



(12)发明专利申请

(10)申请公布号 CN 109784065 A

(43)申请公布日 2019.05.21

(21)申请号 201811474324.3

(22)申请日 2018.12.04

(71)申请人 北京达佳互联信息技术有限公司

地址 100084 北京市海淀区中关村东路1号
院8号楼20层B2201

(72)发明人 刘硕 李明 马彦兵

(74)专利代理机构 北京润泽恒知识产权代理有
限公司 11319

代理人 莎日娜

(51)Int.Cl.

G06F 21/60(2013.01)

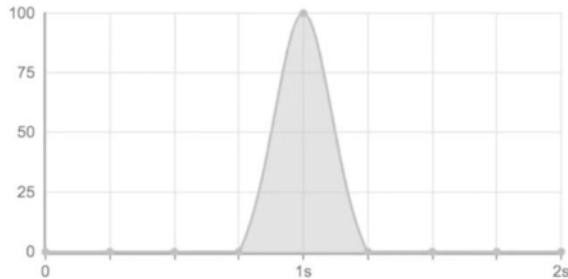
权利要求书2页 说明书11页 附图3页

(54)发明名称

访问控制方法、装置、服务器及存储介质

(57)摘要

本公开是关于一种访问控制方法、装置、服务器及存储介质,以解决相关技术不能准确、完善地进行用户访问控制的问题。其中方法包括:接收客户端发送的接口访问请求,接口访问请求包括所述客户端的标识;针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数;每个控制类对应一个定时时间及一个次数阈值;在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。本公开能够通过更多时间指标,更细粒度的控制,实现一个非常平滑的控制,能够更加完善控制方式,提高控制的准确性,提升服务器的性能。



1. 一种访问控制方法,其特征在于,包括:

接收客户端发送的接口访问请求,所述接口访问请求包括所述客户端的标识;

针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数;每个控制类对应一个定时时间及一个次数阈值;

在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。

2. 根据权利要求1所述的访问控制方法,其特征在于,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;

所述逐个确定在所述控制类中所述客户端的标识对应的当前访问次数的步骤,包括:

针对第一个级别的控制类,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;

针对从第二个级别开始的控制类,依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数。

3. 根据权利要求2所述的访问控制方法,其特征在于,所述依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数的步骤,包括:

如果上一个级别的控制类中所述当前访问次数未超过所述次数阈值,则查找出所述控制类中是否记录有所述客户端的标识对应的访问次数;

如果所述控制类中记录有所述客户端的标识对应的访问次数,则对所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;

如果所述控制类中未记录所述客户端的标识对应的访问次数,则确定所述客户端的标识对应的当前访问次数为0;

如果上一个级别的控制类中所述当前访问次数超过所述次数阈值,则对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数。

4. 根据权利要求1所述的访问控制方法,其特征在于,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;

每个级别的控制类对应的次数阈值小于一个乘积;所述乘积为所述控制类对应的定时时间与上一个级别的控制类对应的定时时间的商,与所述上一个级别的控制类对应的次数阈值的乘积。

5. 根据权利要求1所述的访问控制方法,其特征在于,所述方法还包括:

针对每个控制类,在到达所述控制类对应的定时时间时,将所述控制类中记录的全部访问次数清空。

6. 一种访问控制装置,其特征在于,包括:

接收单元,被配置为接收客户端发送的接口访问请求,所述接口访问请求包括所述客户端的标识;

确定单元,被配置为针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数;每个控制类对应一个定时时间及一个次数阈值;

控制单元,被配置为在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。

7. 根据权利要求6所述的访问控制装置,其特征在于,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;所述确定单元包括:

第一确定模块,被配置为针对第一个级别的控制类,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;

第二确定模块,被配置为针对从第二个级别开始的控制类,依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数。

8. 根据权利要求7所述的访问控制装置,其特征在于,所述第二确定模块包括:

查找子模块,被配置为在上一个级别的控制类中所述当前访问次数未超过所述次数阈值时,查找在所述控制类中是否记录有所述客户端的标识对应的访问次数;

第三确定子模块,被配置为在所述控制类中记录有所述客户端的标识对应的访问次数时,对所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;在所述控制类中未记录所述客户端的标识对应的访问次数时,确定所述客户端的标识对应的当前访问次数为0;

第四确定子模块,被配置为在上一个级别的控制类中所述当前访问次数超过所述次数阈值时,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数。

9. 一种服务器,其特征在于,包括:

处理器;

用于存储处理器可执行指令的存储器;

其中,所述处理器被配置为执行如权利要求1-5任一项所述的访问控制方法。

10. 一种非临时性计算机可读存储介质,当所述存储介质中的指令由服务器的处理器执行时,使得服务器能够执行如权利要求1-5任一项所述的访问控制方法。

访问控制方法、装置、服务器及存储介质

技术领域

[0001] 本公开涉及计算机技术领域,尤其涉及一种访问控制方法、装置、服务器及存储介质。

背景技术

[0002] 随着网络技术的不断发展,网络访问用户的数量不断增加。当大量的用户对服务器进行访问时,在高并发高流量的情况下,将会出现访问阻塞、访问出错等问题,并且会导致服务器的内存不足,甚至崩溃。

[0003] 针对上述问题,可以采用流量控制算法进行访问频率控制。在相关技术中,流量控制算法可以提供在某个时间范围内对某种资源的访问总次数的控制。比如,假设流量控制规则为限制每秒钟不超过100次接口请求,第一个1s时间窗口内,100次接口请求都集中在最后的10ms内,在第二个1s的时间窗口内,100次接口请求都集中在最开始的10ms内。虽然在两个时间窗口内都符合限流要求,但在两个时间窗口临界的20ms内会集中有200次接口请求,如果不进行限流,则集中在这20ms内的200次请求就可能导致服务器崩溃。如图1所示,在1s的前后临界位置出现了突发流量,而并未对该情况进行控制。

[0004] 因此,相关技术中的流量控制算法对于突发流量的问题无法进行处理,因此不能准确、完善地进行用户访问控制。

发明内容

[0005] 为克服相关技术中存在的问题,本公开提供一种访问控制方法、装置、服务器及存储介质,以解决相关技术不能准确、完善地进行用户访问控制的问题。

[0006] 根据本公开实施例的第一方面,提供一种访问控制方法,包括:接收客户端发送的接口访问请求,所述接口访问请求包括所述客户端的标识;针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数;每个控制类对应一个定时时间及一个次数阈值;在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。

[0007] 可选地,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;所述逐个确定在所述控制类中所述客户端的标识对应的当前访问次数的步骤,包括:针对第一个级别的控制类,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;针对从第二个级别开始的控制类,依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数。

[0008] 可选地,所述依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数的步骤,包括:如果上一个级别的控制类中所述当前访问次数未超过所述次数阈值,则查找在所述控制类中是否记录有所述客户端的标识对应的访问次数;如果所述控制类中记录有所述客户端的标识对应的

访问次数,则对所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;如果所述控制类中未记录所述客户端的标识对应的访问次数,则确定所述客户端的标识对应的当前访问次数为0;如果上一个级别的控制类中所述当前访问次数超过所述次数阈值,则对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数。

[0009] 可选地,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;每个级别的控制类对应的次数阈值小于一个乘积;所述乘积为所述控制类对应的定时时间与上一个级别的控制类对应的定时时间的商,与所述上一个级别的控制类对应的次数阈值的乘积。

[0010] 可选地,所述方法还包括:针对每个控制类,在到达所述控制类对应的定时时间时,将所述控制类中记录的全部访问次数清空。

[0011] 根据本公开实施例的第二方面,提供一种访问控制装置,包括:接收单元,被配置为接收客户端发送的接口访问请求,所述接口访问请求包括所述客户端的标识;确定单元,被配置为针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数;每个控制类对应一个定时时间及一个次数阈值;控制单元,被配置为在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。

[0012] 可选地,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;所述确定单元包括:第一确定模块,被配置为针对第一个级别的控制类,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;第二确定模块,被配置为针对从第二个级别开始的控制类,依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数。

[0013] 可选地,所述第二确定模块包括:查找子模块,被配置为在上一个级别的控制类中所述当前访问次数未超过所述次数阈值时,查找在所述控制类中是否记录有所述客户端的标识对应的访问次数;第三确定子模块,被配置为在所述控制类中记录有所述客户端的标识对应的访问次数时,对所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;在所述控制类中未记录所述客户端的标识对应的访问次数时,确定所述客户端的标识对应的当前访问次数为0;第四确定子模块,被配置为在上一个级别的控制类中所述当前访问次数超过所述次数阈值时,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数。

[0014] 可选地,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;每个级别的控制类对应的次数阈值小于一个乘积;所述乘积为所述控制类对应的定时时间与上一个级别的控制类对应的定时时间的商,与所述上一个级别的控制类对应的次数阈值的乘积。

[0015] 可选地,所述装置还包括:定时单元,被配置为针对每个控制类,在到达所述控制类对应的定时时间时,将所述控制类中记录的全部访问次数清空。

[0016] 根据本公开实施例的第三方面,提供一种服务器,包括:处理器;用于存储处理器可执行指令的存储器;其中,所述处理器被配置为执行上述任一种访问控制方法。

[0017] 根据本公开实施例的第四方面,提供一种非临时性计算机可读存储介质,当所述存储介质中的指令由服务器的处理器执行时,使得服务器能够执行上述任一种访问控制方法。

[0018] 根据本公开实施例的第五方面,提供一种计算机程序产品,当所述计算机程序产品中的指令由服务器的处理器执行时,使得服务器能够执行上述任一种访问控制方法。

[0019] 本公开的实施例提供的技术方案可以包括以下有益效果:

[0020] 本公开的实施例中,预先设置多个控制类,每个控制类对应一个定时时间及一个次数阈值。在进行访问控制时,服务器接收客户端发送的接口访问请求,接口访问请求包括所述客户端的标识;针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数;在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。由此可知,本公开的实施例通过多个控制类,能够在不同的定时时间内,对同一客户端的访问次数进行不同次数阈值下的联合控制,因此能够解决在某个时间段内出现突发流量而无法控制的问题,通过更多时间指标,更细粒度的控制,可以实现一个非常平滑的控制,能够更加完善控制方式,提高控制的准确性,提升服务器的性能。

[0021] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本公开。

附图说明

[0022] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本发明的实施例,并与说明书一起用于解释本发明的原理。

[0023] 图1是一种相关技术中的访问控制的效果示意图。

[0024] 图2是根据一示例性实施例示出的一种访问控制方法的流程图。

[0025] 图3是根据一示例性实施例示出的一种访问控制的效果示意图。

[0026] 图4是根据一示例性实施例示出的一种访问控制装置的框图。

[0027] 图5是根据一示例性实施例示出的一种用于访问控制的装置的框图。

具体实施方式

[0028] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本发明相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本发明的一些方面相一致的装置和方法的例子。

[0029] 图2是根据一示例性实施例示出的一种访问控制方法的流程图,如图2所示,访问控制方法用于服务器中,包括以下步骤。

[0030] 在步骤S11中,接收客户端发送的接口访问请求,接口访问请求包括所述客户端的标识。

[0031] 客户端可以通过服务器上的接口访问服务器的资源。服务器中可以包括多个接口,客户端通过访问不同的接口可以访问不同的资源。其中,客户端可以为移动终端、PC(个人电脑)等硬件客户端,也可以为安装于硬件终端上的软件客户端,如APP(应用程序)等。

[0032] 当某个客户端想要访问服务器的某一接口时,该客户端可以向服务器发送接口访问请求。接口访问请求中可以包括发送该接口访问请求的客户端的标识、访问类型等信息。其中,标识可以为客户端的IP(Internet Protocol)地址等能够唯一标识该客户端的信息,访问类型可以为心跳消息、数据请求等类型。服务器可以接收该客户端发送的上述接口访问请求。

[0033] 在步骤S12中,针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数。

[0034] 本公开实施例中,可以预先定义多个控制类(IpRateLimiter),每个控制类对应一个定时时间(period)以及一个次数阈值(max)。其中,定时时间可以为1s、1min、1hour,等等,次数阈值可以根据实际情况设置任意适用的数值。

[0035] 在实现上,可以为每个控制类定义一个定时器。定时器用于控制定时时间,也即定时器的定时时间即为所在控制类对应的定时时间。每个控制类记录一个访问数据,该访问数据的结构可以为一个Map,用于记录每个客户端对应的访问次数。Map用于保存具有映射关系的数据,因此Map中保存着两种值,一种用于保存key,另一种用来保存value。其中, key可以对应客户端的标识,value可以对应访问次数。

[0036] 服务器在第一次接收到一个接口访问请求时,可以针对每个控制类,均按照各自对应的定时时间开始定时。也即,服务器在第一次接收到一个接口访问请求时,每个控制类的定时器均开始工作。

[0037] 针对每个控制类,在到达该控制类对应的定时时间时,将该控制类中记录的全部访问数据(包括客户端的标识和访问次数)清空,并按照该控制类对应的定时时间重新开始定时。也即,每个控制类对应的定时器在到达各自的定时时间时,各自按照对应的定时时间开始进行新一轮的定时。

[0038] 控制类的数据结构的伪代码表示如下:

```
private int max;  
private long period;  
private ConcurrentHashMap<Integer, AtomicInteger> counts;
```

```
public IpRateLimiter(  
    int countMax, long periodInMillis, int possibleIpCount) {
```

[0039] this.max = Math.max(1, countMax);
 periodInMillis = Math.max(1000L, periodInMillis);
 this.period = periodInMillis;

```
     final int ipCount = Math.max(16, possibleIpCount);  
     counts = new ConcurrentHashMap<Integer, AtomicInteger>(ipCount);  
     //reset count every period 每一个 IpRateLimiter 一个定时器
```

```

new Timer(true).scheduleAtFixedRate(new TimerTask() {
    public void run() {
        ConcurrentHashMap<Integer, AtomicInteger> countsOld = counts;
        if (!countsOld.isEmpty()) {
            counts = new ConcurrentHashMap<Integer, AtomicInteger>(
[0040]                                         ipCount);
        }
    }
}, periodInMillis, periodInMillis);
}

```

[0041] 每个控制类对应的定时时间和次数阈值均不同,如果一个控制类的定时时间比另一个控制类的定时时间长,那么该控制类的次数阈值也比另一个控制类的次数阈值大。

[0042] 本公开实施例中,可以对多个控制类,按照对应的定时时间以及次数阈值均递增的顺序级联,也即升序排序。可以将排序后的控制类放到一个数组里面,这样就形成了有规则的按照定时时间升序的级联时间指标。

[0043] 对多个控制类进行排序的伪代码表示如下:

```

IpRateLimiter[] limiters = null;

public IpRateLimiter(IpRateLimiter... limiters) {
    this.limiters = limiters != null ? limiters :
        new IpRateLimiter[] {};
}

//按 period 升序排序
for (int i1 = 0; i1 < this.limiters.length; i1++) {
    for (int i2 = i1; i2 < this.limiters.length; i2++) {
        if (this.limiters[i2].period < this.limiters[i1].period) {
            IpRateLimiter l1 = this.limiters[i1];
            this.limiters[i1] = this.limiters[i2];
            this.limiters[i2] = l1;
        }
    }
}

//此 RateLimiter (组)的 period 认为是最长的 RateLimiter 的 period
this.period = this.limiters[i1].period;

```

```
        }  
[0045]    }
```

[0046] 需要说明的是,在对每个控制类对应的定时时间及次数阈值进行设置时,每个级别的控制类对应的次数阈值小于一个乘积,该乘积为该控制类对应的定时时间与上一个级别的控制类对应的定时时间的商,与上一个级别的控制类对应的次数阈值的乘积。比如,设置有3个控制类级联,第一级别为控制类1,第二级别为控制类2,第三级别为控制类3。控制类1对应的定时时间为A,次数阈值为a;控制类2对应的定时时间为B,次数阈值为b;控制类3对应的定时时间为C,次数阈值为c。那么,b小于 $(B/A) \times a$,c小于 $(C/B) \times b$ 。

[0047] 举例来说,控制类1对应的额定时时间为1s,次数阈值为5次;控制类2对应的定时时间为5s,次数阈值为9次;控制类3对应的定时时间为9s,次数阈值为13次.则表示允许同一个客户端在1s内最多访问5次,5s内最多访问9次,9s内最多访问13次。这3个控制类的伪代码表示如下:

```
limiter = new IpRateLimiter(  
    new IpRateLimiter( 9, 5 * 1000L, 1000),  
[0048]    new IpRateLimiter(13, 9 * 1000L, 1000),  
    new IpRateLimiter( 5, 1 * 1000L, 1000)  
)
```

[0049] 服务器在接收到客户端发送的接口访问请求后,可以从第一个级别的控制类开始,逐个确定在该控制类中该客户端的标识对应的当前访问次数,并判断当前访问次数在该控制类对应的定时时间内,是否超过该控制类对应的次数阈值。

[0050] 针对每个级别的控制类,在确定出在该控制类中所述客户端的标识对应的当前访问次数后,判断当前访问次数是否超过该控制类对应的次数阈值。在判断出客户端在该控制类中的当前访问次数是否超过该控制类的次数阈值后,可以将判断结果告知下一个级别的控制类。比如,可以通过一个变量表示,如果该变量为true,则表示未超过,如果该变量为false,则表示超过。

[0051] 针对第一个级别的控制类,确定在该控制类中发送接口访问请求的客户端的标识对应的当前访问次数的步骤可以包括:对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数。也即,无论在该控制类中是否记录有所述客户端的标识对应的访问次数,均对在所述控制类中所述客户端的标识对应的访问次数增1。比如,如果在该控制类中记录有所述客户端的标识对应的访问次数,并且记录的访问次数为1,则确定在该控制类中所述客户端的标识对应的当前访问次数为2,如果当前访问次数没有超过次数阈值则更新记录,否则可以不更新记录。如果在该控制类中未记录所述客户端的标识对应的访问次数,则记录所述客户端的标识对应的访问次数,并且访问次数为1。

[0052] 确定出在第一个级别的控制类中所述客户端的标识对应的当前访问次数后,可以判断所述当前访问次数是否超过(此处超过表示大于)第一个级别的控制类对应的次数阈值。比如,按照上述举例,第一个级别的控制类的次数阈值为5次,如果当前访问次数为6次,则表示超过次数阈值。

[0053] 针对从第二个级别开始的控制类,确定在该控制类中发送接口访问请求的客户端的标识对应的当前访问次数的步骤可以包括:依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数。

[0054] 如果上一个级别的控制类中所述当前访问次数未超过所述次数阈值,则查找在所述控制类中是否记录有所述客户端的标识对应的访问次数。如果所述控制类中记录有所述客户端的标识对应的访问次数,则对所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;如果所述控制类中未记录所述客户端的标识对应的访问次数,则确定所述客户端的标识对应的当前访问次数为0。比如,如果在该控制类中记录有所述客户端的标识对应的访问次数,并且记录的访问次数为2,则确定在该控制类中所述客户端的标识对应的当前访问次数为3,如果当前访问次数没有超过次数阈值则更新记录,否则可以不更新记录。如果在该控制类中未记录所述客户端的标识对应的访问次数,则不对该客户端的访问次数进行计数。

[0055] 如果上一个级别的控制类中所述当前访问次数超过所述次数阈值,则对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数。比如,如果在该控制类中记录有所述客户端的标识对应的访问次数,并且记录的访问次数为2,则确定在该控制类中所述客户端的标识对应的当前访问次数为3,如果当前访问次数没有超过次数阈值则更新记录,否则可以不更新记录。如果在该控制类中未记录所述客户端的标识对应的访问次数,则记录所述客户端的标识对应的访问次数,并且访问次数为1。

[0056] 上述对从第二个级别开始的控制类的计数方式,可以避免每个控制类在每次均计数的情况造成的严重空间浪费。比如,三个控制类的定时时间设置是分别按照分钟、小时和天来设计的,如果每个用户的请求都在三个控制类中记录的话,而绝大多数的正常请求用户都不会超过第一个控制类的设置,那么第二个控制类和第三个控制类是可以不记录的。

[0057] 每个控制类的计数方法的伪代码表示如下:

[0058]

```
private boolean countAsSingle(int ip, boolean onlyIfExist) {  
    //获得计数, 或插入计数(如果不是 onlyIfExist) (在上一个级别没有超过次数阈值  
    //时会获得计数, 超过了则本级别也开始记录, 即插入计数)  
    AtomicInteger count = null;  
    if (onlyIfExist) { //如果上一个级别没有超过次数阈值, 则只判断本级别是否有对应客  
    //户端的访问次数记录  
        count = counts.get(ip);  
    } else { //如果上一个级别超过次数阈值, 则在本级别也开始计数, 即后一级别的计数  
    //是前一级别访问限制后触发的, 触发以后就可以获取该客户端的计数了  
        count = new AtomicInteger(0);  
        AtomicInteger countOld = counts.putIfAbsent(ip, count); //如果不存在则初始化为  
    //计数器为 0  
        if (countOld != null) {count = countOld;}  
    }  
  
    //不存在计数, 说明前一个级别没有超过次数阈值, 则本级别不计数  
    if (count == null) {return true;}  
  
    //如果计数超限, 不再继续递增计数, 直接返回超过超限  
    if (count.get() >= max) {  
        return false;  
    }  
  
    //计数, 同时再次检查是否超限  
    if (count.getAndIncrement() >= max) {  
        return false;  
    }  
  
    //返回未超限  
    return true;  
}
```

[0059]

[0060] 在步骤S13中,在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。

[0061] 经过上述步骤S12,每个控制类判断各自对应的当前访问次数是否超过次数阈值。如果在某个控制类中,所述当前访问次数超过所述次数阈值,则拒绝所述接口访问请求。如果在全部控制类中,所述当前访问次数均未超过所述次数阈值,则允许所述接口访问请求。

[0062] 级联控制的伪代码表示如下:

[0063]

```
private boolean countAsGroup(int ip, boolean onlyIfExist) {  
    if (limiters.length == 0) {return true;}  
  
    boolean ok = onlyIfExist; //ok 含义: true - 之前一级未超限  
    boolean okFinal = true; //okFinal 含义: 总体未超限  
  
    //第一层计数  
    //第二层开始, 若之前一级超限了, 即 ok 为 false, 则计数, 否则对应的 map 中存在  
    //该 IP 才计数  
    for (int i = 0; i < limiters.length; i++) {  
        ok = limiters[i].countAsSingle(ip, ok);  
        okFinal &= ok; //okFinal&是左右级联指标的结果的与操作, 即全部为 true 才为  
        //true, 有一个为 false 则不能访问资源  
    }  
  
    return okFinal;  
}
```

[0064] 图3是根据一示例性实施例示出的一种访问控制的效果示意图。图3中上方的曲线表示未采用本公开实施例的访问控制方式控制之前的控制效果曲线,下方的曲线表示采用本公开实施例的访问控制方式控制之后的控制效果曲线。由图3可知,采用本公开实施例的访问控制方式控制之后,可以实现一个非常平滑的访问控制。

[0065] 每个控制类的定时器会在到达定时时间后,清空所在控制类维护的map数据,保证每个定时时间周期内的访问次数在规定的定时时间内有效,当前定时时间结束以后则下一轮重新计数。但是不同控制类的定时时间不同,因此清空时间不同就可以很好的彼此帮衬记录自己监管的时间范围内的访问次数,共同限制一个客户端的访问权限。

[0066] 本公开的实施例通过多个控制类,能够在不同的定时时间内,对同一客户端的访问次数进行不同次数阈值下的联合控制,因此能够解决在某个时间段内出现突发流量而无法控制的问题,通过更多时间指标,更细粒度的控制,可以实现一个非常平滑的控制,能够

更加完善控制方式,提高控制的准确性,提升服务器的性能,并且控制过程基于内存计算,速度快,效果好。

[0067] 图4是根据一示例性实施例示出的一种访问控制装置框图。参照图4,该装置包括接收单元401、确定单元402和控制单元403。

[0068] 接收单元401,被配置为接收客户端发送的接口访问请求,所述接口访问请求包括所述客户端的标识;

[0069] 确定单元402,被配置为针对预设的多个控制类,逐个确定在所述控制类中所述客户端的标识对应的当前访问次数;每个控制类对应一个定时时间及一个次数阈值;

[0070] 控制单元403,被配置为在所述当前访问次数在所述控制类对应的定时时间内超过所述控制类对应的次数阈值时,拒绝所述接口访问请求。

[0071] 在一种可选实施方式中,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联。所述确定单元402包括:第一确定模块,被配置为针对第一个级别的控制类,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;第二确定模块,被配置为针对从第二个级别开始的控制类,依据上一个级别的控制类中所述当前访问次数是否超过所述次数阈值,确定在所述控制类中所述客户端的标识对应的当前访问次数。

[0072] 在一种可选实施方式中,所述第二确定模块包括:查找子模块,被配置为在上一个级别的控制类中所述当前访问次数未超过所述次数阈值时,查找在所述控制类中是否记录有所述客户端的标识对应的访问次数;第三确定子模块,被配置为在所述控制类中记录有所述客户端的标识对应的访问次数时,对所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数;在所述控制类中未记录所述客户端的标识对应的访问次数时,确定所述客户端的标识对应的当前访问次数为0;第四确定子模块,被配置为在上一个级别的控制类中所述当前访问次数超过所述次数阈值时,对在所述控制类中所述客户端的标识对应的访问次数增1,作为所述客户端的标识对应的当前访问次数。

[0073] 在一种可选实施方式中,所述多个控制类按照所述定时时间以及所述次数阈值均递增的顺序级联;每个级别的控制类对应的次数阈值小于一个乘积;所述乘积为所述控制类对应的定时时间与上一个级别的控制类对应的定时时间的商,与所述上一个级别的控制类对应的次数阈值的乘积。

[0074] 在一种可选实施方式中,所述装置还包括:定时单元,被配置为针对每个控制类,在到达所述控制类对应的定时时间时,将所述控制类中记录的全部访问次数清空。

[0075] 本公开的实施例通过多个控制类,能够在不同的定时时间内,对同一客户端的访问次数进行不同次数阈值下的联合控制,因此能够解决在某个时间段内出现突发流量而无法控制的问题,通过更多时间指标,更细粒度的控制,可以实现一个非常平滑的控制,能够更加完善控制方式,提高控制的准确性,提升服务器的性能。

[0076] 图5是根据一示例性实施例示出的一种用于访问控制的装置500的框图。例如,装置500可以被提供为一服务器。

[0077] 参照图5,装置500包括处理组件522,其进一步包括一个或多个处理器,以及由存储器532所代表的存储器资源,用于存储可由处理组件522的执行的指令,例如应用程序。存储器532中存储的应用程序可以包括一个或一个以上的每一个对应于一组指令的模块。此

外,处理组件522被配置为执行指令,以执行上述方法。

[0078] 装置500还可以包括一个电源组件526被配置为执行装置500的电源管理,一个有线或无线网络接口550被配置为将装置500连接到网络,和一个输入输出(I/O)接口558。装置500可以操作基于存储在存储器532的操作系统,例如Windows ServerTM,Mac OSXTM,UnixTM,LinuxTM,FreeBSDTM或类似。

[0079] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器,上述指令可由电子设备的处理器执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0080] 在示例性实施例中,还提供了一种计算机程序产品,当所述计算机程序产品中的指令由服务器的处理器执行时,使得服务器能够执行上述访问控制方法。

[0081] 本领域技术人员在考虑说明书及实践这里公开的发明后,将容易想到本发明的其它实施方案。本申请旨在涵盖本发明的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本发明的一般性原理并包括本公开未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本发明的真正范围和精神由下面的权利要求指出。

[0082] 应当理解的是,本发明并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本发明的范围仅由所附的权利要求来限制。

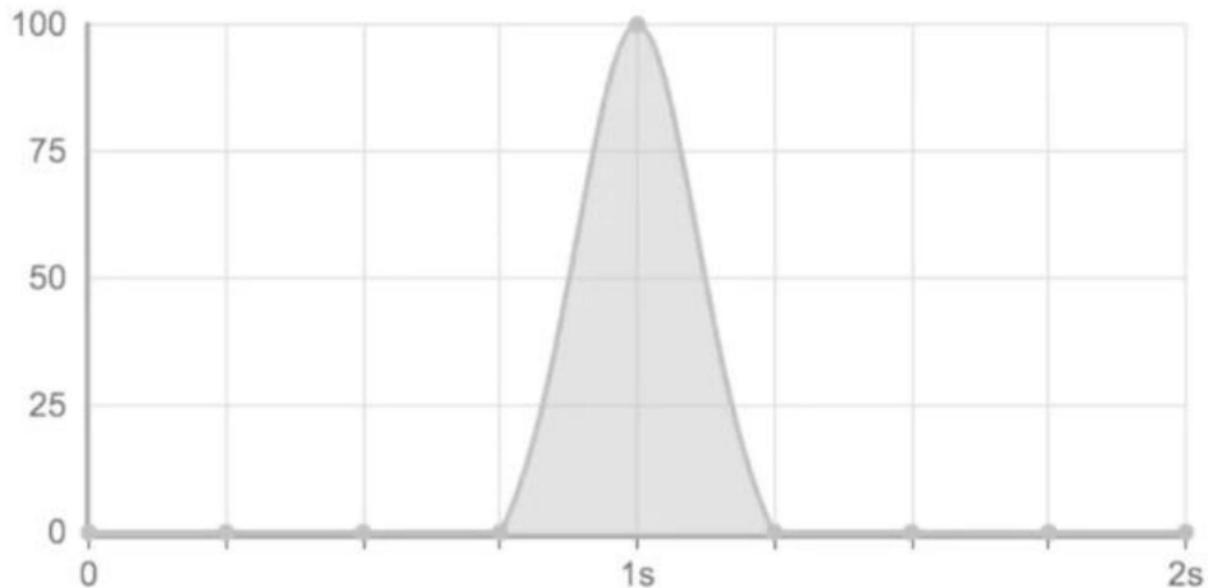


图1

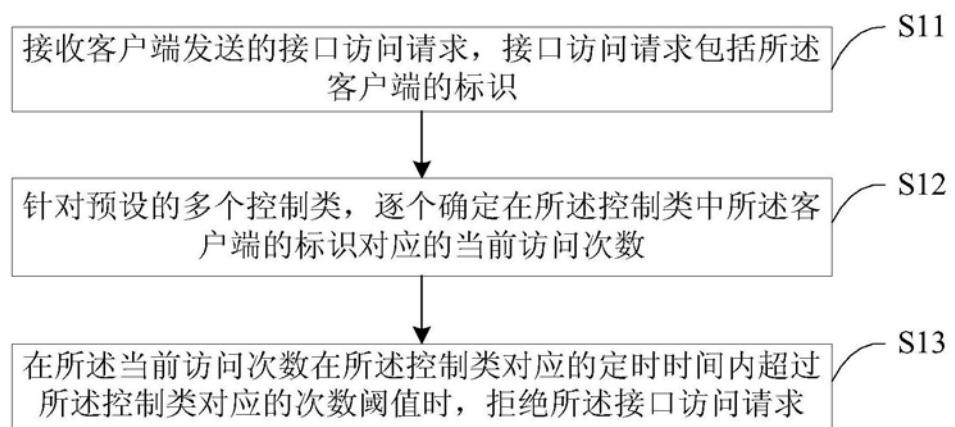


图2

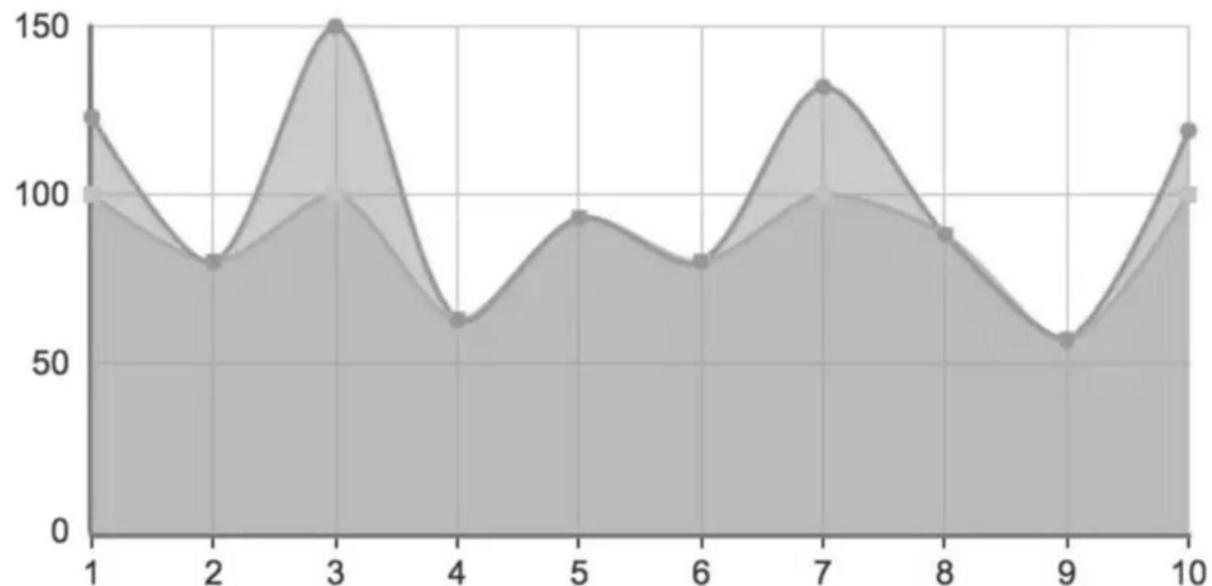


图3



图4

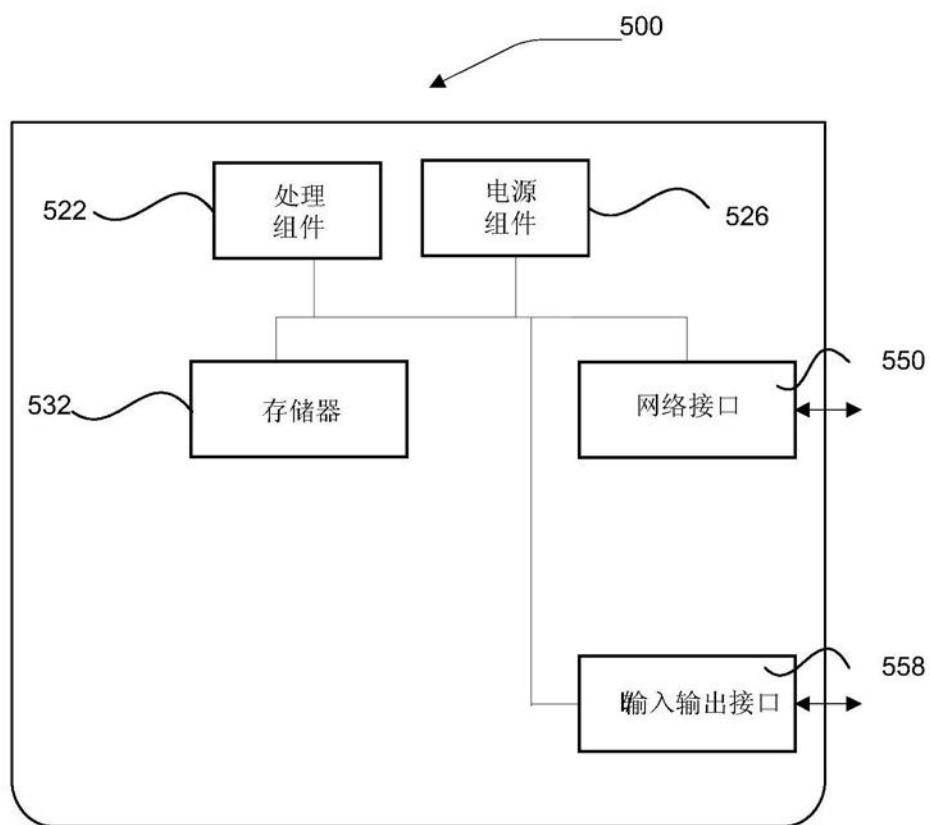


图5