的第一令牌集合中获取到第一待处理用户请求所

需的令牌，则将第一待处理用户请求添加至任务处理队列。

[0012] 若从第二待处理用户请求对应的第二令牌集合中获取到第二待处理用户请求所需的令牌，则将第二待处理用户请求添加至任务处理队列。

[0013] 在本申请的另一实施例中，更新单元，具体用于根据各请求队列的优先级，确定各请求队列对应的令牌集合的令牌投放速率。

[0014] 在本申请的另一实施例中，接收单元，用于接收用户端发送的用户请求。

[0015] 处理单元，用于确定用户请求的优先级，根据用户请求的优先级将用户请求添加至优先级的请求队列。

[0016] 在本申请的另一实施例中，处理单元，具体用于根据用户请求的源端口，确定用户请求的优先级。

[0017] 在本申请的另一实施例中，处理单元，具体用于基于用户请求的优先级，确定请求队列的标识；根据请求队列的标识，将用户请求添加至请求队列所对应的数据链表中。

[0018] 第三方面，本申请实施例提供了一种计算机设备，包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序，该处理器执行该程序时实现如本申请实施例描述的方法。

[0019] 第四方面，本申请实施例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如本申请实施例描述的方法。

[0020] 第五方面，本申请实施例提供一种计算机程序产品，该计算机程序产品包括指令，当该指令被运行时，使得如本申请实施例描述的方法被执行。

[0021] 本申请提出的用户请求处理方法、装置、设备和介质，由于现有技术需要针对高优先级的用户预留出一部分带宽来应对这些用户请求的突发情况，当这些用户请求没有突发情况时，会出现整个带宽的利用率不高的问题。因此，本申请通过设置不同优先级的请求队列来存储对应优先级的用户请求，对用户请求按照优先级进行区别，并按照设定的换队条件来对各个优先级的用户请求进行流量调控。具体的，确定当前处理的请求队列中的第一待处理用户请求的数量，若根据该数量确定当前处理的请求队列满足换队条件时，则从优先级不同于当前处理的请求队列的下一请求队列中获取第二待处理用户请求，并根据第一待处理用户请求和第二待处理用户请求更新服务器中用于存储该服务器待响应的用户请求的任务处理队列。这样，通过换队条件实现不同优先级用户请求之间的切换，使对用户请求的调度不止于一个优先级的请求队列，而是在满足换队条件的情况下，对其他优先级的用户请求进行调度，保障了各优先级的用户请求的带宽使用。因此，本申请针对优先级的这种用户等级制度下，引入了多队列机制，并结合换队条件这一队列切换制度，不仅实现了高低优先级用户请求的灵活调度，并且能够合理分配带宽，提高了带宽的利用率。

[0022] 本申请附加的方面和优点将在下面的描述中部分给出，部分将从下面的描述中变得明显，或通过本申请的实践了解到。

## 附图说明

[0023] 通过阅读参照以下附图所作的对非限制性实施例所作的详细描述，本申请的其它特征、目的和优点将会变得更明显：

[0024] 图1为本申请实施例提供的用户请求处理系统的结构示意图；

[0025] 图2为本申请实施例提供的用户请求处理方法的流程示意图；

[0026] 图3为本申请实施例提供的请求队列的结构示意图；

[0027] 图4为本申请实施例提供的令牌集合的效果示意图；

[0028] 图5为本申请实施例提供的又一用户请求处理方法的流程示意图；

[0029] 图6为本申请实施例提供的数据链表的结构示意图；

[0030] 图7为本申请实施例提供的用户请求处理效果示意图；

[0031] 图8为本申请实施例提供的用户请求处理装置的结构示意图；

[0032] 图9为本申请实施例提供的计算机设备的结构示意图。

## 具体实施方式

[0033] 下面结合附图和实施例对本申请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相关发明，而非对该发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与发明相关的部分。

[0034] 需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。下面将参考附图并结合实施例来详细说明本申请。

[0035] 现如今的用户请求处理技术在通过优先级的高低来区分不同的用户时，会为高优先级的用户提前预留部分带宽来应对这些用户请求的突发情况。由于处理用户请求的服务提供商所提供的带宽有限；因此，总体能够使用的带宽需要控制在其所提供的带宽范围之内，当这些高优先级的用户请求没有突发情况时，这些预留带宽就处于空闲状态，如此导致整个带宽的利用率不高。

[0036] 基于此，本申请实施例提供一种用户请求处理方法、装置、设备和介质，其主要原理是：通过配置多个优先级的请求队列，将不同优先级的用户请求存储至对应优先级的请求队列中，首先确定当前处理的请求列中的第一待处理用户请求的数量，并根据该数量确定当前处理的请求队列是否满足换队条件，若满足，可以进行请求队列之间的切换，具体是从优先级不同于当前处理的请求队列的下一请求队列中获取第二待处理用户请求，并根据第一待处理用户请求和第二待处理用户请求更新服务器的任务处理队列，从而等待服务器响应。

[0037] 这里需要说明的是，用户请求的突发状况是指在预设的时间段内接收到来自用户端超过预设数量的用户请求。同理，针对本申请实施例中各优先级的请求队列，当某一优先级的请求队列接收到来自用户端超过预设数量的用户请求可以理解为该优先级的用户请求出现突发状况。

[0038] 图1为本申请实施例提供的一种用户请求处理系统的结构示意图。本申请实施例所提供的用户请求处理方法可应用于该用户请求处理系统100。参考图1，该用户请求处理系统100包括用户端101、中间设备102、服务器103。其中，中间设备102是位于用户端101与服务器103之间的转发设备，用于实现用户端101与服务器103之间通信数据的转发，其可以是如路由器或者交换机等转发设备。

[0039] 在一个实施例中，用户端101可以指代终端设备或者终端设备中的软件或硬件装置，例如，软件装置可以是应用程序，硬件装置可以是运行该应用程序的芯片等。

[0040] 示例性的，终端设备可以是包括但不限于个人计算、平台电脑、智能手机、可穿戴

设备、车载终端等设备，本申请实施例对此不作限定。服务器103可以是独立的物理服务器，也可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供用户请求处理技术的基础云计算服务的云服务器。

[0041] 可以理解的是，用于用户请求处理方法可以是中间设备102，也可以是服务器，该服务器可以是用户请求处理系统中的服务器103，也可以是其他服务器，本申请实施例对此不作限定。

[0042] 作为一个示例，当中间设备102用于执行本申请实施例提供的用户请求处理方法，中间设备102包括队列管理(Queue Manager，QM)模块，队列管理模块中部署有至少两个不同优先级的请求队列，例如图1中所示出的请求队列q0和请求队列q1；请求队列q0用于存储一个优先级的用户请求，请求队列q1用于存储与请求队列q0的优先级不同的另一优先级的用户请求。服务器103中部署有任务处理队列，该任务处理队列可以按照来自请求队列的用户请求的时间先后顺序，存储服务器103待响应的用户请求。当然，当服务器用于执行本申请实施例提供的用户请求处理方法时，服务器中同样部署有至少两个不同优先级的请求队列，请求队列的部署示例可参考上述的中间设备102部署的请求队列示例，此处不在赘述。

[0043] 在一种实现方式中，根据优先级(priority)对不同的用户端101进行分类，每个用户端101都可以设置用户侧在发送用户请求是所对应的最大带宽(max_bw)和最大深度(max_d epth)，用户端101会不断尝试向队列管理(Queue Manager，QM)模块发送用户请求，在用户端101的用户请求未达到最大带宽和/或最大深度时，即可成功向队列管理模块发送一个用户请求，不同优先级的用户端101发送的用户请求会落入队列管理模块不同的请求队列中。

[0044] 用户端101主要包括以下属性：用户的优先级priority、用户所属的线程编号thread_id、用户在该线程中的编号user_num、最大深度max_depth、最大带宽max_bw、当前深度now_depth、发送请求的无锁队列send_pipe、接收回包的无锁队列recv_pipe、每秒请求数qps统计对象flow、时延统计对象latency、计时器flow_timer。用户端101能够实现的相关功能包括：1、初始化和内存分配；2、更新自身的状态，具体包括取出recv_pipe中的回包(即来自服务器针对用户请求的处理结果)，每取出一次计算延时统计到latency中，并且对now_depth减1；3、发送用户请求，具体为首先更新自身状态，然后检查flow统计的当前的qps，是否超过max_bw，是则直接返回，否则判断now_depth是否超过max_depth，是则直接返回，否则发送一次用户请求；之后，flow统计一次，now_depth加1；4、提供最大带宽或最大深度的修改功能；5、基于latency和flow中的数据统计并输出预设时间段内发送的用户请求以及接收到用户请求所对应的回包的平均时延。

[0045] 另外，队列管理模块主要包括以下属性：允许的最大带宽max_bw、不同优先级取满比例rate、不同优先级的请求队列、流量统计对象flow、令牌集合。队列管理模块的相关功能包括：1、初始化和内存分配；2、添加或删除请求队列；3、从请求队列中获取用户请求，具体通过不断的轮询当前指向的某个优先级的请求队列的用户请求，若该请求队列的send_pipe中存在请求，则取出一个，如此循环，直到取满或者取空；4、将获取到的用户请求输出至任务处理队列；5、统计并输出请求队列中剩余未获取的用户请求的数量。

[0046] 此外，服务器103主要包括以下属性：任务处理队列task_queue、流量控制对象flow。服务器103的相关功能：1、初始化和内存分配；2、不断轮询task_queue，对从中取出用

户请求进行任务处理(即响应用户请求);3、统计各优先级的用户请求的流量。

[0047] 下面将结合图1,以具体地实施例对本申请的技术方案以及本申请的技术方案如何解决上述技术问题进行详细说明。以下具体地实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例中不再赘述。

[0048] 如图2所示,本申请实施例提供一种用户请求处理方法,应用于上述的中间设备102或者服务器103,该方法具体包括以下步骤:

[0049] 201、确定当前处理的请求队列中的第一待处理用户请求的数量;请求队列用于存储优先级相同的用户请求。

[0050] 需要说明的是,当前处理的请求队列是指当前正处于执行状态的请求队列。针对每一请求队列,获取请求队列中的待处理用户请求的获取过程可以理解为请求队列的执行状态。例如,针对当前处理的请求队列的执行状态可以理解为获取该请求队列中第一待处理用户请求的过程。执行状态包括执行中以及执行结束。

[0051] 本申请实施例中,请求队列的数量为两个或多个,每个请求队列对应不同的优先级,按照各请求队列的优先级从高至低的顺序,采用循环访问的方式(即轮询)访问每一请求队列,以获取每一请求队列中的用户请求;例如,假设有请求队列q0和请求队列q1两个请求队列,请求队列q0的优先级高于请求队列q1,在访问完请求队列q0后,访问请求队列q1,在访问完请求队列q1后,此轮访问完成,再按照上述方式对请求队列q0和请求队列q1进行下一轮的访问,以此类推。需要说明的是,一轮访问即为一个处理周期。

[0052] 进一步的,访问每一请求队列时,在各请求队列中获取不超过针对各请求队列所设定的数量的用户请求。例如,假设有请求队列q0和请求队列q1两个请求队列,请求队列q0的优先级高于请求队列q1,请求队列q0所设定的数量为a,请求队列q1所设定的数量为b;首先获取请求队列q0中不大于a的用户请求,并在获取完成后,从请求队列q1中获取不大于b的请求队列,完成此轮的用户请求的获取;之后,按照上述方式对请求队列q0和请求队列q1进行下一轮的访问,以此类推。

[0053] 可以理解的是,第一待处理用户请求则为在本轮访问流程中,当前处理的请求队列中能够获取得到的用户请求。具体的,若当前处理的请求队列中有多个用户请求,当多个用户请求大于当前处理的请求队列所设定的数量,则该多个用户请求中的部分用户请求为第一待处理用户请求。又如,当多个用户请求不大于(即小于或等于)当前处理的请求队列所设定的数量,则该多个用户请求均为第一待处理用户请求。也就是说,第一待处理用户请求的确定具体是根据当前处理的请求队列所设定的数量以及当前处理的请求队列中实际存在的用户请求的数量综合确定。

[0054] 在一种实现方式中,步骤201的具体包括:确定当前处理的请求队列中第一待处理用户请求的数量。具体地,基于不超过针对当前处理的请求队列所设定的数量的前提下,在当前处理的请求队列中确定能够获取的用户请求,将获取到的用户请求确定为第一待处理用户请求;其中,所设定的数量用于表征在当前处理的请求队列中可获取的用户请求的最大数量。

[0055] 示例性的,假设有请求队列q0和请求队列q1两个请求队列,请求队列q0的优先级高于请求队列q1,其中,请求队列q0的所设定的数量为9,请求队列q1的所设定的数量为1,若请求队列q0为当前处理的请求队列。确定当前处理的请求队列中第一待处理用户请求的

数量,具体分为一下三种情况:

[0056] 情况一、若请求队列q0中的用户请求的数量为0,则确定请求队列q0中的第一待处理用户请求为0。

[0057] 情况二、若请求队列q0中的用户请求的数量为(0,9]中的任一整数值,则第一待处理用户请求的数量等于请求队列q0中用户请求的数量。

[0058] 情况三、若请求队列q0中的用户请求的数量大于或等于9,则第一待处理用户请求的数量则为9。

[0059] 202、若根据所述数量确定所述当前处理的请求队列满足换队条件,则从下一请求队列中获取第二待处理用户请求;当前处理的请求队列的优先级与下一请求队列的优先级不同。

[0060] 可以理解的是,若根据所述数量确定所述当前处理的请求队列不满足换队条件,则在当前处理的请求队列中继续获取第一待处理用户请求。

[0061] 在一个优选的方案中,为了保证高优先级的请求队列中的用户请求能够得到优先处理,可以按照各请求队列的优先级的高低,从高至低的在各请求队列中获取用户请求;因此,当前处理的请求队列的优先级下一请求队列的优先级为相邻的上下级关系。这样,当结合步骤201和步骤203,能够实现请求队列的优先级从高到低的顺序处理用户请求,无需为高优先级的用户请求单独预留带宽,也能保证高优先级的用户请求得到优先处理。另外,根据换队条件实现不同优先级用户请求之间的切换,使对用户请求的调度不止于高优先级,而是在满足换队条件的情况下,对低优先级的用户请求进行调度,保障了低优先级的用户请求的带宽使用。

[0062] 在一种可能的实现方式中,针对每一请求队列,该请求队列的换队条件为针对该请求队列中按照允许获取的用户请求的上限值,来获取该请求队列中的用户请求,若实际获取到的用户请求小于上限值(如0)时,或者,若实际获取到的用户请求等于上限值时,则切换至优先级低于该请求队列的下一请求队列中获取用户请求。

[0063] 进一步的,不同优先级的请求队列的换队条件的上限值可以相同,也可以不同。例如,假设当前处理的请求队列之后还有两个不同优先级的请求队列;针对当前处理的请求队列的换队条件包括当第一待处理用户请求的数量等于第一上限值时,则切换至下一请求队列中获取第二待处理用户请求;而针对下一请求队列的换队条件包括若第二待处理用户请求的数量等于第二上限值时,则切换至下下一请求队列(即下一请求队列的下一请求队列)中获取第三待处理用户请求。又如,针对当前处理的请求队列的换队条件包括当第一待处理用户请求的数量第一上限值时,则切换至下一请求队列中获取第二待处理用户请求;而针对下一请求队列的换队条件包括若第二待处理用户请求的数量同样第一上限值时,则切换至下下一请求队列中获取第三待处理用户请求。

[0064] 作为一个示例,针对每一请求队列,各请求队列中的用户请求具有子优先级,各用户请求的子优先级可以按照存储至该请求队列的时间确定,如存储至该请求队列的时间越早,子优先级越高。后续在请求队列中获取用户请求时,可按照子优先级从高至低的顺序。

[0065] 结合上述,假设请求队列q0中包含用户请求a1、用户请求b1以及用户请求c1,存储至请求队列q0的时间先后顺序依次为用户请求a1、用户请求b1以及用户请求c1,那么,用户请求a1的子优先级高于用户请求b1的子优先级,用户请求b1的子优先级高于用户请求c1的

子优先级。

[0066] 203、根据第一待处理用户请求和第二待处理用户请求更新服务器的任务处理队列;任务处理队列用于存储服务器待响应的用户请求。

[0067] 可选的,根据第一待处理用户请求和第二待处理用户请求更新服务器的任务处理队列是根据设备形态的不同主要包括以下两种更新方式:

[0068] 第一种,当用户请求处理装置为服务器或服务器中的功能模块时,用户请求处理装置将请求队列中的用户请求取出,并将该用户请求存储至任务处理队列,以实现任务处理队列的更新。

[0069] 第二种,当用户请求处理装置为除了上述服务器之外的其他设备(如路由器)或其他设备中的功能模块时,请求处理装置将请求队列中的用户请求传输至服务器,服务器将接收到的用户请求存储至任务处理队列中,以实现任务处理队列的更新。

[0070] 结合步骤202中子优先级的实施例,参照图3,假设有请求队列q0和请求队列q1两个请求队列,请求队列q0的优先级(设为第一优先级)高于请求队列q1的优先级(设为第二优先级),其中,请求队列q0的中包含用户请求a1、用户请求b1以及用户请求c1,请求队列q0中的用户请求a1的子优先级高于用户请求b1的子优先级,用户请求b1的子优先级高于用户请求c1的子优先级。请求队列q1中包含用户请求a2、用户请求b2,用户请求a2的子优先级高于用户请求b2的子优先级。若请求队列q0中的上限值为9,请求队列q1的上限值为1。可以理解的是,根据两队实际的用户请求数量以及各自的上限值可以确定,请求队列q0中的三个用户请求均可以按照子优先级从高至低依次获取,并传输至服务器的任务处理队列中;而请求队列q1中只有用户请求a2在此轮访问中被获取,用户请求b2在下一轮访问中被获取。

[0071] 本申请实施例提出的用户请求处理方法,由于现有技术需要针对高优先级的用户预留出一部分带宽来应对这些用户请求的突发情况,当这些用户请求没有突发情况时,会出现整个带宽的利用率不高的问题。因此,本申请实施例通过设置不同优先级的请求队列来存储对应优先级的用户请求,对用户请求按照优先级进行区别,并按照设定的换队条件来对各个优先级的用户请求进行流量调控。具体的,确定当前处理的请求队列中的第一待处理用户请求的数量,若根据该数量确定当前处理的请求队列满足换队条件时,则从优先级不同于当前处理的请求队列的下一请求队列中获取第二待处理用户请求,并根据第一待处理用户请求和第二待处理用户请求更新服务器中用于存储该服务器待响应的用户请求的任务处理队列。这样,通过换队条件实现不同优先级用户请求之间的切换,使对用户请求的调度不止于一个优先级的请求队列,而是在满足换队条件的情况下,对其他优先级的用户请求进行调度,保障了各优先级的用户请求的带宽使用。因此,本申请针对优先级的这种用户等级制度下,引入了多队列机制,并结合换队条件这一队列切换制度,不仅实现了高低优先级用户请求的灵活调度,并且能够合理分配带宽,提高了带宽的利用率。

[0072] 在本申请的另一实施例中,针对请求队列提前设置有该请求队列的换队条件,在获取当前处理的请求队列中用户请求的过程中,通过换队条件判定是否继续从当前处理的请求队列中获取用户请求,因此,第一待处理用户请求为第一阈值限定下,在当前处理的请求队列中可获取的用户请求;第一阈值用于表征一个处理周期中,在当前处理的请求队列中允许获取的用户请求的上限数量,处理周期用于遍历所有请求队列;换队条件,包括:数量为第一阈值限定下,在当前处理的请求队列中可获取到用户请求的最大数量。

[0073] 可选的，遍历所有请求队列可以是按照优先级顺序(从高至低或者从低至高)依次遍历，也可以按照预设的遍历条件，遍历所有请求队列，预设的遍历条件例如可以是，对所有请求队列进行编号，按照编号顺序遍历所有请求队列。例如，有8个请求队列，优先级分别为1、2、3、4、5、6、7、8。对应的编号可以是01、03、05、07、08、06、04、02，则按照01、02、03、04、05、06、07、08的编号顺序依次获取各请求队列中的用户请求。上述仅示例性的说明了一种预设的遍历条件，可在实际应用中按照实际需求随时预设各种遍历条件，本申请实施例对此不作限定。

[0074] 需要说明的是，最大数量实际是由第一阈值与当前处理的请求队列在当前的处理周期中所存储的用户请求的数量综合决定。具体的，若当前处理的请求队列在当前的处理周期中所存储的用户请求的数量小于第一阈值时，那么最大数量即为当前处理的请求队列中所述存储的用户请求的数量；若当前处理的请求队列在当前的处理周期中所存储的用户请求的数量大于或等于第一阈值时，则最大数量为第一阈值。例如，假设第一阈值为10，若当前处理的请求队列在当前的处理周期中所存储的用户请求的数量为8，则最大数量为8；若当前处理的请求队列在当前的处理周期中所存储的用户请求的数量为20，则最大数量为10.

[0075] 在实际应用中，换队条件具体设置有取空和取满两个条件，取空是针对在当前的处理周期中，当前处理的请求队列中存储的用户请求的数量小于第一阈值时，那么可以获取到的第一待处理用户请求的数量必然小于第一阈值，所以在取完当前处理的请求队列中的所有用户请求后，则从下一请求队列中获取第二待处理用户请求；取满是针对在当前的处理周期中，当前处理的请求队列中存储的用户请求的数量大于或等于第一阈值这一情况，那么当获取到的第一待处理用户请求的数量为第一阈值时，则从下一请求队列中获取第二待处理用户请求。

[0076] 其中，在一轮访问过程中，取空可以理解为在当前处理的请求队列中存在无法获取到用户请求的情况；取满可以理解为在当前处理的请求队列中能够获取数量为第一阈值的用户请求，这些获取的用户请求则为第一待处理用户请求。

[0077] 假设获取当前处理的请求队列中的用户请求为一个一个获取的情况，在当前处理的请求队列中存在无法获取到用户请求具体包括以下两种情况：

[0078] 情况一，针对当前处理的请求队列无法获取第一个用户请求，则当前处理的请求队列中无用户请求。

[0079] 情况二，针对当前处理的请求队列无法获取到第N个用户请求，则当前处理的请求队列中无用户请求，其中，0＜N＜M，M表示第一阈值。

[0080] 作为一个示例，假设当前处理的请求队列中针对第一待处理用户请求的第一阈值的设定为9，当前处理的请求队列中的用户请求的数量为5，则可知，当前处理的请求队列在获取到第5个用户请求时，当前处理的请求队列中的用户请求的数量变为0，即可理解为当前处理的请求队列未获取到第6个用户请求。

[0081] 在实际应用中，当前处理的请求队列的第一阈值并非定值，可以根据实际需求进行修改或删除。

[0082] 本实施例中，将第一待处理用户请求的数量为第一阈值限定下，在当前处理的请求队列中可获取到用户请求的最大数量作为换队条件，可以实现针对各优先级的用户请求

的灵活调用，提高带宽的利用率。

[0083] 在本申请的一个实施例中，为了确定出当前处理的请求队列中能够获取最大数量，本申请实施例根据当前处理的请求队列的优先级在所有优先级中所占的权重，确定第一阈值。

[0084] 在实际应用中，当前处理的请求队列的优先级越高，则在所有优先级中所占的权重越大，对应的第一阈值越大。

[0085] 在一种实现方式中，可以根据当前处理的请求队列的优先级所对应的用户数量在总用户数量中所占的比例，确定当前处理的请求队列的优先级在所有优先级中所占的权重。

[0086] 可选的，将当前处理的请求队列的优先级在所有优先级中所占的权重与预设的比例值的乘积，确定为第一阈值。其中，比例值可以基于能够提供的总带宽确定。例如，若当前当前处理的请求队列的优先级在所有优先级中所占的权重为9，预设的比例值为2，则第一阈值可以为2×9＝18。

[0087] 在另一种实现方式中，可以首先指定各优先级的上限数量比例，根据上限数量比例。基于实际应用需求，通过上限数量比例，确定各优先级的预设阈值。其中预设阈值包括当前请求队列的第一阈值；可以理解的是，提前确定上限数量比例后，各优先级的预设阈值时等比例设定的。例如，有两个优先级，可以设置它们的上限数量比例为9:1。那么这两个优先级的预设阈值一个为9时，另一个则为1，一个为18时，另一个则为2，以此类推。

[0088] 本实施例中，通过当前处理的请求队列的优先级在所有优先级中所占的权重，确定第一阈值，能够更加合理为各优先级的用户请求分配带宽，以便为不同优先级的用户提供更加精准的服务。

[0089] 在本申请的另一实施例中，为了保证用户请求能够顺利传输至服务器的任务处理队列，本申请实施例还引入了令牌机制，不同的优先级对应有不同的令牌集合，根据不同优先级，确定各优先级的令牌集合发放令牌的速度。具体的，根据第一待处理用户请求和第二待处理用户请求更新服务器的任务处理队列，包括：若从第一待处理用户请求对应的第一令牌集合中获取到第一待处理用户请求所需的令牌，则将第一待处理用户请求添加至任务处理队列；若从第二待处理用户请求对应的第二令牌集合中获取到第二待处理用户请求所需的令牌，则将第二待处理用户请求添加至任务处理队列。

[0090] 可以理解的是，请求队列与令牌集合之间可以是一一对应关系。

[0091] 为了有效控制每个优先级的请求队列中的用户请求对服务器的任务请求队列的访问频率，可以预先通过标识维护请求队列与令牌集合之间的映射关系，即建立请求队列的标识与令牌集合的标识之间的映射关系，例如，以键值对的方式建立请求队列的标识与令牌集合的标识之间的映射关系，使得每个请求队列存在与其对应的令牌集合。该令牌集合的标识可以是令牌集合名称或令牌集合编号等，可以由数字、字母和/或文字等组成，具体内容在此处不作限定。

[0092] 其中，令牌集合中可以存储有令牌，各优先级的令牌集合主要用于对各优先级的请求队列的访问频率进行控制，例如，针对每一优先级，可以通过令牌集合中是否存在令牌来指示是否可以允许该请求队列中的用户请求访问服务器的任务处理队列，即若该令牌集合中存在令牌，则可以为该请求队列的用户请求提供访问服务，若该令牌集合中不存在令

牌，则不为该请求队列的用户请求提供访问服务。令牌集合中的每一个令牌可以代表为该请求队列提供访问服务的次数，例如，一个令牌可以为该请求队列提供一次或多次的访问服务；或者是，令牌集合中的每一个令牌可以代表为该请求队列提供访问服务的时间段，例如一个令牌可以为该请求队列提供1秒或3秒内的访问服务；或者是，令牌集合中的每一个令牌可以代表每秒内为该请求队列提供访问服务的次数，即令牌集合中的每一个令牌可以代表每秒内允许该优先级的请求队列访问的次数(即访问频率)等。可以理解的是，令牌可以代表的权限可以根据实际需要进行灵活设置，具体内容在此不作限定。

[0093] 令牌集合的工作过程可以包括：产生令牌(即令牌投放速率)、消耗令牌和判断是否为对应优先级的请求队列提供访问服务等。其中，产生令牌可以是周期性的以预设速率向令牌集合中增加令牌，使得令牌集合中的令牌不断增多，如果令牌集合中的令牌数超过预设数量阈值，则可以丢弃多余的令牌。消耗令牌可以是在为请求队列提供访问服务时会消耗令牌集合中的令牌，以便统计该请求队列的访问频率，该访问频率可以是每秒内访问的次数，或每分钟内访问的次数等。判断是否为该请求队列提供访问服务可以是，当令牌集合中的令牌数量可以满足该请求队列中当前获取的用户请求对令牌的需求时，可以为该请求队列提供访问服务，否则不为该请求队列提供访问服务。

[0094] 以第一待处理用户请求为例，在得到携带了当前处理的请求队列的标识的第一待处理用户请求后，可以从第一待处理用户请求中提取出该请求队列的标识，并根据请求队列的标识与令牌集合的标识之间的映射关系，确定与请求队列的标识对应的令牌集合的标识。然后获取与令牌集合的标识对应的第一令牌集合，查询该第一令牌集合。当根据第一令牌集合中的令牌判断第一待处理用户请求的访问频率，例如，当第一令牌集合中存在令牌时，可以确定第一待处理用户请求的访问频率小于或等于预设阈值；当第一令牌集合中不存在令牌时，可以确定第一待处理用户请求的访问频率大于预设阈值。

[0095] 在某些实施方式中，确定与请求队列的标识对应的令牌集合的标识，并查询与令牌集合的标识对应的令牌集合的步骤之后，还可以包括：

[0096] 当第一令牌集合中存在令牌时，从第一令牌集合中获取第一待处理用户请求所需的令牌；根据获取到的令牌将第一待处理用户请求添加至任务处理队列。

[0097] 在查询第一令牌集合的过程中，当查询到第一令牌集合中存在令牌时，说明该第一待处理用户请求的访问频率小于或等于预设阈值，此时可以从第一令牌集合中获取第一待处理用户请求所需的令牌。其中，该预设阈值可以根据实际需要进行灵活设置，第一待处理用户请求所需的令牌的数量可以是一个或多个，当第一待处理用户请求所需的令牌的数量为多个时，查询到第一令牌集合中存在第一待处理用户请求所需的令牌时，从第一令牌集合中获取第一待处理用户请求所需的令牌。在获取到令牌后，将第一待处理用户请求添加至任务处理队列。

[0098] 需要说明的是，当第一令牌集合中的每一个令牌代表每周期内为当前处理的请求队列提供访问服务的次数时，在查询到第一令牌集合中存在令牌时，可以判断当前时刻是否需要更新第一令牌集合中的令牌，例如，如果当前时刻仍在当前令牌周期内，则不会更新第一令牌集合中的令牌；如果当前时刻已经进入下一个新的令牌周期内，则从第一令牌集合中扣减第一待处理用户请求所需的令牌，更新第一令牌集合中的令牌。

[0099] 本实施例中，当从待处理用户请求对应的令牌集合中获取到该待处理用户请求所

需的令牌,则将该待处理用户请求添加至任务处理队列中,从而更好的控制各优先级用户请求的处理进度,使得带宽得到更加合理的分配,进一步提高了带宽的利用率。

[0100] 在本申请的另一实施例中,为了更加合理将带宽分配给不同优先级的用户请求,可以基于各请求队列的优先级,确定对应优先级的令牌集合所产生令牌的速度,因此本申请实施例所提供的方式还能够根据各请求队列的优先级,确定各请求队列对应的令牌集合的令牌投放速率。

[0101] 具体的,针对每一请求队列,确定该请求队列的优先级在所有请求队列的优先级中所占的权重,并基于该权重确定该请求队列所对应的令牌集合的令牌投放速率。

[0102] 作为一个示例,参照图4,假设有请求队列q0和请求队列q1两个请求队列,请求队列q0的优先级高于请求队列q1,请求队列q0对应的令牌集合为B0,请求队列q1对应的令牌集合为B1。假设请求队列q0为当前处理的请求队列,请求队列q0的优先级的权重为9,请求队列q1的优先级的权重为1,则可以得到令牌投放速率其中一种假设的情况为:第一令牌集合中令牌投放速率为每秒增加1个,第一令牌集合中令牌投放速率为每9秒增加1个。也就是说,当第一令牌集合中有9个令牌时,第二令牌集合中有1个令牌。

[0103] 本实施例中,通过各请求队列的优先级,确定各请求队列对应的令牌集合的令牌投放速率,能够更好的控制各优先级用户请求的处理进度,使得带宽得到更加合理的分配,进一步提高了带宽的利用率。

[0104] 在本申请的另一实施例中,参照图5,本申请实施例所提供的用户请求处理方法还包括:

[0105] 501、接收用户端发送的用户请求,确定用户请求的优先级。

[0106] 可选的,用户请求为用户端发送的用于请求某项业务的请求信息。业务是服务器的(如指定业务服务器、接入层服务器或后台服务器)为用户端提供的服务。指定业务服务器提供区别于默认业务的服务,接入层服务器提供数据分发及默认业务的服务,后台服务器提供业务所需的数据服务。默认业务可以为对普通用户可见的业务,区别于默认业务的服务可以是为开发团队提供的业务,一般对普通用户不可见。

[0107] 具体的,根据用户请求确定用户请求的优先级。

[0108] 在一种实现方式中,在接收用户请求后,从接收到的用户请求中提取用户信息,基于用户信息确定用户请求的优先级。例如,可以在用户请求中提取用户标识,通过用户标识确定用户请求的优先级。其中,用户标识可以是用户账号或者用户账号所属的企业账号等。

[0109] 在另一种实现方式中,用户请求中携带用户请求的优先级。进一步的,用户请求的优先级根据用户请求中用于指示优先级的字段确定。具体的,用户请求的数据包的预设字段中包含表征用户请求的优先级的字段。或者,所述第一报文的丢弃优先级根据所述第一报文的协议的丢弃敏感度确定。

[0110] 举例来说,用户请求中用于指示优先级的字段为MPLS标签中的试验比特(experimental bits,EXP)字段、虚拟局域网(virtual local area network,VLAN)标签中的优先级码点(priority code point,PCP)字段或者网际互连协议(internet protocol,IP)头中的差分服务代码点(differentiated services code point,DSCP)字段。

[0111] 502、根据用户请求的优先级将用户请求添加至优先级的请求队列。

[0112] 可选的,基于用户请求的优先级,确定请求队列的标识;并根据请求队列的标识,

将用户请求添加至该标识对应的请求队列中。

[0113] 在一种实现方式中，接收用户端发送的第一待处理用户请求，确定第一待处理用户请求的优先级，根据第一待处理用户请求的优先级将用户请求添加至当前处理的请求队列中。

[0114] 在另一种实现方式中，接收用户端发送的第二待处理用户请求，确定第二待处理用户请求的优先级，根据第二待处理用户请求的优先级将用户请求添加至下一请求队列中。

[0115] 需要说明的是，可参考上述确定用户请求的优先级的实施例，确定第一待处理用户请求或者第二待处理用户请求的优先级，此处不再赘述。

[0116] 本实施例中，通过确定接收到的用户请求的优先级，将用户请求添加至该优先级的请求队列中，以便对用户请求按照优先级进行区别，从而按照设定的换队条件来对各个优先级的用户请求进行流量调控。

[0117] 在本申请的另一实施例中，可以通过用户请求的源端口，确定用户请求的优先级，以便将其存储至对应的请求队列中，根据用户请求确定用户请求的优先级，包括：根据用户请求的源端口，确定用户请求的优先级。

[0118] 在一些可行的实施方式中，上述用户请求的源端口可以理解为发出该用户请求的用户端的端口号。这里的用户请求可以是第一待处理用户请求，也可以是第二待处理用户请求。

[0119] 可选的，本申请实施例不限于上述的基于用户请求的源端口，还可根据用户请求的源地址，确定用户请求的优先级。其中，源地址可以包括但不限于以下的任一项：源IP地址、源媒体访问控制(media access control，MAC)地址。

[0120] 进一步的，还可以基于用户请求的互联网安全协议(internet protocol security，IPSEC)、用于提供安全外壳(secure shell，SSH)协议服务的端口(TCP22端口)、超文本传输协议(hypertext transfer protocol，HTTP)服务的端口(TCP80端口)以及超文本传输安全协议(hyper text transfer protocol over secure socket layer，HTTPS)服务的端口(TCP443端口)中的任一项，来确定用户请求的优先级。

[0121] 本实施例中，根据用户请求的源端口来确定用户请求的优先级，无需通过在用户请求中额外添加优先级的相关字段，也能够识别出用户请求的优先级，从而减少对用户请求的数据包的解析工作。

[0122] 在本申请的另一实施例中，根据用户请求的优先级将用户请求添加至请求队列，包括：基于用户请求的优先级，确定请求队列的标识；根据请求队列的标识，将用户请求添加至请求队列所对应的数据链表中。

[0123] 可以理解的是，同一请求队列的数据链表用于存储同一优先级的用户请求，而针对请求队列的数据链表中的用户请求，可以按照各用户请求的子优先级由高至低存储于数据链表中。

[0124] 具体的，数据链表可以为单向链表、双向链表以及循环链表中的任一种链表，具体可根据实际应用需求在上述的单向链表、双向链表以及循环链表中选择，本申请实施例对此不作限定；数据链表是通过指针来指示存储于数据链表中的用户请求的获取顺序，指针的方向用于表征用户请求获取的顺序，例如，参照图6，简单展示了用户请求1、用户请求2、

用户请求3、用户请求4存储于单向链表的结构示意图中，由图6中指针的方向可知，用户请求的获取顺序为用户请求1、用户请求2、用户请求3、用户请求4，则可以理解为子优先级从高至低为用户请求1、用户请求2、用户请求3、用户请求4。需要说明的是，图6中NULL表示数据链表的结尾。

[0125] 作为一个示例，当从数据链表中获取一个用户请求，则将数据链表中关于该用户请求的数据进行删除；或者，当从数据链表中完成一轮的用户请求获取后，将数据链表中关于已获取的用户请求的数据进行删除。

[0126] 在另一种实现方式中，根据用户请求的优先级将用户请求添加至请求队列，还可以包括：基于用户请求的优先级，确定请求队列的标识；将请求队列的标识将用户请求添加至请求队列的数组中。

[0127] 本实施例中，根据用户请求的优先级将用户请求添加至请求队列所对应数据链表中，由于数据链表能够根据实际存储的用户请求的变化情况，随时调节自身的大小，因此可以极大的提高内存的利用率。

[0128] 为了体现本申请的实际效果，参照图7，以云服务为例，本申请实施例可以应对不同优先级的用户请求的情况所能够达到的处理效果。其中，以高优先级和低优先级这两种优先级举例，而在实际场景可以支持多个优先级。如图7中的(a)展示了针对云服务所能提供的带宽，需要将所有用户的总体流量控制在一个这一带宽之内，可能会产生如下三种情况并达到相应的效果：情况一，如图7中的(b)展示的，当高优先级的用户请求与低优先的用户请求不能占满云服务所能提供的带宽的占用情况；情况二，如图7中的(c)展示的，当高优先级的用户请求与低优先的用户请求刚好占满云服务所能提供的带宽的占用情况；情况三，如图7中的(d)展示的，当高优先级的用户请求出现突发情况，高优先级的用户请求与低优先的用户请求在云服务所能提供的带宽的占用情况。由此可知，本申请提出的用户请求处理方法能够通过请求队列的优先级从高到低的顺序处理用户请求，无需为高优先级的用户请求单独预留带宽，也能保证高优先级的用户请求得到优先处理。另外，根据换队条件实现不同优先级用户请求之间的切换，使对用户请求的调度不止于一个优先级，而是在满足换队条件的情况下，对其他优先级的用户请求进行调度。保障了各优先级的用户请求的带宽使用。

[0129] 综上，本申请实施例针对高低优先级的这种用户等级制度下，引入了多队列机制，并结合换队条件这一队列切换制度，不仅实现了高低优先级用户请求的灵活调度，并且能够合理分配带宽，提高了带宽的利用率。

[0130] 应当注意，尽管在附图中以特定顺序描述了本申请方法的操作，但是，这并非要求或者暗示必须按照该特定顺序来执行这些操作，或是必须执行全部所示的操作才能实现期望的结果。

[0131] 图8为本申请一个实施例的用户请求处理装置80的方框示意图，该用户请求处理装置80可以是图1所示的中间设备102或者服务器103，也可以是设置于中间设备102或者服务器103中用户实现上述用户请求处理方法的硬件模块(如芯片等)或者软件程序。当用户请求处理装置80为硬件模块时，可以是上述的

[0132] 如图8所示，用户请求处理装置80包括：确定单元801、处理单元802、更新单元803、接收单元804。其中，

[0133] 确定单元801，用于确定当前处理的请求队列中的第一待处理用户请求的数量；请求队列用于存储优先级相同的用户请求。

[0134] 处理单元802，用于若根据所述数量确定所述当前处理的请求队列满足换队条件，则从下一请求队列中获取第二待处理用户请求；当前处理的请求队列的优先级与下一请求队列的优先级不同。

[0135] 更新单元803，用于根据第一待处理用户请求和第二待处理用户请求更新服务器的任务处理队列；任务处理队列用于存储服务器待响应的用户请求。

[0136] 在本申请的另一实施例中，第一待处理用户请求为第一阈值限定下，在当前处理的请求队列中可获取的用户请求；第一阈值用于表征一个处理周期中，在当前处理的请求队列中允许获取的用户请求的上限数量，处理周期用于遍历所有请求队列；换队条件，包括：数量为第一阈值限定下，在当前处理的请求队列中可获取到用户请求的最大数量。

[0137] 在本申请的另一实施例中，处理单元802，具体用于根据当前处理的请求队列的优先级在所有优先级中所占的权重，确定第一阈值。

[0138] 在本申请的另一实施例中，更新单元803，具体用于，

[0139] 若从第一待处理用户请求对应的第一令牌集合中获取到第一待处理用户请求所需的令牌，则将第一待处理用户请求添加至任务处理队列。

[0140] 若从第二待处理用户请求对应的第二令牌集合中获取到第二待处理用户请求所需的令牌，则将第二待处理用户请求添加至任务处理队列。

[0141] 在本申请的另一实施例中，更新单元803，具体用于根据各请求队列的优先级，确定各请求队列对应的令牌集合的令牌投放速率。

[0142] 在本申请的另一实施例中，接收单元804，用于接收用户端发送的用户请求。

[0143] 处理单元802，用于确定用户请求的优先级，根据用户请求的优先级将第一待处理用户请求添加至优先级的请求队列。

[0144] 在本申请的另一实施例中，处理单元802，具体用于根据用户请求的源端口，确定用户请求的优先级。

[0145] 在本申请的另一实施例中，处理单元802，具体用于基于用户请求的优先级，确定请求队列的标识；根据请求队列的标识，将用户请求添加至请求队列所对应的数据链表中。

[0146] 本申请实施例提出的用户请求处理装置，由于现有技术需要针对高优先级的用户预留出一部分带宽来应对这些用户请求的突发情况，当这些用户请求没有突发情况时，会出现整个带宽的利用率不高的问题。因此，本申请实施例通过设置不同优先级的请求队列来存储对应优先级的用户请求，对用户请求按照优先级进行区别，并按照设定的换队条件来对各个优先级的用户请求进行流量调控。具体的，确定当前处理的请求队列中的第一待处理用户请求的数量，若根据该数量确定当前处理的请求队列满足换队条件时，则从优先级不同于当前处理的请求队列的下一请求队列中获取第二待处理用户请求，并根据第一待处理用户请求和第二待处理用户请求更新服务器中用于存储该服务器待响应的用户请求的任务处理队列。这样，通过换队条件实现不同优先级用户请求之间的切换，使对用户请求的调度不止于一个优先级的请求队列，而是在满足换队条件的情况下，对其他优先级的用户请求进行调度，保障了各优先级的用户请求的带宽使用。因此，本申请针对优先级的这种用户等级制度下，引入了多队列机制，并结合换队条件这一队列切换制度，不仅实现了高低

优先级用户请求的灵活调度，并且能够合理分配带宽，提高了带宽的利用率。

[0147] 应当理解，用户请求处理装置中记载的诸单元与参考图7描述的方法中的各个步骤相对应。由此，上文针对方法描述的操作和特征同样适用于用户请求处理装置及其中包含的单元，在此不再赘述。用户请求处理装置可以预先实现在计算机设备的浏览器或其他安全应用中，也可以通过下载等方式而加载到计算机设备的浏览器或其安全应用中。用户请求处理装置中的相应单元可以与计算机设备中的单元相互配合以实现本申请实施例的方案。

[0148] 在上文详细描述中提及的若干模块或者单元，这种划分并非强制性的。实际上，根据本申请的实施方式，上文描述的两个或更多模块或者单元的特征和功能可以在一个模块或者单元中具体化。反之，上文描述的一个模块或者单元的特征和功能可以进一步划分为由多个模块或者单元来具体化。

[0149] 需要说明的是，本申请实施例的用户请求处理装置中未披露的细节，请参照本申请上述实施例中所披露的细节，这里不再赘述。

[0150] 下面参考图9，图9示出了适于用来实现本申请实施例的计算机设备的结构示意图，如图9所示，计算机系统900包括中央处理单元(CPU)901，其可以根据存储在只读存储器(ROM)902中的程序或者从存储部分909加载到随机访问存储器(RAM)903中的程序而执行各种适当的动作和处理。在RAM903中，还存储有系统的操作指令所需的各种程序和数据。CPU901、ROM902以及RAM903通过总线904彼此相连。输入/输出(I/O)接口905也连接至总线904。

[0151] 以下部件连接至I/O接口905；包括键盘、鼠标等的输入部分906；包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分907；包括硬盘等的存储部分909；以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分909。通信部分909经由诸如因特网的网络执行通信处理。驱动器910也根据需要连接至I/O接口905。可拆卸介质911，诸如磁盘、光盘、磁光盘、半导体存储器等等，根据需要安装在驱动器910上，以便于从其上读出的计算机程序根据需要被安装入存储部分909。

[0152] 特别地，根据本申请的实施例，上文参考流程图图2描述的过程可以被实现为计算机软件程序。例如，本申请的实施例包括一种计算机程序产品，其包括承载在计算机可读介质上的计算机程序，该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中，该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中，该计算机程序可以通过通信部分909从网络上被下载和安装，和/或从可拆卸介质911被安装。在该计算机程序被中央处理单元(CPU)901执行时，执行本申请的系统中限定的上述功能。

[0153] 需要说明的是，本申请所示的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或者是上述两者的任意组合。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地例子可以包括但不限于：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(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。在本申请中，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。而在本申请中，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为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。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。

[0154] 附图中的流程图和框图，图示了按照本申请各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作指令。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分，前述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所标注的功能也可以不同于附图中所标注的顺序发生。例如，两个连接表示的方框实际上可以基本并行地执行，他们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或操作指令的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0155] 描述于本申请实施例中所涉及到的单元或模块可以通过软件的方式实现，也可以通过硬件的方式来实现。所描述的单元或模块也可以设置在处理器中，例如，可以描述为：一种处理器包括违规人物检测单元、多模态检测单元以及识别单元。其中，这些单元或模块的名称在某种情况下并不构成对该单元或模块本身的限定。

[0156] 作为另一方面，本申请还提供了一种计算机可读存储介质，该计算机可读存储介质可以是上述实施例中描述的计算机设备中所包含的，也可以是单独存在，而未装配入该计算机设备中的。上述计算机可读存储介质存储有一个或多个程序，当上述程序被一个或者一个以上的处理器用来执行本申请所述的用户请求处理方法。例如，可以执行图2所示的用户请求处理方法的各个步骤。

[0157] 本申请实施例提供一种计算机程序产品，该计算机程序产品包括指令，当该指令被运行时，使得如本申请实施例描述的方法被执行。例如，可以执行图2所示的用户请求处理方法的各个步骤。

[0158] 以上描述仅为本申请的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解，本申请中所涉及的公开范围，并不限于上述技术特征的特定组合而成的技术方案，同时也应涵盖在不脱离前述公开构思的情况下，由上述技术特征或其等同特征进行任意组合而形成的其他技术方案。例如上述特征与本申请中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。

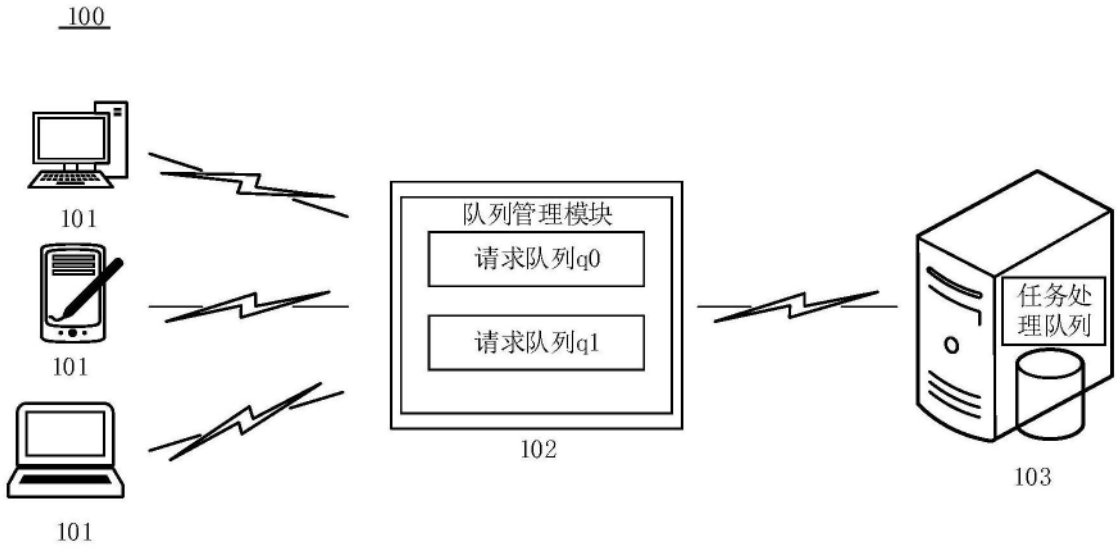

100
101
101
101
队列管理模块
请求队列q0
请求队列q1
102
任务处理队列
103

图1

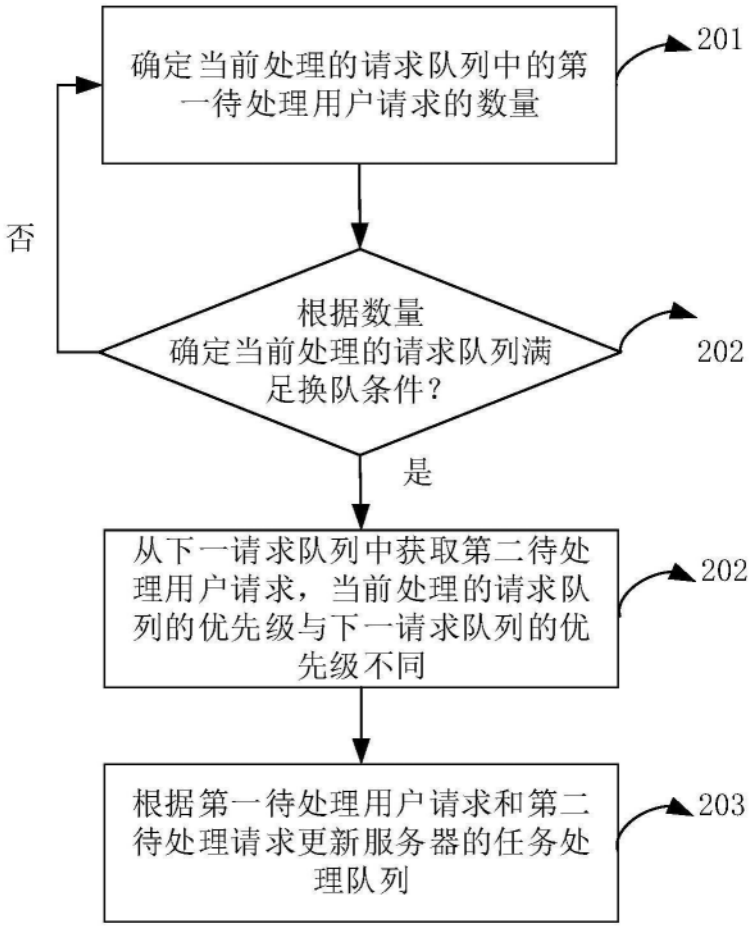

确定当前处理的请求队列中的第一待处理用户请求的数量
201
根据数量确定当前处理的请求队列满足换队条件？
202
否
是
从下一请求队列中获取第二待处理用户请求，当前处理的请求队列的优先级与下一请求队列的优先级不同
202
根据第一待处理用户请求和第二待处理请求更新服务器的任务处理队列
203

图2

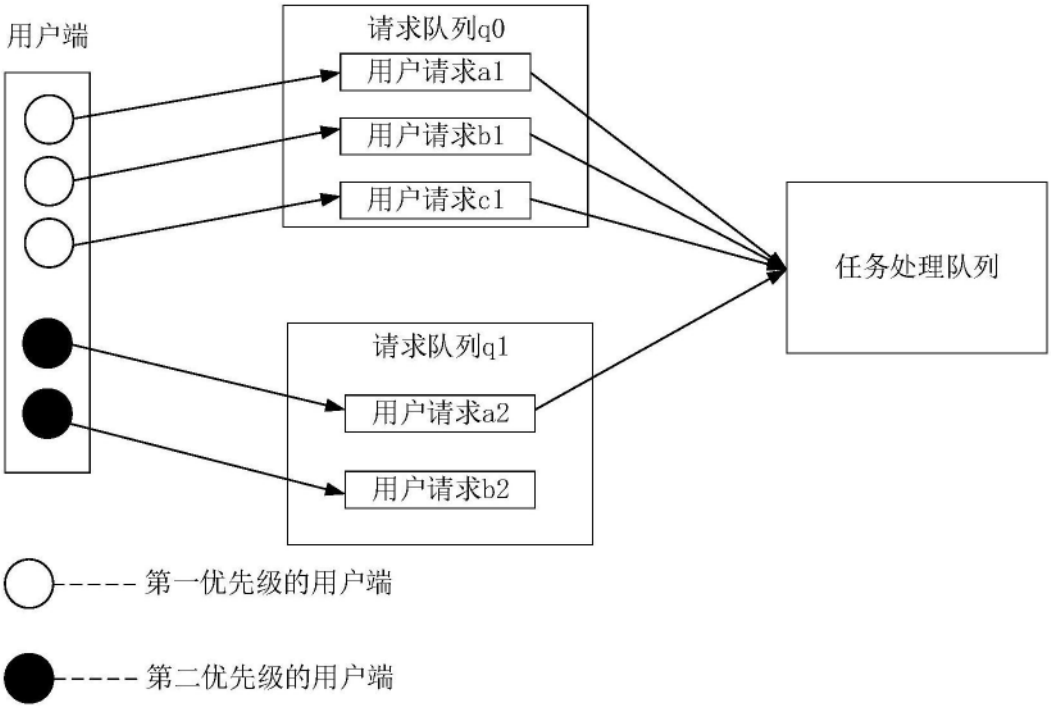

用户端
请求队列q0
用户请求a1
用户请求b1
用户请求c1
任务处理队列
请求队列q1
用户请求a2
用户请求b2
----- 第一优先级的用户端
----- 第二优先级的用户端

图3

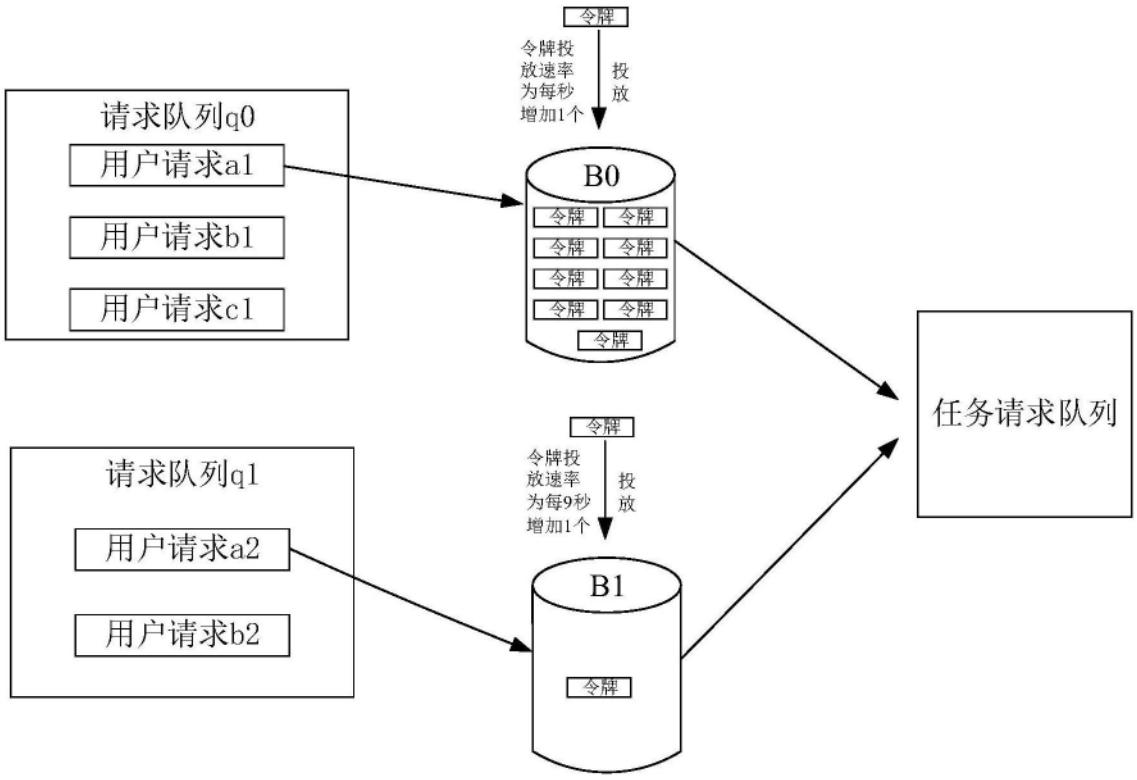

令牌
令牌投放速率为每秒增加1个
投放
请求队列q0
用户请求a1
用户请求b1
用户请求c1
B0
令牌
任务请求队列
请求队列q1
用户请求a2
用户请求b2
令牌投放速率为每9秒增加1个
B1

图4

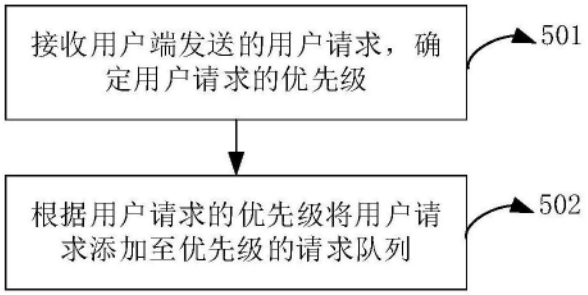

接收用户端发送的用户请求，确定用户请求的优先级
501
根据用户请求的优先级将用户请求添加至优先级的请求队列
502

图5

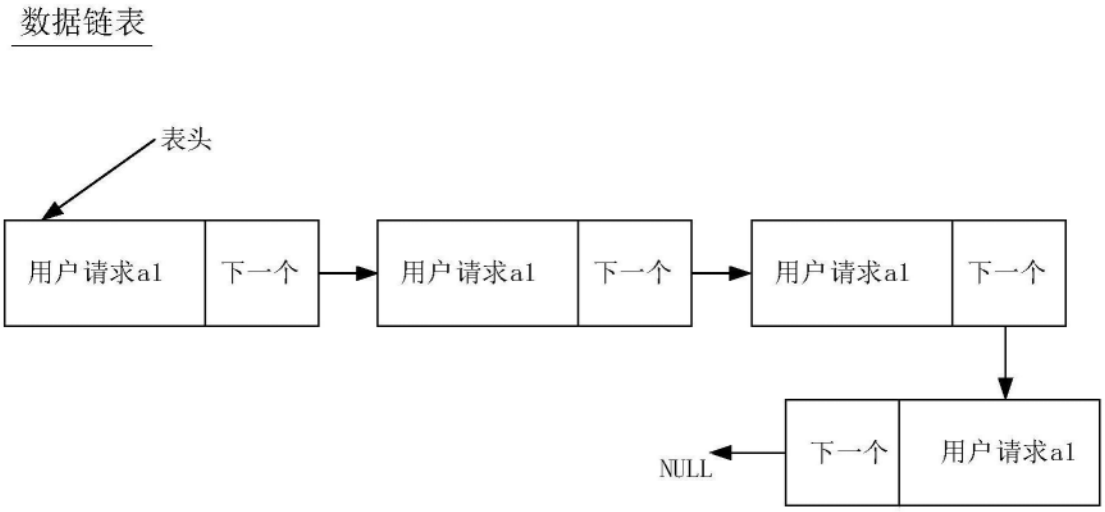

数据链表
表头
用户请求a1
下一个
用户请求a1
下一个
用户请求a1
下一个
下一个
用户请求a1
NULL

图6

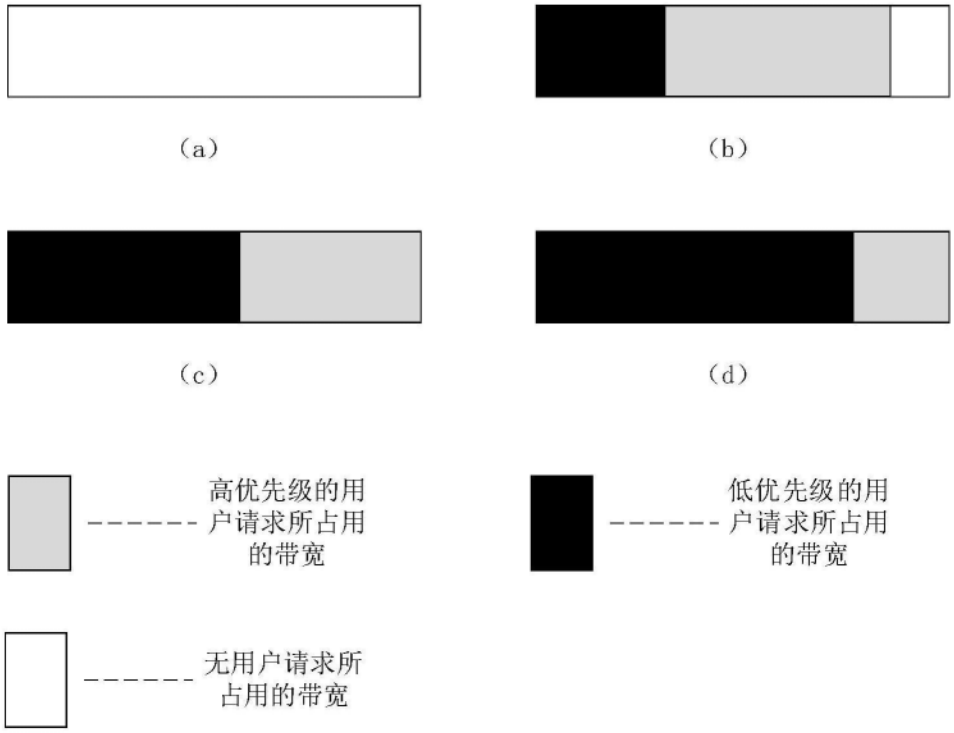

(a)
(b)
(c)
(d)
高优先级的用户请求所占用的带宽
低优先级的用户请求所占用的带宽
无用户请求所占用的带宽

图7

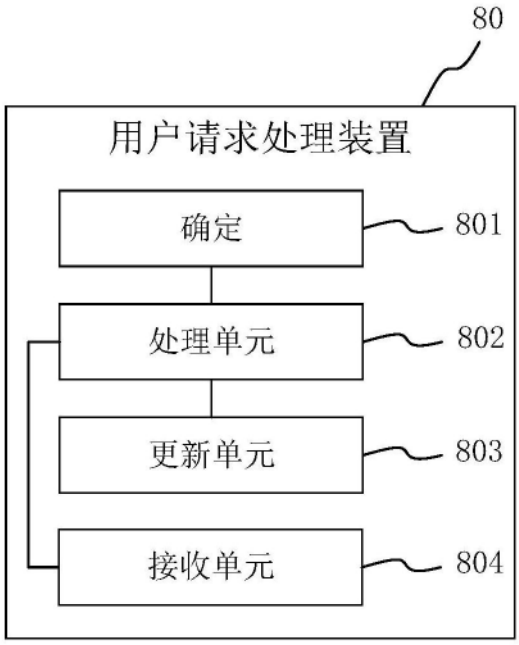

80
用户请求处理装置
确定
801
处理单元
802
更新单元
803
接收单元
804

图8

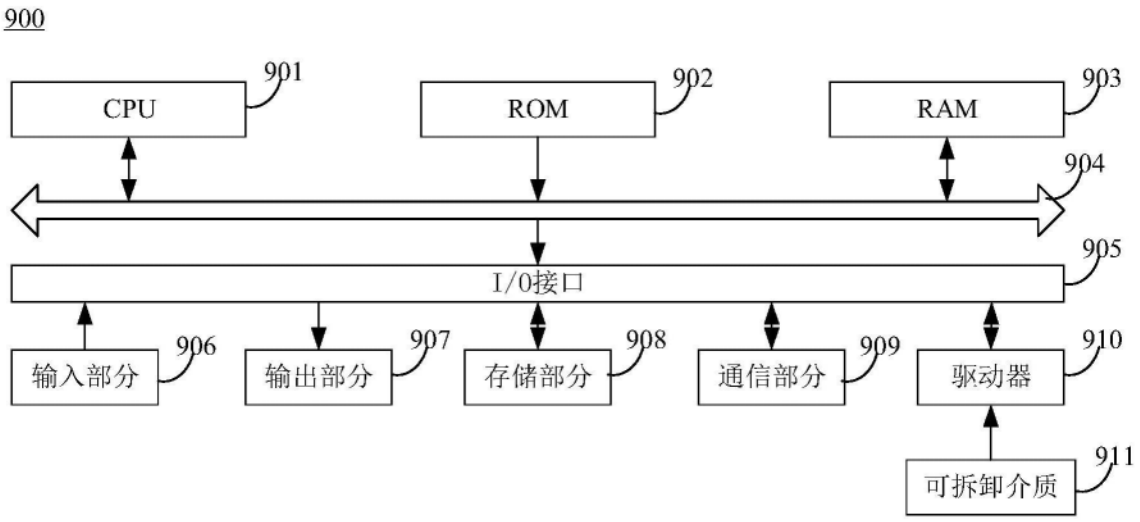

900
CPU
901
ROM
902
RAM
903
904
I/O接口
905
输入部分
906
输出部分
907
存储部分
908
通信部分
909
驱动器
910
可拆卸介质
911

图9