(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 112165508 B
(45)授权公告日 2021.07.09

(21)申请号 202010889194.0

(22)申请日 2020.08.28

(65)同一申请的已公布的文献号
申请公布号 CN 112165508 A

(43)申请公布日 2021.01.01

(66)本国优先权数据
202010855391.0 2020.08.24 CN

(73)专利权人 北京大学
地址 100871 北京市海淀区颐和园路5号

(72)发明人 孙广宇 王晓阳

(74)专利代理机构 北京万象新悦知识产权代理有限公司 11360
代理人 黄凤茹

(51)Int.Cl.
H04L 29/08(2006.01)

(56)对比文件
CN 1156304 A,1997.08.06
CN 102456048 A,2012.05.16
US 2013254407 A1,2013.09.26
CN 105706086 A,2016.06.22
CN 102932419 A,2013.02.13
丛犁.《面向工业物联网的分布式数据存储分配方案》.《电力信息与通信技术》.2020,52-57.
Yu Xiang.《Multi-tenant Latency Optimization in Erasure-Coded Storage with Differentiated Services》.《IEEE》.2015,790-791.

审查员 杨浩磊

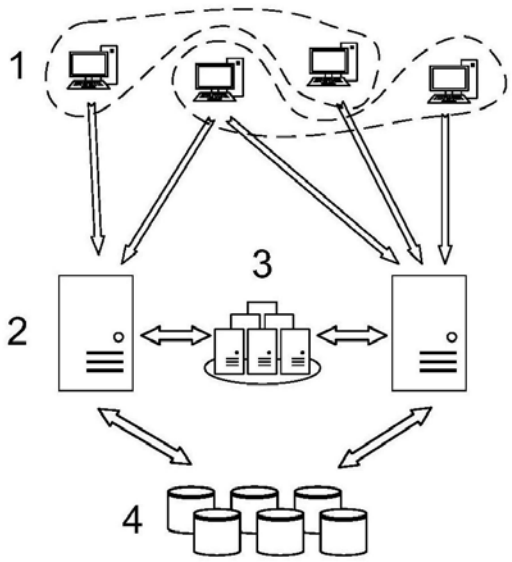
权利要求书3页 说明书5页 附图3页

(54)发明名称

一种多租户云存储请求服务的资源分配方法

(57)摘要

本发明公布了一种多租户分布式存储请求服务的资源分配方法,在多租户分布式存储系统中,为每一个请求设置最近和最远执行时间,优先为已错过最远执行时间的请求分配系统资源,次优先为已到达最近执行时间的请求分配系统资源,即通过优先响应未达到请求响应频率下限的租户请求,推迟响应已达请求响应频率上限的租户请求,实时、精确地限制各个租户请求响应频率,实现多租户分布式存储请求服务的资源分配。本发明能够达到实时、精确地限制各个租户请求响应频率的效果,并保证请求响应的公平性,且具有低代价、可扩展的优势。



CN 112165508 B

1.一种多租户云存储请求服务的资源分配方法,在多租户云存储系统中,为每一个请求设置最近和最远执行时间,优先为已错过最远执行时间的请求分配系统资源,次优先为已到达最近执行时间的请求分配系统资源,即通过优先响应未达到请求响应频率下限的租户请求,推迟响应已达请求响应频率上限的租户请求,实时、精确地限制各个租户请求响应频率,实现多租户云存储请求服务的资源分配;

所述多租户云存储系统包括用于存放用户数据的存储节点阵列、多个门户服务器和多个参数服务器;系统同时为多个租户提供服务;

每个租户含有多个个体;每个租户对应一个租户编号;每个租户的全局请求情况处理对应一台参数服务器;

存储节点阵列包括多个存储节点;

多个门户服务器各自拥有全局唯一的门户服务器编号;多个门户服务器通过与存储节点进行通信,将租户服务请求中的控制命令和数据内容转发到相应的存储节点阵列中存放目标数据的存储节点,并为租户个体返回状态码和所需数据;每个门户服务器同时响应多个服务请求;门户服务器定期与参数服务器同步全局统计信息;

所述多租户云存储请求服务的资源分配方法包括以下步骤:

A.多租户云存储系统中的所有租户根据自身需求,设定请求响应频率的上限和下限;

多租户云存储系统中的租户t根据自身需求,设定请求响应频率的上限l_t和下限r_t;

租户t设定的请求响应频率的上限l_t和下限r_t表示单位时间内多租户云存储系统应当给来自租户t的r_t～l_t个服务请求分配资源;

B.多个门户服务器通过与存储节点进行通信,将租户服务请求中的控制命令和数据内容转发到相应的存储节点阵列;

C.每个门户服务器为每个租户维护一组该服务器已分配资源的请求的本地数据,并定期与参数服务器同步全局统计数据;参数服务器即得到该租户及同租户下所有个体用户请求的累积数量分布;

D.在租户t的服务请求到达门户服务器时,将该服务请求加入该租户的请求等待队列中;门户服务器根据当前的时间点计算得到两个标签,记为L和R,分别表示该请求应当被执行的最近时间和最远时间;

在门户服务器上,针对租户t的R标签计算方式为:

$$\max\{R'_t+\frac{1}{\rho_t^i\cdot r_t},\tau\},$$

其中,R'_t代表该门户服务器收到的上一个来自相同租户的个体的请求所分配的R标签值,τ表示当前本机的时间戳,max表示二者中取较大的值;ρ_t^i为全局服务数量中租户t的R类请求;

将租户t的请求记为δ_t^i;L标签的计算为:

$$\max\{L'_t+\frac{1}{\delta_t^i\cdot l_t},\tau\},$$

其中,L'_t代表该门户收到的上一个来自相同租户的请求所分配的L标签值;

E.当门户服务器有空闲能力处理更多请求时,遍历本地所有租户的请求等待队列,采用以下方法对请求进行资源分配:

a)优先为R类请求分配资源；

所述R类请求是已经迟于最远执行时间的请求，即R值小于当前时间戳的请求；

b)再选择L类请求中L值最小的一组请求进行分配；

所述L类请求是指已经到达最近执行时间的请求，即L值小于当前时间戳的请求；

F.根据门户服务器和参数服务器记录的R类请求和L类请求，计算得到任意时刻每个门户服务器提供的服务数量在全局服务数量中的占比；

G.门户服务器在响应租户请求时，根据请求的内容找到存储节点阵列中的目标节点，触发并监听该节点与租户之间的数据传输与读写过程；

当传输结束，门户节点有能力服务更多的请求时，该节点根据步骤E选择下一批请求进行响应；

通过上述步骤，实现多租户云存储请求服务的资源分配。

2.如权利要求1所述多租户云存储请求服务的资源分配方法，其特征是，步骤F中计算得到任意时刻每个门户服务器提供的服务数量在全局服务数量中的占比，具体方法为：

ρ_t^i中包括第i个门户服务器中的$\rho_t^i \cdot M$个请求；δ_t^i包括L类请求和R类请求；

则第i个门户服务器贡献了其中的$\delta_t^i \cdot M$个请求。

3.如权利要求1所述多租户云存储请求服务的资源分配方法，其特征是，步骤C中，每隔一段时间，门户服务器与参数服务器同步各租户的统计信息，同步的周期为p；

门户服务器i在为来自租户t的请求分配资源时，在门户服务器维护参数P_t和Δ_t；参数P_t和Δ_t分别表示全局服务R类和L类请求的预期频率，单位为个/秒；

门户服务器还维护参数R类请求的计数器CR_t、L类请求的计数器CL_t，分别表示从上次同步到当前为止两类请求在本门户执行的个数；

通过参数服务器$PNode_{\phi(t)}$记录租户t的请求在各个门户服务器的执行分配资源的情况PR_t^i与PL_t^i，PR_t^i与PL_t^i分别代表从上次与门户服务器i同步到当前为止，整个系统中执行分配资源的R类和L类请求的总数量；

上述参数通过以下过程进行初始化和信息更新：

a)初始情况下，门户服务器上关于租户t的参数P_t和Δ_t初始化为g/p，CR_t和CL_t均置为0；参数服务器上关于租户t、门户i的参数PR_t^i与PL_t^i，均初始化为0；

b)门户服务器以R类请求的方式为来自租户t的一个请求分配资源时，本地参数CR_t和CL_t均自增1；以L类请求的方式分配资源时，只有CL_t自增1；

c)在门户服务器i与参数服务器同步时，假设$CL_t \neq 0$，那么门户i应当将计数器CR_t和CL_t的当前值上传至$PNode_{\phi(t)}$；

d)$PNode_{\phi(t)}$在收到来自门户服务器i关于租户t的信息时，返回PR_t^i与PL_t^i的当前值；

同时，参数服务器对租户t的相关参数执行更新如下：

$$PR_t^j \leftarrow \begin{cases} PR_t^j + CR_t, & j \neq i \\ 0, & j = i \end{cases}, \qquad PL_t^j \leftarrow \begin{cases} PL_t^j + CL_t, & j \neq i \\ 0, & j = i \end{cases}$$

e)门户服务器i收到参数服务器返回的PR_t^i与PL_t^i后，更新参数：$P_t \leftarrow \frac{PR_t^i}{p}, \Delta_t \leftarrow \frac{PL_t^i}{p}$；

同时，将计数器CR_t和CL_t的值重置为0。

4.如权利要求1所述多租户云存储请求服务的资源分配方法，其特征是，步骤D计算门户服务器i于τ_k时刻收到来自租户t的请求的R标签和L标签，再将该请求加入等待队列Q_t等待分配资源；计算门户服务器i于τ_k时刻收到来自租户t的请求的R标签和L标签，包括如下过程：

a)如果该门户第一次收到该租户的请求，则R标签和L标签的值为：$R_t^k = L_t^k = \tau_k$；

b)如果该门户之前收到过来自租户t的请求，则：

假设上一个来自租户t的请求于τ_{k-1}时刻到来，当时分配的R标签和L标签值分别为R_t^{k-1}和L_t^{k-1}；

估算出当前门户处理的R类和L类请求在全局中的占比为：$\rho_t = \frac{1}{(\tau_k - \tau_{k-1})P_t + 1}$，$\delta_t = \frac{1}{(\tau_k - \tau_{k-1})\Delta_t + 1}$；参数$P_t$和$\Delta_t$分别表示全局服务R类和L类请求的预期频率，单位为个/秒；

当前请求分配的R标签和L标签的值分别通过计算得到：

$$R_t^k = \max\{R_t^{k-1} + \frac{1}{\rho_t r_t}, \tau\},\ L_t^k = \max\{L_t^{k-1} + \frac{1}{\delta_t l_t}, \tau\},$$

其中max表示二者中取较大的值。

一种多租户云存储请求服务的资源分配方法

技术领域

[0001] 本发明涉及云存储技术，具体涉及一种多租户、多个体(用户)的云存储系统中云存储请求服务的资源分配方法。

背景技术

[0002] 云存储系统能够为用户提供可靠、可扩展且相对廉价的存储服务，同时为用户屏蔽了管理和维护存储系统的代价。典型的云存储系统分配和调度资源的基本单位为租户，即租用云存储服务的用户。每个租户通常包含多个独立的访问个体，同一租户中的多个个体将共享这一租户拥有的存储和带宽等资源。多租户、多个体的云存储系统同时为多个租户提供服务，每个租户含有多个个体。

[0003] 对于云存储系统来说，一方面，云存储的用户要求服务提供商保证请求的响应频率至少达到某一下限，以保证租户获取到的存储服务的稳定性；另一方面，云存储的提供商希望能够在尽量减少设备开支的情况下为尽可能多的租户提供服务，增加收入。常见的商业云存储产品中，租户根据自己的需求定制服务套餐，存储服务提供商将据此分配资源。

[0004] 传统的资源分配算法存在以下不足之处：

[0005] (1)同一租户中经常有多个独立的个体，它们可同时向云存储系统发起资源请求。云存储系统以租户为单位分配资源，所以同租户个体分配到的资源的累加和应满足服务套餐的需求。然而，传统的算法缺乏实时追踪同租户的多个个体的资源使用情况的机制，因此无法准确地为这些多个体租户分配资源。

[0006] (2)在实际应用中，租户的需求通常会随着时间变化，大多数情况下并不会占满所分配的资源，而少数时间会超出套餐限制。为了更好的服务质量，云存储系统应当优先服务未占满资源的租户，延迟响应超出套餐限制的数据请求。传统的算法缺乏对这些请求的优先级划分，难以达到响应来自不同租户的请求时的公平性。

[0007] 综上所述，在多租户场景下，传统的云存储请求分配与调度算法难以实时、精确地为每一个租户的云存储请求分配合适的服务资源。

发明内容

[0008] 为了克服上述现有技术的不足，本发明了提供一种多租户场景下的云存储请求服务的分配方法，达到了实时、精确地限制各个租户请求响应频率的效果，并保证了请求响应的公平性，且具有低代价、可扩展的优势。本发明能够将各个租户的请求响应频率尽量限制在租户所给定的响应频率的上限与下限之间。在带宽、存储、计算等资源有限的情况下，无法同时响应所有的用户请求，优先为未达到响应频率下限的租户请求分配资源，而推迟响应已达响应频率上限的租户的请求。

[0009] 在本发明中，云存储服务的每个租户中包含多个独立的个体(用户)，这些个体向云存储系统提交数据访问请求。请求会被定向转发到一些门户服务器(Gate)，这些服务器负责对用户的响应以及对存储阵列的访问过程。一个云存储系统中通常包含多个门户服务

器。除此之外，系统中还有一些参数服务器(PNode)，负责维护各个租户的请求执行情况的全局统计信息。本发明为每一个请求设置最近和最远执行时间，优先为已错过最远执行时间的请求分配系统资源，次优先为已到达最近执行时间的请求分配系统资源。通过这样的方式，尽量将对各个租户的请求响应频率限制在租户所给定的上下限之间

[0010] 本发明的技术方案是：

[0011] 一种多用户(租户)云存储请求服务的资源分配方法，在多租户云存储系统中，系统包括负责存放用户数据的存储节点阵列(包括多个存储节点)，多个门户服务器(Gate)和多个参数服务器(PNode)；系统同时为多个租户提供服务，每个租户含有多个个体；本发明通过为每一个请求设置最近和最远执行时间，优先为已错过最远执行时间的请求分配系统资源，次优先为已到达最近执行时间的请求分配系统资源，即优先响应未达到下限的租户请求，而推迟响应已达上限的租户的请求，实时、精确地限制各个租户请求响应频率，实现多租户云存储请求服务的资源分配；包括以下步骤：

[0012] A.多租户云存储系统中的所有租户根据自身需求，设定请求响应频率的上限和下限；

[0013] 在多租户云存储系统中，每个租户对应一个租户编号；每个租户的全局请求情况处理对应一台参数服务器；租户t根据自身需求，设定请求响应频率的上限l_t和下限r_t；

[0014] 租户t设定的请求响应频率的上限l_t和下限r_t表明单位时间内多租户云存储系统应当给r_t～l_t个来自租户t的服务请求分配资源。

[0015] 每个租户在多租户云存储系统中注册时，被分配一个全局唯一的编号(租户编号)。多租户云存储系统根据该租户编号挑选一台参数服务器，用来负责该租户的全局请求执行情况的信息统计。

[0016] B.多个门户服务器通过与存储节点进行通信，将租户服务请求中的控制命令和数据内容转发到相应的存储阵列(存储节点阵列)，每个门户服务器同时响应多个服务请求，为服务请求分配资源。

[0017] 系统中有多个门户服务器，各自拥有全局唯一的门户服务器编号，负责与存储阵列通信，将请求中的控制命令和数据等内容转发到存储阵列中存放目标数据的节点，并为租户个体返回状态码和所需数据。每个门户服务器可以同时服务多个请求。

[0018] C.每个门户服务器为每个租户维护一组该服务器已分配资源的请求的本地数据，并定期与参数服务器同步全局统计数据；参数服务器即得到该租户及同租户下所有个体用户请求的累积数量分布；

[0019] 门户服务器定期与参数服务器同步全局统计信息。每个门户服务器为租户维护一组本地的统计数据，表明从上次同步到某时刻为止，本服务器已分配资源的请求的累积数量。参数服务器收到这些同步数据后，可以得到同租户下所有用户的累积数量和分布。

[0020] D.在租户的服务请求到达门户服务器时，将该服务请求加入该租户的请求等待队列中。门户服务器根据当前的时间点，计算出两个标签，记为L和R，分别表示该请求应当被执行的最近执行时间和最远执行时间。

[0021] 在门户服务器上，针对租户t的R标签计算方式为：

[0022] $$\max\{R_t' + \frac{1}{\rho_t^t \cdot r_t}, \tau\},$$

[0023] 其中，R'_t代表该门户收到的上一个来自相同租户(但不一定是同一个个体)的请求所分配的R标签值，τ表示当前本机的时间戳，max表示二者中取较大的值；

[0024] 同样，L标签的计算则为：

[0025] $$\max\{L'_t + \frac{1}{\delta_t^i \cdot l_t}, \tau\},$$

[0026] 其中L'_t代表该门户收到的上一个来自相同租户的请求所分配的L标签值。

[0027] E.门户服务器在有空闲能力处理更多请求时，会遍历本地所有租户的请求等待队列，采用以下方法对请求进行资源分配：

[0028] a)优先为R类请求分配资源；

[0029] 所述R类请求是已经迟于最远执行时间的请求，即R值小于当前时间戳的请求；

[0030] b)再选择L类请求中L值最小的一组请求进行分配；

[0031] 所述L类请求是指已经到达最近执行时间的请求，即L值小于当前时间戳的请求；

[0032] 也就是，如果存在已经迟于最远执行时间的请求，即R值小于当前时间戳的请求，这类请求被称为R类请求，则优先为该请求分配资源。否则，找到已经到达最近执行时间的请求，即L值小于当前时间戳的请求，这类请求被称为L类请求，则选择L值最小的一组请求进行分配。

[0033] F.根据门户服务器和参数服务器记录的R类和L类请求，计算得到任意时刻每个门户服务器提供的服务数量在全局中的占比；

[0034] 门户服务器的局部统计信息和参数服务器的全局统计信息区分R类和L类请求，并计算得到任意时刻每个门户服务器提供的服务数量，在全局服务(系统中所有门户为同租户提供的服务)中的占比。如ρ_t^i代表全局一共执行了租户t的M个R类请求时，其中包括了第i个门户服务器中的$\rho_t^i \cdot M$个请求。同理，δ_t^i代表系统响应了租户t的M个请求(包括L类和R类)时，第i个门户服务器贡献了其中的$\delta_t^i \cdot M$个请求。t代表租户编号，i代表门户服务器编号。

[0035] G.门户服务器在响应租户请求时，会根据请求的内容找到存储阵列中的目标节点，触发并监听该节点与租户之间的数据传输与读写过程。当传输结束，门户节点有能力服务更多的请求时，该节点将根据步骤E选择下一批请求进行响应。

[0036] 通过上述步骤，实现多租户云存储请求服务的资源分配。

[0037] 与现有技术相比，本发明的有益效果是：

[0038] 本发明提供一种多用户(租户)云存储请求服务的资源分配方法，多租户云存储系统同时为多个包含多个个体的租户提供服务，通过优先响应未达到下限的租户请求，推迟响应已达上限的租户的请求，实时、精确地限制各个租户请求响应频率，实现多租户云存储请求服务的资源分配。本发明方法为每一个请求设置最近和最远执行时间，优先为已错过最远执行时间的请求分配系统资源，次优先为已到达最近执行时间的请求分配系统资源。通过这样的方式，尽量将对各个租户的请求响应频率限制在租户所给定的上下限之间。这种方法有以下一些优点：

[0039] 本发明可以精确地限制系统对各租户请求的响应频率，保证了对不同租户间请求进行资源分配调度的公平性；本发明中与资源分配相关的额外交互只有门户节点和参数服务器之间的定期同步，代价较小；在本发明所中，各个租户的信息单独收集，请求单独分配

资源，系统有着高可扩展性的优势。

附图说明

[0040] 图1为本发明具体实施采用的多租户云存储系统的底层结构示意图；

[0041] 图2为本发明所提供的资源分配方法的时序图；

[0042] 图1、2中，1为租户，每个租户均由多个独立的个体组成；2为云存储系统中的门户节点；3为参数服务器；4为存储阵列；a为租户向门户节点发送请求的过程；b为门户节点向租户告知请求已完成的过程；c为门户节点向参数服务器发送本地统计信息的过程；d为参数服务器返回全局统计信息的过程。

[0043] 图3为本发明资源分配方法的标签分配的流程框图。

[0044] 图4为本发明资源分配方法根据标签选择待执行请求进行资源分配的流程框图。

具体实施方式

[0045] 下面结合附图，通过实施例进一步描述本发明，但不以任何方式限制本发明的范围。

[0046] 附图1展示了本发明的底层架构，附图2展示了本发明提出的资源分配方法的时序图。两图中，1所指代的每个租户均由多个独立的个体组成，它们向2所指代的云存储系统中的门户节点发送请求。这些门户节点根据资源分配方法，将允许执行的请求转发给4所标识的存储阵列进行具体的读写等操作。除了门户节点和存储阵列之外，云存储系统中存在一些特殊的节点3作为参数服务器PNode，门户节点将定期与这些服务器同步所需要的全局信息。

[0047] 具体包含以下步骤：

[0048] 1.每个租户在云存储系统中注册时，将被分配一个全局唯一的编号t。系统根据该编号，通过一致性哈希算法Φ挑选一台参数服务器$PNode_{\Phi(t)}$来维护该租户的全局统计信息。租户t在选择服务套餐时，需要根据自身需求设定请求响应频率的上限l_t和下限r_t。

[0049] 2.系统中有g个门户服务器，各自拥有全局唯一的门户编号i。门户服务器负责与存储节点通信，将请求中的控制命令和数据等内容转发到相应的存储阵列，并为租户个体返回执行结果。每个门户服务器可以同时服务多个请求。

[0050] 3.每隔一段时间，门户服务器与参数服务器同步各租户的统计信息，下设同步的周期为p。门户服务器i在为来自租户t的请求分配资源时，需要在门户服务器本机维护参数P_t和Δ_t，表示全局服务R类和L类请求的预期频率，单位为个/秒；以及R类、L类请求的计数器CR_t和CL_t，分别表示从上次同步到当前为止两类请求在本门户执行的个数。与此对应，参数服务器$PNode_{\Phi(t)}$负责记录租户t的请求在各个门户服务器的执行分配资源的情况PR_t^i与PL_t^i，PR_t^i与PL_t^i分别代表从上次与门户服务器i同步到当前为止，整个系统中执行分配资源的R类和L类请求的总数量。这些全局信息的初始化和更新过程如下：

[0051] a)初始情况下，门户服务器上关于租户t的参数P_t和Δ_t初始化为g/p，CR_t和CL_t均置为0；参数服务器上关于租户t、门户i的参数PR_t^i与PL_t^i，均初始化为0。

[0052] b)门户服务器以R类请求的方式为来自租户t的一个请求分配资源时，本地参数

CR_t和CL_t均自增1;以L类请求的方式分配资源时,只有CL_t自增1。

[0053] c)在门户服务器i与参数服务器同步时,假设$CL_t \neq 0$,那么门户i应当将计数器CR_t和CL_t的当前值上传至$PNode_{\phi(t)}$。附图2中表示为c消息。

[0054] d)$PNode_{\phi(t)}$在收到来自门户服务器i关于租户t的信息时,返回PR_t^i与PL_t^i的当前值。附图2中表示为d消息。同时,参数服务器也将对租户t的相关参数执行更新:

$$PR_t^j \leftarrow \begin{cases} PR_t^j + CR_t, & j \neq i \\ 0, & j = i \end{cases}, \quad PL_t^j \leftarrow \begin{cases} PL_t^j + CL_t, & j \neq i \\ 0, & j = i \end{cases}。$$

[0055] e)门户服务器i收到参数服务器返回的PR_t^i与PL_t^i后,更新参数:$P_t \leftarrow \frac{PR_t^i}{p}, \Delta_t \leftarrow \frac{PL_t^i}{p}$。同时,将计数器$CR_t$和$CL_t$的值重置为0。

[0056] 4.门户服务器i于τ_k时刻收到来自租户t的请求时,它将该请求加入等待队列Q_t等待分配资源。同时,计算出该请求的R标签和L标签。附图3展示了这一过程:

[0057] a)如果这是该门户第一次收到该租户的请求,则为之分配的R标签和L标签的值为$R_t^k = L_t^k = \tau_k$。

[0058] b)如果该门户之前收到过来自租户t的请求,则假设上一个来自租户t的请求于τ_{k-1}时刻到来,当时分配的R标签和L标签值分别为R_t^{k-1}和L_t^{k-1}。据此可估算出当前门户处理的R类和L类请求在全局中的占比:$\rho_t = \frac{1}{(\tau_k - \tau_{k-1})P_t + 1}, \delta_t = \frac{1}{(\tau_k - \tau_{k-1})\Delta_t + 1}$。当前请求分配的R标签和L标签的值分别计算为:$R_t^k = \max\{R_t^{k-1} + \frac{1}{\rho_t r_t}, \tau\}, L_t^k = \max\{L_t^{k-1} + \frac{1}{\delta_t l_t}, \tau\}$,其中max表示二者中取较大的值。

[0059] 5.假设在时刻τ,门户服务器有能力处理更多请求,则会遍历本地所有租户的请求等待队列Q_t的头部,选择合适的请求执行。附图4展示了这一过程:

[0060] a)如果存在R值小于τ的请求,则应当找到所有这样的请求,按R值从小到大排序,依顺序进行优先执行。以这种形式被挑选出的请求为步骤3b中提到的“R类请求”。

[0061] b)否则,找到L值小于τ的请求,选择L值最小的一组请求执行。以这种形式被挑选出的请求为步骤3b中提到的“L类请求”。

[0062] 需要注意的是,公布实施例的目的在于帮助进一步理解本发明,但是本领域的技术人员可以理解:在不脱离本发明及所附权利要求的精神和范围内,各种替换和修改都是可能的,包括但不限于:占比值ρ和δ的计算方式、更新周期p的动态调整、租户信息与参数服务器的映射关系等。因此,本发明不应局限于实施例所公开的内容,本发明要求保护的范围以权利要求书界定的范围为准。

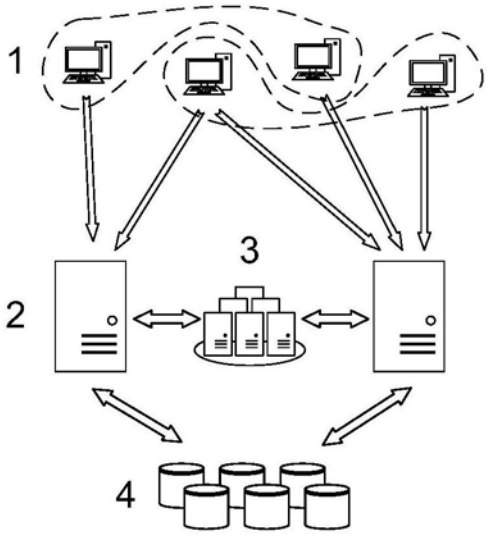


图1

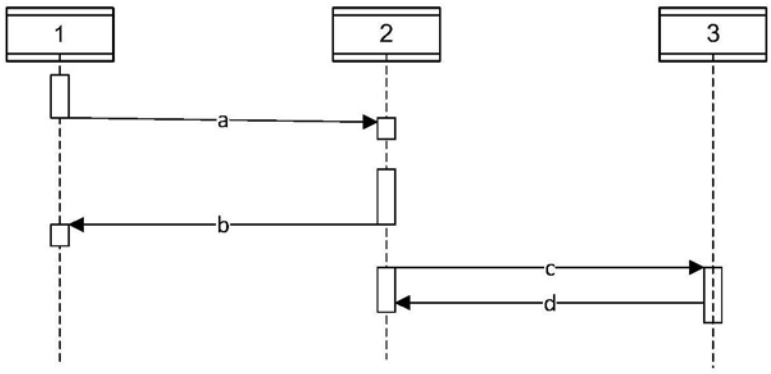


图2

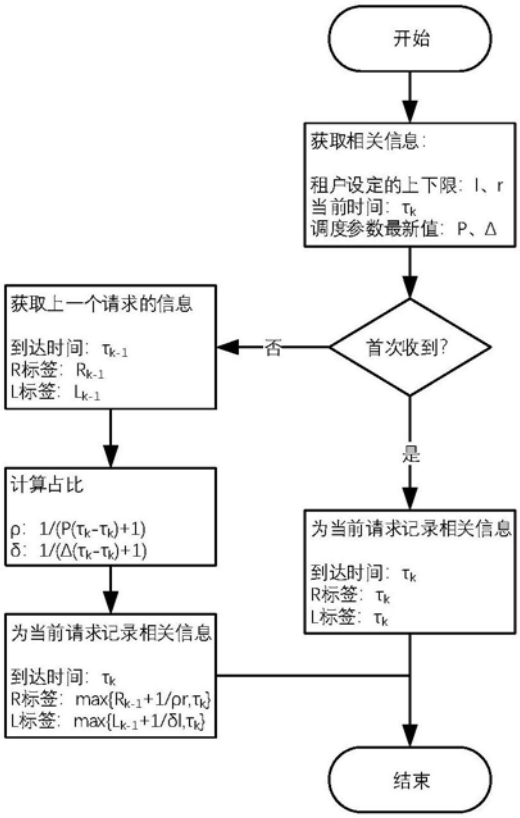

开始
获取相关信息:
租户设定的上下限: l、r
当前时间: τ_k
调度参数最新值: P、Δ
首次收到?
否
是
获取上一个请求的信息
到达时间: τ_{k-1}
R标签: R_{k-1}
L标签: L_{k-1}
计算占比
ρ: $1/(P(\tau_k-\tau_k)+1)$
δ: $1/(\Delta(\tau_k-\tau_k)+1)$
为当前请求记录相关信息
到达时间: τ_k
R标签: $\max\{R_{k-1}+1/\rho r,\tau_k\}$
L标签: $\max\{L_{k-1}+1/\delta l,\tau_k\}$
为当前请求记录相关信息
到达时间: τ_k
R标签: τ_k
L标签: τ_k
结束

图3

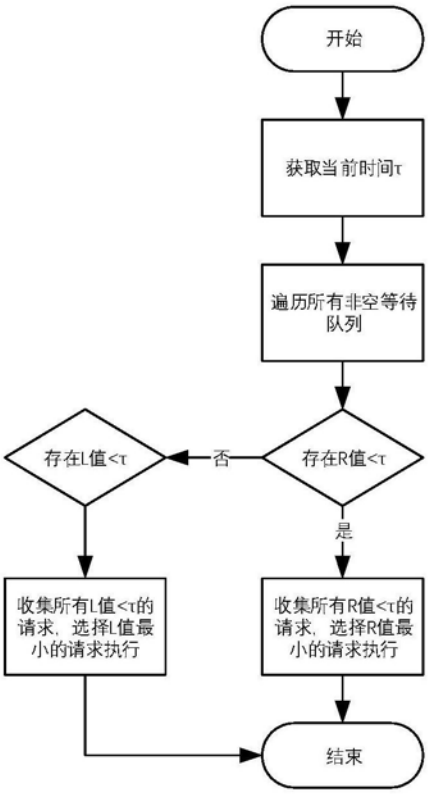

开始
获取当前时间τ
遍历所有非空等待队列
存在R值<τ
否
存在L值<τ
是
收集所有L值<τ的请求，选择L值最小的请求执行
收集所有R值<τ的请求，选择R值最小的请求执行
结束

图4