

# 基于负载反馈的分布式数字集群 动态负载均衡算法\*

曲乾聪, 王 俊

(北京交通大学 电子信息工程学院, 北京 100044)

**摘要:** 针对传统负载均衡算法不能满足公网数字集群系统高并发用户请求和快速呼叫建立等需求, 提出一种基于负载反馈的分布式数字集群动态负载均衡算法, 实现公网数字集群系统负载均衡, 提高用户容量。首先建立参与 MCPTT 服务器的静态负载和动态负载监控机制和指标; 然后利用加权轮询算法为用户分配参与 MCPTT 服务器, 并通过用户请求的处理获得复合负载参数; 根据负载指标的反馈更新参与 MCPTT 服务器权值以动态调整服务器负载。仿真结果表明, 该算法的负载均衡效果优于传统算法和其他动态反馈算法, 负载均衡度更小、用户请求响应延迟更低。

**关键词:** 分布式数字集群; 动态负载均衡; 动态权值; 负载反馈

**中图分类号:** TP393.07

**文献标志码:** A

**文章编号:** 1001-3695(2022)02-034-0526-05

doi:10.19734/j.issn.1001-3695.2021.07.0280

## Dynamic load balancing algorithm based on load feedback for distributed digital trunk

Qu Qiancong, Wang Jun

(School of Electronic & Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** Traditional load balancing algorithms can't meet the demand of high concurrent requests and quick calling of public network digital trunk. This paper proposed a dynamic load balancing algorithm based on load feedback for distributed digital trunk to achieve load balancing and increase user capacity. Firstly, this paper established monitoring method and monitoring indexes for static and dynamic load of participating MCPTT server. Then, it used weighted round robin algorithm to allocate participating MCPTT server to users and got complex load parameters by dealing user requests. Finally, it updated the weight of participating MCPTT server based on load feedback to dynamically adjust server load. Simulation results show that the proposed algorithm has better load balancing effect, smaller load balancing degree and lower response delay compared with traditional algorithms and other dynamic feedback algorithms.

**Key words:** distributed digital trunk; dynamic load balancing; dynamic weight; load feedback

## 0 引言

公网数字集群是一种专业指挥调度系统, 应用于政府、公安、交通等部门<sup>[1]</sup>。分布式公网数字集群通过合理分配用户登录和呼叫请求至多台服务器, 实现负载均衡, 提高用户容量。

分布式公网数字集群系统可通过高效的负载均衡算法实现高并发的用户请求。负载均衡算法分为静态负载均衡算法和动态负载均衡算法。静态负载均衡算法在负载均衡服务器中预先配置服务器分配策略, 使用固定参数进行计算与分配, 常见的静态负载均衡算法有随机算法、简单轮询算法、加权轮询算法等<sup>[2]</sup>; 动态负载均衡算法收集服务器的性能指标, 通过某种数学模型计算来动态分配用户请求, 常见的动态负载均衡算法有最小连接数算法、加权最小连接数算法、最快响应速度算法等<sup>[2]</sup>。根据分布式公网数字集群系统组呼特点, 用户的呼叫业务量不同, 导致不同参与 MCPTT 服务器的负载不同。静态负载均衡算法和动态负载均衡算法以用户数作为负载单位, 未充分考虑在线用户的业务量差异, 服务器的实际负载可能会严重失衡<sup>[3]</sup>。因此, 分布式公网数字集群的负载均衡算法需要以服务器 CPU、内存、网络带宽等实际负载为评价指标

并动态调整服务器分配策略。

国内外提出了多种以服务器实际性能和负载为评价指标的负载均衡算法, 动态调整服务器分配策略。文献[4]提出了一种基于最小连接数算法改进的动态负载均衡算法 ECON, 该算法在根据节点响应时间和连接数计算复合负载后, 根据服务器权重与复合负载的比值选择最优节点, 降低了高并发请求下的响应时间和失败率, 但选择单一节点的机制在处理高并发请求时会造成请求的堆积。文献[5]提出了一种基于排队论的动态负反馈负载均衡算法 DFQT, 将服务器节点视为 M/M/1 无限资源等待系统模型, 综合考虑节点的承载能力和当前等待队列的长度, 并通过计算节点的期望请求到达率为节点分配相应权重的请求, 降低了高并发请求下的响应时间、提高了吞吐量, 但该算法假设请求到达速率和处理时间是相互独立的, 忽略了服务器节点在高负载情况下处理时间的增加。

为了克服选择单一节点请求堆积和高负载下处理时间增加的问题, 满足分布式公网数字集群负载均衡的需求, 本文提出一种用于分布式公网数字集群的自适应负载均衡算法——基于负载反馈的加权轮询算法。该算法建立参与 MCPTT 服务器的静态负载和动态负载监控机制和指标, 利用加权轮询算法为用户分配服务器, 并通过用户请求的处理获得复合负载参

收稿日期: 2021-07-12; 修回日期: 2021-09-02 基金项目: 国家重点研发计划资助项目(2018YFB1601504)

作者简介: 曲乾聪(1996-), 男, 河北张家口人, 硕士研究生, 主要研究方向为无线通信(19120111@bjtu.edu.cn); 王俊(1982-), 男, 江苏淮安人, 讲师, 博士(后), 主要研究方向为专业移动通信及其数字信号处理。

数,最后通过服务器负载指标反馈的方式更新参与 MCPTT 服务器权值,动态调整服务器负载,实现分布式公网数字集群系统的负载均衡,提高用户容量<sup>[6-8]</sup>。

## 1 分布式公网数字集群系统

分布式公网数字集群系统由负载均衡服务器、参与 MCPTT 服务器集群、控制 MCPTT 服务器集群、媒体服务器集群、数据库集群和 Zookeeper 集群组成。分布式公网数字集群系统的架构如图 1 所示。

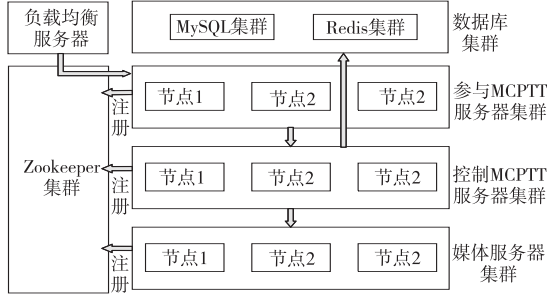


图1 分布式公网数字集群系统架构

Fig. 1 Architecture of distributed digital trunking system

分布式公网数字集群系统依赖 Zookeeper 集群实现分布式服务协调。参与 MCPTT 服务器集群、控制 MCPTT 服务器集群和媒体服务器集群中每个节点启动后注册在 Zookeeper 集群中,并与 Zookeeper 集群维持链路监控,实现 Zookeeper 集群对所有服务器信息的维护。参与 MCPTT 服务器向 Zookeeper 集群订阅控制 MCPTT 服务器集群的节点信息,用于实现参与 MCPTT 服务器节点对控制 MCPTT 服务器节点的服务调用和负载均衡。控制 MCPTT 服务器向 Zookeeper 集群订阅媒体服务器集群的节点信息,用于实现控制 MCPTT 服务器节点对媒体服务器节点的服务调用和负载均衡<sup>[9]</sup>。

用户通过负载均衡服务器接入分布式公网数字集群系统。负载均衡服务器获取参与 MCPTT 服务器集群的节点信息,为用户分配参与 MCPTT 服务器,实现负载均衡<sup>[10]</sup>。

## 2 基于负载反馈的加权轮询算法

### 2.1 加权轮询算法

加权轮询算法用服务器权值表示服务器之间 CPU、内存、网络带宽等性能的差异,性能强的服务器权值高,分配更多请求。通过预先设置服务器权值,使服务器根据处理能力接收请求,实现负载均衡<sup>[11,12]</sup>。加权轮询算法服务器权值的设置过程中,仅考虑了服务器的初始承载能力,忽略了系统运行过程中用户业务量的差异<sup>[13]</sup>。

分布式公网数字集群用户登录过程中,在线用户会随机发起呼叫,不同服务器用户业务量不同。在分布式公网数字集群中,可以根据 CPU、内存、网络带宽等性能为参与 MCPTT 服务器预先配置权值,实现用户连接数的负载均衡,但是服务器实际负载与权值不匹配,导致负载严重失衡。因此,加权轮询算法无法有效实现分布式公网数字集群的负载均衡。

### 2.2 基于负载反馈的加权轮询算法

#### 2.2.1 算法模型

基于加权轮询算法忽略了用户业务量差异,导致服务器负载失衡的问题,为了增强分布式公网数字集群的负载自适应能力,提出一种用于分布式公网数字集群的动态负载均衡算法——基于负载反馈的加权轮询算法,其算法模型如图 2 所示。

当用户登录时,负载均衡服务器计算参与 MCPTT 服务器权值,执行加权轮询,为用户分配参与 MCPTT 服务器。在登录

后,用户会随机发起呼叫,用户的呼叫需求不同,所以不同参与 MCPTT 服务器的实际负载不同。在处理用户请求之后,计算参与 MCPTT 服务器负载指标。负载均衡服务器按照一定周期根据服务器负载指标重新计算权值。通过负载反馈机制,以参与 MCPTT 服务器负载指标动态调整分配策略,适应负载的变化,使服务器负载趋于均衡<sup>[14,15]</sup>。

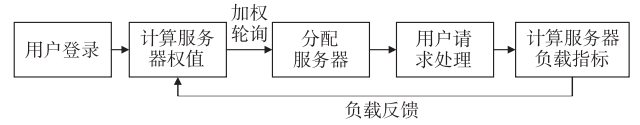


图2 基于负载反馈的加权轮询算法模型

Fig. 2 Algorithm model of weighted round robin based on load feedback

#### 2.2.2 服务器负载指标与服务器权值

1) 服务器负载指标 服务器负载指标由静态负载因素和动态负载因素组成。静态负载因素定义为服务器的初始承载能力,即服务器空闲时的性能;动态负载因素定义为服务器运行过程中的剩余承载能力<sup>[16]</sup>。

(1) 静态负载因素 假设服务器的平均 CPU 频率  $C_{avg}$ 、平均内存容量  $M_{avg}$ 、平均网络带宽  $N_{avg}$  分别为

$$C_{avg} = \frac{1}{n} \sum_{i=1}^n c_i, M_{avg} = \frac{1}{n} \sum_{i=1}^n m_i, N_{avg} = \frac{1}{n} \sum_{i=1}^n n_i \quad (1)$$

其中: $c_i$ 、 $m_i$ 、 $n_i$  分别为服务器  $i$  的 CPU 频率、内存容量和网络带宽; $n$  为服务器总数。服务器静态负载因素  $SC_i$  为

$$SC_i = \alpha_c \frac{c_i}{C_{avg}} + \alpha_m \frac{m_i}{M_{avg}} + \alpha_n \frac{n_i}{N_{avg}} \text{ 且 } \alpha_c + \alpha_m + \alpha_n = 1 \quad (2)$$

其中: $c_i$ 、 $m_i$ 、 $n_i$  分别为服务器  $i$  的 CPU 频率、内存容量和网络带宽; $n$  为服务器总数; $\alpha_c$ 、 $\alpha_m$ 、 $\alpha_n$  分别为 CPU 频率、内存容量和网络带宽的影响程度。

(2) 动态负载因素 服务器动态负载因素反映了服务器的实际负载情况。服务器的资源占用负载  $RO_i$  为

$$RO_i = \alpha_c L_{c_i} + \alpha_m L_{m_i} + \alpha_n L_{n_i} \text{ 且 } \alpha_c + \alpha_m + \alpha_n = 1 \quad (3)$$

其中: $L_{c_i}$ 、 $L_{m_i}$ 、 $L_{n_i}$  分别为服务器  $i$  的 CPU 占用率、内存占用率和网络带宽占用率; $\alpha_c$ 、 $\alpha_m$ 、 $\alpha_n$  分别为 CPU 占用率、内存占用率和网络带宽占用率的影响程度。

服务器资源占用率是服务器负载的宏观指标,无法直接反映参与 MCPTT 服务器进程的剩余处理能力,需要用三个指标反映剩余处理能力:a) 用户已登录分布式公网数字集群系统的数量(即用户连接数)反映服务器进程的繁忙程度;b) 用户请求处理延迟定义为用户发起请求与服务器返回响应的时间差,是服务器动态负载的直接体现,负载越高,用户请求处理时延越高;c) 待处理请求数量反映服务器的负载,队列越长,服务器的响应延迟越高。将用户连接数、用户请求处理延迟和待处理请求数量加权求和作为复合负载<sup>[17,18]</sup>。服务器的复合负载  $CL_i$  为

$$CL_i = \alpha_{con} N_{con} + \alpha_t N_t + \alpha_q N_q \text{ 且 } \alpha_{con} + \alpha_t + \alpha_q = 1 \quad (4)$$

其中: $N_{con}$ 、 $N_t$ 、 $N_q$  分别为服务器  $i$  的用户连接数、用户请求处理时延和待处理请求数量; $\alpha_{con}$ 、 $\alpha_t$ 、 $\alpha_q$  分别为用户连接数、用户请求处理时延和待处理请求数量的影响程度。

服务器动态负载因素  $DL_i$  为

$$CL_i = \beta_r RO_i + \beta_c CL_i \text{ 且 } \beta_r + \beta_c = 1 \quad (5)$$

其中: $RO_i$  为服务器的资源占用负载; $CL_i$  为服务器  $i$  的复合负载; $\beta_r$ 、 $\beta_c$  分别为资源占用负载和复合负载的影响程度。

(3) 负载指标 服务器负载指标  $L_i$  为

$$L_i = \frac{\gamma_1 DL_i}{\gamma_2 SC_i} \text{ 且 } \gamma_1 + \gamma_2 = 1 \quad (6)$$

其中: $SC_i$  为服务器  $i$  的静态负载因素; $DL_i$  为服务器  $i$  的动态负载因素; $\gamma_1$ 、 $\gamma_2$  分别为动态负载因素和静态负载因素的影响程度。

服务器负载指标与静态负载因素成反比,与动态负载因素成正比,即静态负载因素越大,服务器的初始承载能力越强,相应的当前负载越低,动态负载因素越大,服务器的负载越重,实际剩余承载能力越少。

2) 服务器权值 服务器权值  $\omega_i$  为

$$\omega_i = \frac{1}{L_i} / \sum_{k=1}^n \frac{1}{L_k} \quad (7)$$

其中:  $L_i$  为服务器的负载指标;  $n$  为服务器总数。

为了使算法能根据实时负载动态改变服务器处理用户请求的数量,实现负载自适应,权值应该能够反映服务器当前的请求处理能力。根据服务器权值定义,服务器权值之比为服务器负载指标倒数之比,实时负载高的服务器权值占比较低,处理更少请求;实时负载低的服务器权值占比较高,处理更多请求<sup>[19]</sup>。

### 2.2.3 算法描述

基于负载反馈的加权轮询算法的基本思想是:在服务器处理请求的过程中记录服务器负载指标,并根据负载动态改变服务器权值以实现负载自适应。动态权值使负载高的服务器接下来处理请求次数变少、负载低的服务器接下来处理请求次数变多,保证服务器在一个轮询周期中处理请求的次数与权值成正比,与负载成反比,并且不会连续处理多次请求,防止请求产生堆积<sup>[20]</sup>。负载均衡服务器根据参与 MCPTT 服务器运行过程中的负载指标实时更新服务器权值,负反馈调整服务器分配策略。基于负载反馈的加权轮询算法的描述如下:

a) 负载指标和权值初始化。根据式(2)~(6)初始化服务器负载指标  $L_i$  和权值  $\omega_i$ ,并为每台服务器设置临时权值  $currentWeight$  为 0。

b) 参与 MCPTT 服务器分配。当接收到用户登录请求时,负载均衡服务器为用户分配参与 MCPTT 服务器的流程如下:

(a) 计算真实权值之和,服务器节点的真实权值之和  $totalWeight = \sum_{i=1}^n \omega_i$ 。其中,  $\omega_i$  为服务器  $i$  的权值,  $n$  为服务器总数。

(b) 给每个服务器的临时权值加上服务器真实权值,服务器  $i$  的临时权值  $currentWeight_i = currentWeight_i + \omega_i$ 。其中,  $\omega_i$  为服务器  $i$  的权值。

(c) 取  $currentWeight$  最大的服务器为本次分配服务器,  $currentWeight_{max} = \max \{ currentWeight_i \}$ 。其中,  $currentWeight_i$  为服务器  $i$  的临时权值。

(d) 本次分配服务器的临时权值减去总权值,本次分配服务器的临时权值  $currentWeight_{ass} = currentWeight_{max} - totalWeight$ 。其中,  $currentWeight_{max}$  为临时权值的最大值,  $totalWeight$  为服务器真实权值之和。

(e) 更新本次分配服务器的待处理请求数,本次分配服务器的待处理请求数  $N_q = N_q + 1$ 。

c) 用户请求处理。当处理用户请求时,参与 MCPTT 服务器需要更新服务器的复合负载参数。服务器待处理请求数  $N_q = N_q + 1$ ; 服务器请求处理时延  $N_t = T_{end} - T_{start}$ 。其中,  $T_{start}$  为请求接收时间,  $T_{end}$  为请求处理完成时间。服务器用户连接数  $N_{con} = N_{con} + 1$ 。

d) 服务器权值更新。当权值更新定时器到期时,负载均衡服务器根据式(2)~(6)重新计算服务器负载指标  $L_i$ ,更新权值  $\omega_i$ ,并设置临时权值  $currentWeight$  为 0。

e) 循环。按照服务器当前权值,循环步骤 b)~d)。

### 2.2.4 算法实现流程

基于负载反馈的加权轮询算法的流程如图 3 所示。当系统启动时,初始化参与 MCPTT 服务器的负载指标并计算服务器权值;当接收到用户的接入请求时,负载均衡服务器以当前服务器权值执行加权轮询,分配服务器并通知用户;用户向指定参与 MCPTT 服务器发送登录请求,参与 MCPTT 服务器将请

求存入待处理队列中,并更新待处理请求数;当处理登录请求时,参与 MCPTT 服务器更新待处理请求数,记录本次请求处理时延并更新用户连接数;负载均衡服务器设置权值更新定时器,当定时器到期时,计算参与 MCPTT 服务器的负载指标并更新服务器权值。

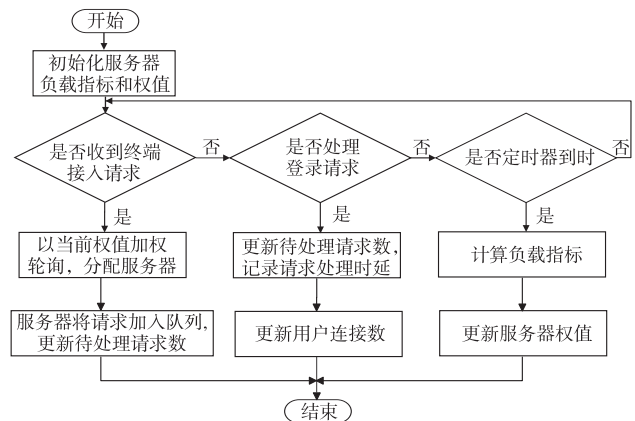


图 3 基于负载反馈的加权轮询算法的流程

Fig. 3 Process of weighted round robin based on load feedback algorithm

## 3 仿真实验

### 3.1 验证环境

为了完成算法验证,搭建分布式公网数字集群系统测试环境,基于负载反馈的加权轮询算法的验证环境如图 4 所示。

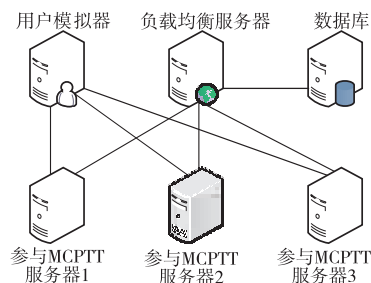


图 4 基于负载反馈的加权轮询算法的验证环境

Fig. 4 Verification environment of weighted round robin based on load feedback algorithm

用户模拟器可以模拟多个分布式公网数字集群用户发送并发登录和呼叫请求。负载均衡服务器负责算法的执行,在接收到用户的连接请求后,根据数据库中参与 MCPTT 服务器权值执行加权轮询,为用户分配服务器,同时负责定期计算参与 MCPTT 服务器负载指标,更新服务器权值并存入数据库。参与 MCPTT 服务器接收、处理用户的登录和呼叫请求,并记录用户连接数、请求处理时延、待处理请求数量,定期存入数据库中。服务器硬件配置如表 1 所示。

表 1 服务器硬件配置

Tab. 1 Hardware configuration of servers

服务器	CPU 频率/GHz	内存容量/GB	网络带宽/ Mbps
用户模拟器	1	16	8
负载均衡服务器	1	16	8
参与 MCPTT 服务器 1	2.5	4	2
参与 MCPTT 服务器 2	2.5	4	3
参与 MCPTT 服务器 3	2.5	2	1

### 3.2 验证指标

采用负载均衡效果、负载均衡度和用户登录时延指标验证基于负载反馈的加权轮询算法的性能。

a) 负载均衡效果。负载均衡算法的目标是通过合理分配请求,使资源不同的服务器之间负载保持平衡。服务器中待处



理请求数量反映了服务器当前的繁忙程度,将待处理队列长度定义为服务器负载,队列越长,服务器的响应延迟越高。通过观察服务器待处理请求队列长度随时间变化的趋势评价不同算法的负载均衡效果。

b)负载均衡度。其为服务器负载的偏差程度,定义为服务器负载的均方差。服务器负载均衡度 $\varepsilon$ 定义为

$$\varepsilon = \sqrt{\frac{1}{n} \sum_{i=1}^n (L_i - L_{avg})^2} \quad (8)$$

其中: $L_i$ 为服务器*i*的负载指标; $L_{avg}$ 为服务器的平均负载指标; $n$ 为系统中服务器的数量。以负载均衡度定量评价服务器之间的负载均衡程度,比较不同负载均衡算法的性能。负载均衡度越小,服务器负载的偏差越小,负载均衡程度越好。

c)用户登录时延。负载均衡的最终目标是用户体验的提升,也就是请求响应时延的降低。以用户并发登录时延作为负载均衡算法的评价数据,当在线用户随机呼叫时,记录每个用户的登录时延,并统计最短登录时延、最长登录时延和平均登录时延,比较不同算法对请求响应时延的改善。

3.3 验证方法与流程

1)验证方法 分别编写参与MCPTT服务器模拟程序、用户模拟器程序以及负载均衡服务器。参与MCPTT服务器模拟程序启动三个线程,分别是登录接收线程、登录处理线程以及呼叫模拟线程。当收到来自负载均衡服务器的用户登录请求时,登录接收线程将用户请求加入待处理队列中,记录接收时间并更新待处理请求数。登录处理线程模拟处理登录的过程,每条登录请求处理200 ms,处理完成后记录处理完成时间,计算用户请求响应时延并更新待处理请求数和用户连接数。呼叫模拟线程模拟在线用户随机发起呼叫,随机呼叫规则为:用户连接数小于10时系统中无呼叫;用户连接数大于10时,每隔一段随机时间,系统中产生1次并发呼叫,时间间隔的随机范围为10~20 s;每次并发呼叫的用户数是随机的,随机范围为5~(5+用户连接数/10);每条呼叫请求处理为300 ms。

用户模拟器程序每60 ms向负载均衡服务器发送一条用户登录请求。当接收到用户的登录请求后,负载均衡服务器根据参与MCPTT服务器权值执行加权轮询,分配服务器并通知用户;负载均衡服务器每隔2 s计算参与MCPTT服务器负载指标并更新权值。

2)验证流程 基于负载反馈的加权轮询算法的仿真验证流程如下:a)建立用户模拟器程序与负载均衡服务器的连接,建立负载均衡服务器与参与MCPTT服务器的连接;b)用户模拟器程序每60 ms发送一条登录请求,并根据测试需要选择并发请求数量;c)测试过程中持续记录每台参与MCPTT服务器的负载、负载均衡度以及每条登录请求的响应时延;d)统计数据,比较不同算法的性能。

3.4 验证结果

算法的仿真参数如表2所示。

表2 算法的仿真参数  
Tab.2 Simulation parameters of algorithm

参数	值	参数	值
参与MCPTT服务器数	3	并发登录间隔	60 ms
登录处理时间	0.2 s	随机呼叫间隔	10~20 s
呼叫处理时间	0.5 s	权值更新周期	3 s

分别采用简单轮询、最小连接数两种传统算法,ECON<sup>[4]</sup>、DFQT<sup>[5]</sup>两种动态反馈算法以及本文提出的基于负载反馈的加权轮询算法,从负载均衡效果、负载均衡度以及用户登录时延对算法进行评价,对比不同算法的负载均衡性能。

1)负载均衡效果 用户模拟器每60 ms发送一条登录请求,1 min向负载均衡服务器发送1 000条登录请求。在1 000

个用户登录期间,已登录用户会随机发起呼叫。记录参与MCPTT服务器1 min内的负载变化,抽样周期为1 s。简单轮询算法、最小连接数算法、ECON算法、DFQT算法和本文的基于负载反馈的加权轮询算法的负载曲线如图5所示。

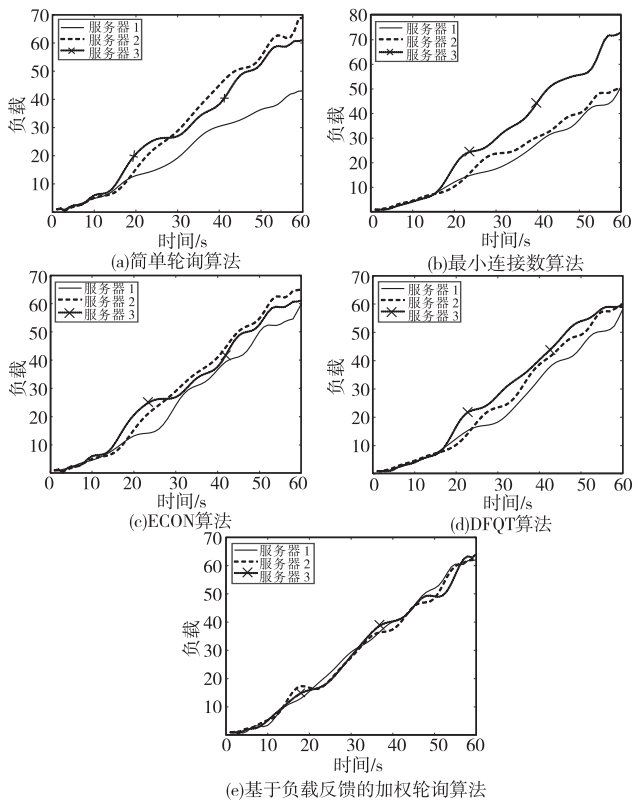


图5 对比算法的负载曲线  
Fig.5 Load curves of different algorithms

随着时间的推移,并发用户数量增加,所有算法的负载不断增长。在简单轮询算法和最小连接数算法下,当服务器中已连接用户发起呼叫时,服务器之间的负载发生偏离,随着呼叫次数的增加,偏差逐渐增大。由于简单轮询算法和最小连接数算法仅考虑服务器的用户连接数,未考虑实际负载,所以当服务器中已登录用户发起呼叫时,服务器的实际负载失去平衡。

两种动态反馈算法ECON和DFQT以服务器的实际负载情况反馈调整了分配策略,可以在负载偏离时进行调整,因此负载均衡效果优于传统算法。ECON算法由于始终将请求分配至最优节点,当服务器QPS较高时最优节点会发生请求堆积,导致动态调整迟滞,负载波动较大。DFQT算法假设请求到达速率和处理时间相互独立,未充分考虑服务器处理时间随负载的变化,因此请求分配比例误差较大,服务器负载间存在一定偏差。基于负载反馈的加权轮询算法以实际负载作为负反馈,每当呼叫产生,实际负载发生偏差时,服务器权值总是实时调整,使服务器的负载总是趋于平衡,具有负载自适应能力。根据负载均衡效果的测试数据可知,在基于负载反馈的加权轮询算法下,服务器之间的负载偏差最小,负载均衡效果最好,优于传统算法、ECON算法和DFQT算法。

2)负载均衡度 用户模拟器每60 ms发送一条登录请求,1 min向负载均衡服务器发送1 000条登录请求。在1 000个用户登录期间,已登录用户会随机发起呼叫。记录1 min内参与MCPTT服务器的负载均衡度,抽样周期为1 s。传统算法和本文算法的负载均衡度曲线如图6所示。ECON算法、DFQT算法和本文算法的负载均衡度曲线图如图7所示。

随着时间的推移,用户连接数增大,所有算法的负载均衡度均有所增加。简单轮询算法和最小连接数算法波动较大,最大负载均衡度超过了7,且随着用户连接数增大负载均衡度逐

渐增大。由于简单轮询算法和最小连接数算法以用户连接数为负载单位,未考虑用户呼叫业务量不同,每当呼叫产生就会产生较大的负载偏差,负载均衡度上升。ECON 和 DFQT 算法由于动态调整了分配策略,降低了服务器之间的负载偏差,负载均衡度有所改善。基于负载反馈的加权轮询算法的负载均衡度低于 ECON 和 DFQT 算法,负载均衡度围绕 2 波动,不超过 4。由于基于负载反馈的加权轮询算法以服务器的实际负载进行动态调度,每当呼叫产生,负载偏离时,负反馈机制会实时修正服务器权值使负载趋于平衡,资源的实时调度与分配更合理。

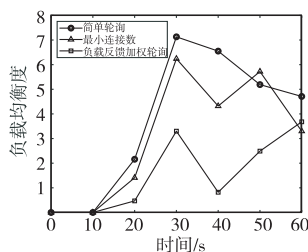


图6 传统算法和本文算法的负载均衡度曲线

Fig. 6 Load balancing degree curves of traditional algorithm and the proposed algorithm

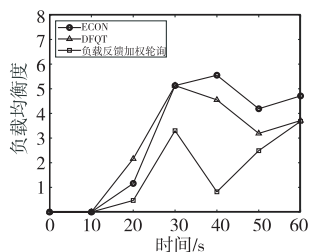


图7 ECON、DFQT 算法和本文算法的负载均衡度曲线

Fig. 7 Load balancing degree curves of ECON, DFQT and the proposed algorithm

所有算法的负载均衡度数据统计如表3所示。在1 min内,简单轮询算法和最小连接数算法的平均负载均衡度超过了4,且最大负载均衡度超过了7;ECON 和 DFQT 算法负载均衡度低于传统算法。基于负载反馈的加权轮询算法优于传统算法、ECON 和 DFQT 算法,平均负载均衡度为2,且最大负载均衡度不超过4。基于负载反馈的加权轮询算法的负载偏差更小,且负载均衡度更稳定,体现了资源的实时调度与分配更合理,资源分配越合理,服务器的整体资源利用率越高。

表3 三类算法的负载均衡度数据统计

Tab. 3 Load balancing degree statistics of three kinds of algorithms

算法	最大均衡度	平均均衡度
简单轮询	7.13	5.15
最小连接数	6.24	4.20
ECON	5.55	4.15
DFQT	5.13	3.74
负载反馈加权轮询	3.68	2.15

3) 用户登录时延 用户模拟器每60 ms发送一条登录请求,1 min向负载均衡服务器发送1 000条登录请求。在1 000个用户登录期间,已登录用户会随机发起呼叫。记录所有用户的登录请求响应时延,并统计最短登录时延、最长登录时延和平均登录时延。所有算法的用户登录响应时延如表4所示。

表4 三类算法的用户登录响应时延

Tab. 4 User login response delay of three kinds of algorithm /s

算法	平均时延	最短时延	最长时延
简单轮询	6.145	0.200	17.010
最小连接数	6.223	0.200	16.390
ECON	5.545	0.200	15.790
DFQT	5.796	0.200	14.230
负载反馈加权轮询	5.186	0.200	12.013

基于负载反馈的加权轮询算法在最长登录时延和平均登录时延上都小于传统算法、ECON 算法和 DFQT 算法。由于采用基于负载反馈的加权轮询算法进行负载均衡时,服务器的实际负载最均衡,资源分配合理,服务器响应更及时,使得在同样的服务器资源下,该算法下的用户登录时延更低、体验更好。

## 4 结束语

本文根据分布式公网数字集群系统的架构,分析了分布式

公网数字集群用户业务量不同的特点,提出了一种基于负载反馈的加权轮询算法。首先建立参与 MCPTT 服务器的静态负载和动态负载监控机制和指标;然后利用加权轮询算法为用户分配服务器,并通过用户请求的处理获得复合负载参数;最后通过服务器负载指标反馈的方式更新参与 MCPTT 服务器权值,动态调整服务器负载,实现负载均衡。本文对算法进行了仿真验证,验证结果表明在用户业务量不同的场景下,本文算法的负载均衡效果优于传统算法和其他动态反馈算法,负载均衡度更小,用户请求响应延迟更低。下一步研究方向是优化服务器负载指标和权值计算方法,降低参数计算带来的性能下降;研究权值更新周期与动态负载均衡性能的相关性,通过设置最优更新周期来提高负载均衡性能。

## 参考文献:

- [1] 孙昕,李海. 浅谈我国数字集群的发展[J]. 移动通信,2010,34(5):69-73. (Sun Xin, Li Hai. Talking about the development of digital cluster in China[J]. Mobile Communications, 2010, 34(5): 69-73.)
- [2] 毛可嘉. 客户端负载均衡算法研究及在即时通讯架构中的应用与实现[D]. 成都:电子科技大学,2018. (Mao Kejia. Research on client site load balancing algorithm and its application and implementation in instant messaging architecture[D]. Chengdu: University of Electronic Science & Technology of China, 2018.)
- [3] 李冬梅,施海虎. 负载平衡调度问题的一般模型研究[J]. 计算机工程与应用,2007,43(8):121-125. (Li Dongmei, Shi Haihu. Study on general load balancing scheduling model[J]. Computer Engineering and Applications, 2007, 43(8): 121-125.)
- [4] Zhu Liangshuai, Cui Jianming, Xiong Gaofeng. Improved dynamic load balancing algorithm based on least-connection scheduling[C]//Proc of the 4th IEEE Information Technology and Mechatronics Engineering Conference. Piscataway, NJ: IEEE Press, 2018: 1858-1862.
- [5] Wen Zepeng, Li Gongliang, Yang Guanghong. Research and realization of Nginx-based dynamic feedback load balancing algorithm[C]//Proc of the 3rd IEEE Advanced Information Technology, Electronic and Automation Control Conference. Piscataway, NJ: IEEE Press, 2018: 2541-2546.
- [6] Qin E, Wang Yuanli, Yuan Liping, et al. Research on Nginx dynamic load balancing algorithm[C]//Proc of the 12th International Conference on Measuring Technology and Mechatronics Automation. Washington DC: IEEE Computer Society, 2020: 620-624.
- [7] 吴俊鹏,刘晓东. 一种基于集群的动态负载均衡算法研究[J]. 电子设计工程,2021,29(16):75-78. (Wu Junpeng, Liu Xiaodong. Research on a dynamic load balancing algorithm based on cluster[J]. Electronic Design Engineering, 2021, 29(16): 75-78.)
- [8] 宋文文. 基于SDN的数据中心网络负载均衡研究与实现[D]. 南京:南京邮电大学,2020. (Song Wenwen. Research and implementation on load balancing of data center network based on SDN[D]. Nanjing: Nanjing University of Posts & Telecommunications, 2020.)
- [9] Jadhav K A, Mulla M M, Narayan D G. An efficient load balancing mechanism in software defined networks[C]//Proc of the 12th International Conference on Computational Intelligence and Communication Networks. Piscataway, NJ: IEEE Press, 2020: 116-122.
- [10] Kishor A, Niyogi R, Veeravalli B. Fairness-aware mechanism for load balancing in distributed systems[J]. IEEE Trans on Services Computing, 2020, DOI:10.1109/TSC. 2020. 3044104.
- [11] 李元洪. 基于反馈的动态负载均衡调度算法在Hadoop异构环境中的设计与实现[D]. 南京:南京理工大学,2012. (Li Yuanhong. Design and implementation of dynamic load balancing scheduling algorithm based on feedback in Hadoop heterogeneous environment[D]. Nanjing: Nanjing University of Science & Technology, 2012.)
- [12] Kushwaha M, Raina B L, Singh S N. Advanced weighted round robin procedure for load balancing in cloud computing environment[C]//Proc of the 11th International Conference on Cloud Computing, Data Science & Engineering. 2021: 215-219. (下转第542页)

### 3 结束语

本文提出将高斯—马尔可夫移动模型用于双圆柱 UAV-MIMO 系统,来表示非平稳的空对地信道,并扩展该模型,让其具有广泛性。扩展后的模型不仅能调整模型的随机性,还能描述随机运动、线性运动等不同情况下的移动轨迹。基于双圆柱模型,本文推导了时变的空时相关函数和多普勒功率谱密度,研究了 UAV 水平运动和垂直运动对统计特性的影响。仿真结果表明,动态运动仅会导致时域的非平稳,对空域没有影响,并可能导致对时间相关函数的影响。通过与测量值的对比发现,引入移动模型更符合无人机实际的通信场景。这些研究可以作为具有逼真的移动轨迹的 UAV-MIMO 空对地非平稳信道建模的参考。在未来空地一体化通信网络中,无人机可以作为卫星对地通信的中继站,在未来的研究中,还需要将信道模型推广到中继站对地的通信场景中。

#### 参考文献:

- [1] Shakhathreh H, Sawalmeh A H, Al-Fuqaha A, *et al.* Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges[J]. *IEEE Access*, 2019, 7: 48572-48634.
- [2] Wang Chengxiang, Huang Jie, Wang Haiming, *et al.* 6G wireless channel measurements and models: trends and challenges[J]. *IEEE Vehicular Technology Magazine*, 2020, 15(4): 22-32.
- [3] 贾汝冰. 无人机场景下的宽带信道建模与仿真[D]. 北京: 北京交通大学, 2019. (Jia Rubing. Modeling and simulation of broadband channel in UAV scenario[D]. Beijing: Beijing Jiaotong University, 2019.)
- [4] Jiang Hao, Zhang Zaichen, Wu Liang, *et al.* A non-stationary geometry-based scattering vehicle-to-vehicle MIMO channel model[J]. *IEEE Communications Letters*, 2018, 22(7): 1510-1513.
- [5] Greenberg E, Levp P. Channel characteristics of UAV to ground links over multipath urban environments[C]//Proc of IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems. Piscataway, NJ: IEEE Press, 2017.
- [6] Matolak D W, Sun Ruoyu. Air-ground channel characterization for unmanned aircraft systems, part I: methods, measurements, and models for over-water settings[J]. *IEEE Trans on Vehicular Technology*, 2017, 66(1): 26-44.
- [7] Matolak D W, Sun Ruoyu. Unmanned aircraft systems: air-ground channel characterization for future applications[J]. *IEEE Vehicular Technology Magazine*, 2015, 10(2): 79-85.
- [8] Jin Kun, Cheng Xiang, Xiao Huge, *et al.* Three dimensional modeling and space-time correlation for UAV channels[C]//Proc of the 85th IEEE Vehicular Technology Conference. Piscataway, NJ: IEEE Press, 2017.
- [9] Zeng Linzhou, Cheng Xiang, Wang Chengxiang, *et al.* A 3D geometry-based stochastic channel model for UAV-MIMO channels[C]//Proc of IEEE Wireless Communications and Networking Conference. Piscataway, NJ: IEEE Press, 2017.
- [10] Jiang Hao, Zhang Zaichen, Wu Jiang, *et al.* Three-dimensional geometry-based UAV-MIMO channel modeling for A2G communication environments[J]. *IEEE Communications Letters*, 2018, 22(7): 1438-1441.
- [11] Cai Xuesong, Gonzalez-Plaza A, Alonso D, *et al.* Low altitude UAV propagation channel modelling[C]//Proc of the 11th European Conference on Antennas and Propagation. Piscataway, NJ: IEEE Press, 2017: 1443-1447.
- [12] Biomo J D M M, Kunz T, St-Hilaire M. An enhanced Gauss-Markov mobility model for simulations of unmanned aerial Ad hoc networks[C]//Proc of the 7th IFIP Wireless and Mobile Networking Conference. Piscataway, NJ: IEEE Press, 2014.
- [13] He Ruisi, Ai Bo, Stüber G L, *et al.* Mobility model-based non-stationary mobile-to-mobile channel modeling[J]. *IEEE Trans on Wireless Communications*, 2018, 17(7): 4388-4400.
- [14] Batabyal S, Bhaumik P. Mobility models, traces and impact of mobility on opportunistic routing algorithms: a survey[J]. *IEEE Communications Surveys & Tutorials*, 2015, 17(3): 1679-1707.
- [15] 王亚文, 朱秋明, 陈小敏, 等. 基于几何随机的无人机空地信道模型[J]. 数据采集与处理, 2020, 35(1): 118-127. (Wang Yawen, Zhu Qiuming, Chen Xiaomin, *et al.* Novel geometry-based stochastic model for UAV-based air-to-ground channels[J]. *Journal of Data Acquisition & Processing*, 2020, 35(1): 118-127.)
- [16] Ma Zhangfeng, Ai Bo, He Ruisi, *et al.* Impact of UAV rotation on MIMO channel characterization for air-to-ground communication systems[J]. *IEEE Trans on Vehicular Technology*, 2020, 69(11): 12418-12431.
- [17] Chang Hengtai, Bian Ji, Wang Chengxiang, *et al.* A 3D wideband geometry-based stochastic model for UAV air-to-ground channels[C]//Proc of IEEE Global Communications Conference. Piscataway, NJ: IEEE Press, 2018: 206-212.
- [18] Zhu Qiuming, Jiang Kaili, Chen Xiaomin, *et al.* A novel 3D non-stationary UAV-MIMO channel model and its statistical properties[J]. *China Communications*, 2018, 15(12): 147-158.
- [19] Cheng Xiang, Li Yiran, Wang Chengxiang, *et al.* A 3D geometry-based stochastic model for unmanned aerial vehicle MIMO Ricean fading channels[J]. *IEEE Internet of Things Journal*, 2020, 7(9): 8674-8687.
- [20] Mathews J, Fink D K. Numerical methods using MATLAB[M]. 4th ed. New Jersey: Prentice-Hall Inc., 2005.
- [21] Simunek M, Fontán F P, Pechac P. The UAV low elevation propagation channel in urban areas: statistical analysis and time-series generator[J]. *IEEE Trans on Antennas and Propagation*, 2013, 61(7): 3850-3858.
- [13] 邵滢滢, 鹿影, 段苛苛, 等. 基于改进权重的 D-S 证据理论的动态负载均衡算法[J]. 计算机应用, 2018, 38(10): 2976-2981, 2989. (Tai Yingying, Pang Ying, Duan Keke, *et al.* Dynamic algorithm of load balancing based on D-S evidence theory with improved weight[J]. *Journal of Computer Applications*, 2018, 38(10): 2976-2981, 2989.)
- [14] 王宇耕, 肖鹏, 张力, 等. 基于负载预测的自适应权重负载均衡算法[J]. 计算机工程与设计, 2019, 40(4): 1033-1037. (Wang Yugeng, Xiao Peng, Zhang Li, *et al.* Adaptive weight load balancing algorithm based on load prediction[J]. *Computer Engineering and Design*, 2019, 40(4): 1033-1037.)
- [15] 吴宝花. 基于 Nginx 的服务器集群负载均衡策略研究与优化[D]. 南昌: 南昌大学, 2020. (Wu Baohua. Research and optimization of server cluster load balancing strategy based on Nginx[D]. Nanchang: Nanchang University, 2020.)
- [16] Xu Zongyu, Wang Xingxuan. A predictive modified round robin scheduling algorithm for Web server clusters[C]//Proc of the 34th Chinese Control Conference. Piscataway, NJ: IEEE Press, 2015: 5804-5808.
- [17] Sun Yu, Mao Shaojie, Huang Songhua, *et al.* Load balancing method for service scheduling of command information system[C]//Proc of the 2nd Information Communication Technologies Conference. Piscataway, NJ: IEEE Press, 2021: 297-301.
- [18] Murti K W, Riza T A, Mulyana A. Comparative analysis of load balancing dynamic ratio and server ratio algorithms[C]//Proc of FORTEL-International Conference on Electrical Engineering. 2020: 162-167.
- [19] 孟晓景, 张春勇. 一种改进加权最小连接算法及在 CDN 的负载均衡技术中的应用分析[J]. 山东科技大学学报: 自然科学版, 2020, 39(1): 85-90. (Meng Xiaojing, Zhang Chunyong. An improved weighted least connection algorithm and its application analysis in CDN load balancing technology[J]. *Journal of Shandong University of Science & Technology: Natural Science*, 2020, 39(1): 85-90.)
- [20] Manirabona A, Boudjit S, Fourati L C. A priority-weighted round robin scheduling strategy for a WBAN based healthcare monitoring system[C]//Proc of the 13th IEEE Annual Consumer Communications & Networking Conference. Piscataway, NJ: IEEE Press, 2016: 224-229.

(上接第 530 页)