



(12) 发明专利申请

(10) 申请公布号 CN 112398935 A

(43) 申请公布日 2021.02.23

(21) 申请号 202011224752.8

(22) 申请日 2020.11.05

(71) 申请人 广发证券股份有限公司

地址 510000 广东省广州市黄埔区中新广
州知识城腾飞一街2号618室

(72) 发明人 胡冠睿 谈加虎

(74) 专利代理机构 深圳市卓科知识产权代理有
限公司 44534

代理人 赵辉丽 潘晓

(51) Int. Cl.

H04L 29/08 (2006.01)

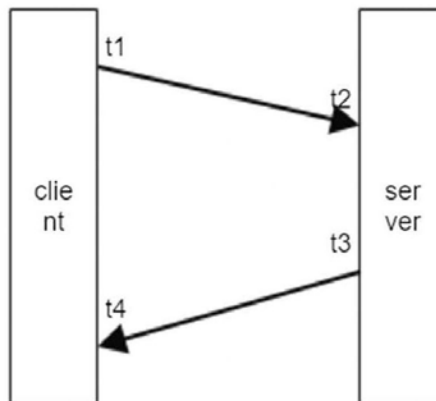
权利要求书1页 说明书4页 附图1页

(54) 发明名称

一种基于时延的全链路负载均衡算法

(57) 摘要

本发明公开了一种基于时延的全链路负载均衡算法,包括后端服务与客户端,本发明提供一套基于时延的全链路负载均衡算法,在发送请求时基于时延而非资源使用率来考虑,从而降低整个系统的平均延时,并提高系统的响应速度。



1. 一种基于时延的全链路负载均衡算法,其特征在于:包括后端服务与客户端;
其中,客户端包括:

- (一) 客户端在初次上线的时候,拉取交易前置列表;
- (二) 客户端对交易前置进行测速,计算网络延时,并获取前置服务器的处理延时;
- (三) 客户端计算链路延时,按延时排序,选取延时最低的交易前置连接,开始交易;

其中,后端服务包括:

(一) 对于inbound的请求,按协议号记录单位时间1个RTT的累计处理延时,以及累计处理次数;

(二) 服务端定时对于所有inbound的请求,按等权的算法计算inbound平均处理延时,以及单个协议的平均处理延时;

(三) 服务端对于outbound的请求,按协议号和目标服务器为二元组计算单位时间内1个RTT的累计处理延时,和累计处理次数;

(四) 服务端定时算出每个outbound的平均处理延时;

(五) 服务端收到客户端延时测速请求,将自己的平均inbound处理延时发送给客户端;

(六) 服务端收到客户端的处理请求,根据协议号,找到outbound处理延时最低的节点,将请求转发出去;

(七) 对于服务端之间请求超时的情况,需要做累加基数,如果单位时间内累计超过阈值,认为该路径不稳定,可以先将其踢出路由,等稳定以后,再加入。

2. 根据权利要求1所述的基于时延的全链路负载均衡算法,其特征在于:所述客户端计算链路延时包括网络延时以及处理延时。

一种基于时延的全链路负载均衡算法

技术领域

[0001] 本发明涉及一种负载均衡算法,具体涉及一种基于时延的全链路负载均衡算法。

背景技术

[0002] 负载均衡技术是一种针对海量请求在高并发高吞吐的情况下,通过将请求按照一定算法分配到不同节点分而治之,从而解除单机硬件处理能力限制,使得服务弹性伸缩成为可能。

[0003] 在证券行业里,对于交易业务,除了安全可靠,一个重要的指标就是“快”,即:如何快速的将委托发出,如何快速的将成交返回,在终端展示。“快”也成为衡量券商技术水平的一个核心业务指标。从某种意义上使用“端到端延迟”这一指标可以表征业务是否足够快速。

[0004] 从端到端的角度来讲,一条委托到发送到交易所需要经过如下路径:

[0005] 1. 客户端到交易前置;

[0006] 2. 交易前置转发请求至交易机房;

[0007] 3. 交易服务将请求转发给柜台服务;

[0008] 4. 柜台服务将请求发送到报盘服务;

[0009] 5. 报盘服务发送到交易所;

[0010] 其中步骤3、4一般是部署在同一交易网段,其延时在整条交易链路里,占比较小

[0011] 步骤5一般采用特定线路,延时相对固定。所以整个算法着眼点在于优化步骤1、2的处理延时。

[0012] 现有技术的不足:

[0013] 目前业内常用的负载均衡算法包括:

[0014] RR:即Round-Robin,请求按照轮询顺序依次发送到目标服务;

[0015] LRU:即Least recently used,请求有限发送给当前负载最轻的服务;

[0016] 以及上述各种算法的含权变体算法。

发明内容

[0017] 本发明所要解决的技术问题是一种基于时延的全链路负载均衡算法,在发送请求时基于时延而非资源使用率来考虑,从而降低整个系统的平均延时,并提高系统的响应速度。

[0018] 本发明是通过以下技术方案来实现的:一种基于时延的全链路负载均衡算法,包括后端服务与客户端;

[0019] 其中,客户端包括:

[0020] (一) 客户端在初次上线的时候,拉取交易前置列表;

[0021] (二) 客户端对交易前置进行测速,计算网络延时,并获取前置服务器的处理延时;

[0022] (三) 客户端计算链路延时,按延时排序,选取延时最低的交易前置连接,开始交

易；

[0023] 其中,后端服务包括:

[0024] (一)对于inbound的请求,按协议号记录单位时间1个RTT的累计处理延时,以及累计处理次数;

[0025] (二)服务端定时对于所有inbound的请求,按等权的算法计算inbound平均处理延时,以及单个协议的平均处理延时;

[0026] (三)服务端对于outbound的请求,按协议号和目标服务器为二元组计算单位时间内1个RTT的累计处理延时,和累计处理次数;

[0027] (四)服务端定时算出每个outbound的平均处理延时;

[0028] (五)服务端收到客户端延时测速请求,将自己的平均inbound处理延时发送给客户端;

[0029] (六)服务端收到客户端的处理请求,根据协议号,找到outbound处理延时最低的节点,将请求转发出去;

[0030] (七)对于服务端之间请求超时的情况,需要做累加基数,如果单位时间呢累计超过阈值,认为该路径不稳定,可以先将其踢出路由,等稳定以后,再加入。

[0031] 作为优选的技术方案,所述客户端计算链路延时包括网络延时以及处理延时。

[0032] 本发明的有益效果是:本发明提供一套基于时延的全链路负载均衡算法,在发送请求时基于时延而非资源使用率来考虑,从而降低整个系统的平均延时,并提高系统的响应速度。

附图说明

[0033] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0034] 图1为本发明的网络延时计算方法的流程图;

[0035] 图2为本发明的服务端inbound处理延时计算流程图;

[0036] 图3为本发明的服务端outbound处理延时计算流程图。

具体实施方式

[0037] 本说明书中公开的所有特征,或公开的所有方法或过程中的步骤,除了互相排斥的特征和/或步骤以外,均可以以任何方式组合。

[0038] 本说明书(包括任何附加权利要求、摘要和附图)中公开的任一特征,除非特别叙述,均可被其他等效或具有类似目的的替代特征加以替换。即,除非特别叙述,每个特征只是一系列等效或类似特征中的一个例子而已。

[0039] 在本发明的描述中,需要理解的是,术语“一端”、“另一端”、“外侧”、“上”、“内侧”、“水平”、“同轴”、“中央”、“端部”、“长度”、“外端”等指示的方位或位置关系为基于附图所示的方位或位置关系,仅是为了便于描述本发明和简化描述,而不是指示或暗示所指的装置或元件必须具有特定的方位、以特定的方位构造和操作,因此不能理解为对本发明的限制。

[0040] 此外,在本发明的描述中,“多个”的含义是至少两个,例如两个,三个等,除非另有明确具体的限定。

[0041] 本发明使用的例如“上”、“上方”、“下”、“下方”等表示空间相对位置的术语是出于便于说明的目的来描述如附图中所示的一个单元或特征相对于另一个单元或特征的关系。空间相对位置的术语可以旨在包括设备在使用或工作中除了图中所示方位以外的不同方位。例如,如果将图中的设备翻转,则被描述为位于其他单元或特征“下方”或“之下”的单元将位于其他单元或特征“上方”。因此,示例性术语“下方”可以囊括上方和下方这两种方位。设备可以以其他方式被定向(旋转90度或其他朝向),并相应地解释本文使用的与空间相关的描述语

[0042] 在本发明中,除非另有明确的规定和限定,术语“设置”、“套接”、“连接”、“贯穿”、“插接”等术语应做广义理解,例如,可以是固定连接,也可以是可拆卸连接,或成一体;可以是机械连接,也可以是电连接;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通或两个元件的相互作用关系,除非另有明确的限定。对于本领域的普通技术人员而言,可以根据具体情况理解上述术语在本发明中的具体含义。

[0043] 如图1所示,本发明的一种基于时延的全链路负载均衡算法,包括后端服务与客户端;

[0044] 其中,客户端包括:(一)客户端在初次上线的时候,拉取交易前置列表;(二)客户端对交易前置进行测速,计算网络延时,并获取前置服务器的处理延时;(三)客户端计算链路延时,按延时排序,选取延时最低的交易前置连接,开始交易;

[0045] 其中,后端服务包括:

[0046] (一)对于inbound的请求,按协议号记录单位时间1个RTT的累计处理延时,以及累计处理次数;(二)服务端定时对于所有inbound的请求,按等权的算法计算inbound平均处理延时,以及单个协议的平均处理延时;(三)服务端对于outbound的请求,按协议号和目标服务器为二元组计算单位时间内1个RTT的累计处理延时,和累计处理次数;(四)服务端定时算出每个outbound的平均处理延时;(五)服务端收到客户端延时测速请求,将自己的平均inbound处理延时发送给客户端;(六)服务端收到客户端的处理请求,根据协议号,找到outbound处理延时最低的节点,将请求转发出去;(七)对于服务端之间请求超时的情况,需要做累加基数,如果单位时间呢累计超过阈值,认为该路径不稳定,可以先将其踢出路由,等稳定以后,再加入。

[0047] 其中,客户端计算链路延时包括网络延时以及处理延时。

[0048] 如图1所示,网络延时计算方法如下:其中,客户端(client)向交易前置(server)于t1时刻发送一个测速数据包,包含时间戳t1,server在t2时刻收到,并于t3时刻返回测速数据包,包含(t1,t2,t3),客户端于t4时刻收到这个测试包,那么延时计算为:

$$[0049] \quad latency = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$$

[0050] 其中latency可以按照0.2Hz的频率计算,以30s为窗口,计算窗口内的网络延时。

[0051] 本发明的有益效果是:本发明提供一套基于时延的全链路负载均衡算法,在发送请求时基于时延而非资源使用率来考虑,从而降低整个系统的平均延时,并提高系统的响

应速度。

[0052] 服务端inbound处理延时计算如图2所示：

[0053] 延时采集：

[0054] 收到客户端请求，记录时间戳，将请求发送给转发逻辑，如果收到回包，则累加inbound处理延时，并更新处理次数，如果没有收到回包，则判断是否超时，如果是，则设置处理延时为超时时间，如果否，则返回到上级。

[0055] 延时计算：

[0056] 是否计算延时，如果是，则计算inbound处理延时，重置延时统计数据，如果否，则进入休眠。

[0057] 服务端outbound处理延时计算如图3所示：

[0058] 延时采集：

[0059] 根据协议号，记录时间戳，如果收到回包，则累加协议的处理时延，处理次数，如果否，则判断是否超时，如果超时，则设置延时时间为超时时间，如果不超时，则返回上级。

[0060] 延时计算：

[0061] 是否计算延时，如果是，则计算outbound处理延时，重置延时统计数据，如果否，则进入休眠。

[0062] 以上所述，仅为本发明的具体实施方式，但本发明的保护范围并不局限于此，任何不经过创造性劳动想到的变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应该以权利要求书所限定的保护范围为准。

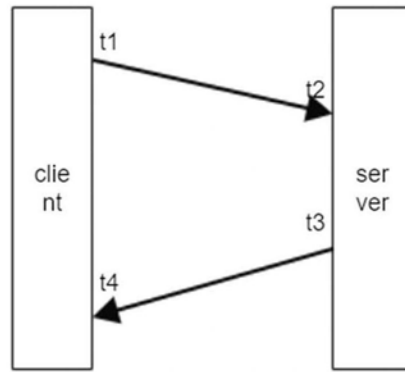


图1

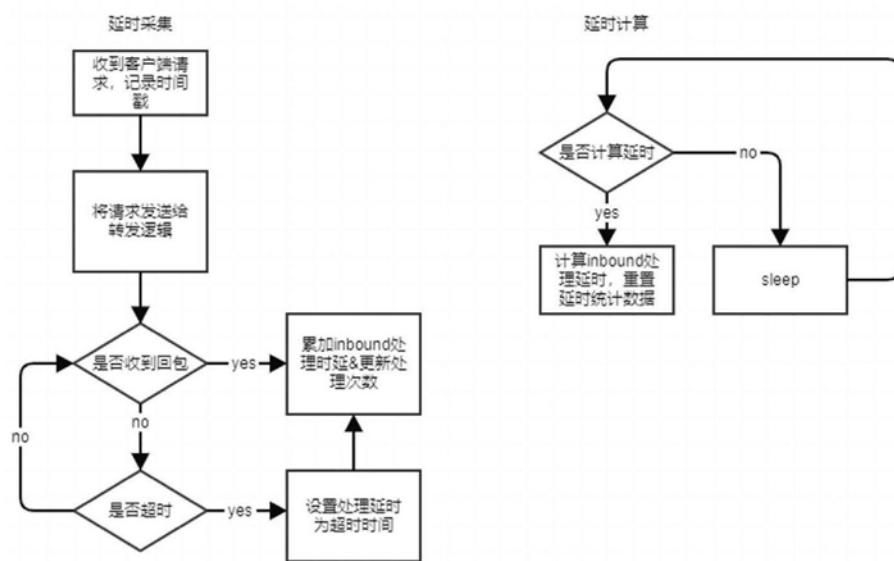


图2

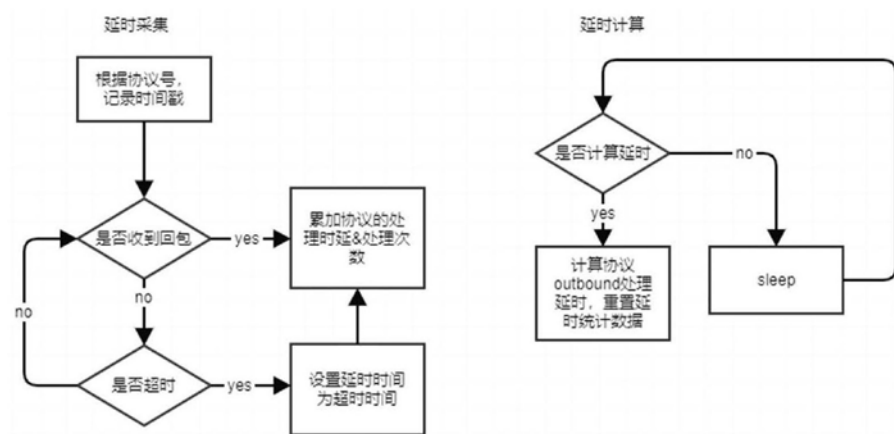


图3