

## 基于动态阈值分配的流媒体边缘云会话迁移策略

姜同全, 王子磊, 奚宏生

(中国科学技术大学 自动化系, 合肥 230027)

**摘 要:** 云模式下的流媒体服务系统需要有效与平滑的资源调度。传统的 last-minute 迁移大多只将负载信息用于迁移策略计算, 但当流行度动态波动时, 间接且单一的方法无法做出准确的策略调整。为此, 将流行度分布应用于迁移模型中, 提出一种基于动态阈值分配的会话迁移策略, 依据流行度分布, 确定每台服务器上各类视频的会话分配阈值, 通过分配阈值的指导性避免请求接入及会话迁移的盲目性。实验结果表明, 该策略能有效改善请求接受率, 并保持较低的迁移代价。

**关键词:** 流媒体边缘云; 资源调度; 会话迁移; 流行度分布; 动态阈值分配; 迁移代价

**中文引用格式:** 姜同全, 王子磊, 奚宏生. 基于动态阈值分配的流媒体边缘云会话迁移策略[J]. 计算机工程, 2017, 43(1): 55-60.

**英文引用格式:** Jiang Tongquan, Wang Zilei, Xi Hongsheng. Session Migration Strategy for Streaming Media Edge Cloud Based on Dynamic Threshold Allocation[J]. Computer Engineering, 2017, 43(1): 55-60.

## Session Migration Strategy for Streaming Media Edge Cloud Based on Dynamic Threshold Allocation

JIANG Tongquan, WANG Zilei, XI Hongsheng

(Department of Automation, University of Science and Technology of China, Hefei 230027, China)

**【Abstract】** For the streaming media system based on the cloud model, more efficient and smooth resource scheduling is strongly required. In the traditional “last-minute migration”, only the load information is used for the migration scheduling. Due to the dynamic fluctuation of the video population, such indirect and single way cannot guarantee effective scheduling. Besides the load information, the video popularity is also used for the migration model, this paper propose a dynamic threshold allocated migration strategy. The allocation thresholds of different videos on each server are obtained on the basis of the new video popularity, and due to the guidance of the allocation threshold distribution, the blindness of request access and session migration can be avoided effectively. Experimental results show that the strategy cannot only effectively improve the request acceptance rate, but also keep the migration cost low enough.

**【Key words】** Streaming Media Edge Cloud (MEC); resource scheduling; session migration; distribution of popularity; dynamic threshold allocation; migration cost

**DOI:** 10.3969/j.issn.1000-3428.2017.01.010

### 0 概述

传统的基于内容分发网络(CDN)、对等网络(P2P)技术的系统在扩展性、可靠性、服务封装能力等方面存在不同程度的局限性。近年来, 随着云计算的成熟, 流媒体服务正向云模式演变。通过在不同区域分别放置流媒体边缘云(Streaming Media Edge Cloud, MEC), 将用户所请求的视频内容推到

网络边缘, 以减小用户响应延迟和降低骨干网络流量负载。目前, 已经有多种成熟的流媒体云平台, 例如一种基于 P2P 流媒体分发技术的“云模式流媒体服务平台”<sup>[1]</sup>, 已成功应用于华数互联网电视中。

在流媒体边缘云内, 系统资源通常被虚拟化为资源池, 以保证服务透明性; 而且, 云资源的分配是根据实际需求的规模而由云平台自动调整, 即云资源的弹性分配。因此, 如何实时地分配资源以满足

**基金项目:** 国家“863”计划项目(2014AA06A503); 国家自然科学基金(61233003); 中央高校基本科研业务费专项资金项目(WK3500000002)。

**作者简介:** 姜同全(1990—), 男, 硕士研究生, 主研方向为云计算、网络多媒体; 王子磊, 副教授; 奚宏生, 教授。

**收稿日期:** 2016-03-04      **修回日期:** 2016-04-11      **E-mail:** zlwang@ustc.edu.cn

用户需求,是流媒体边缘云资源调度的难点<sup>[2]</sup>。特别地,在分配资源有限的条件下,用户请求模式的倾斜性及波动性,往往导致“会话分布不合理”问题:由于用户请求到来的随机性等原因,系统负载均衡受到破坏,进而影响请求接入效果。

为了解决上述问题,会话迁移<sup>[3]</sup>被提出并得到广泛的应用。但是,传统会话迁移直到新请求无法接入时才被触发,即所谓的“last-minute 迁移”<sup>[4]</sup>。显然,这种机制容易产生“迁移滞后”问题。为了进一步提高迁移效果,研究者对会话迁移算法进行了大量研究。文献[5]提出一种 DASD-dance 算法,通过负载相差较大的节点之间进行会话迁移,以保证系统负载均衡。文献[6]提出了一种随机早期迁移机制(Random Early Migration, REM),依据服务器负载状态而确定迁移触发的概率,较好地解决了“迁移滞后”问题。

实际上,已有迁移方法具有相同的处理方式:依据负载状态确定触发时刻,并通过维持负载分布均衡,而间接地优化请求接入。显然,已有迁移方法对新请求的处理,只能完全地依靠节点负载状态(例如最小负载优先),无法保证资源分配与请求模式相匹配。其实,除负载信息之外,各类视频的流行度分布也会影响会话迁移的效果。特别地,当流行度发生波动时,借助于流行度分布信息,既可保证迁移触发及时性,还能提高迁移效果。

通过综合利用负载状态和流行度分布信息,本文提出一种基于动态阈值分配的会话迁移策略。考虑到实际情形下的诸多局限,在迁移策略执行方面,给出一种“Lazy 迁移机制”以及一种基于分支限界法的多步迁移算法,以提高迁移性能。

## 1 流媒体边缘云架构

流媒体边缘云位于网络边缘,用于负责本地区的视频服务。如图 1 所示,结合当前流行的 OpenFlow 技术<sup>[7]</sup>,本文设计了一种新颖的 MEC 架构。整个 MEC 由流媒体服务器、业务管理服务器、OpenFlow 控制机及交换机组成,其中,流媒体服务器负责向用户提供媒体流;业务管理服务器主要负责用户请求的接入调度、生成迁移策略并下发给 OpenFlow 控制器;OpenFlow 控制器及交换机,一方面构成了媒体流分发网络,另一方面负责会话迁移的实际执行<sup>[8]</sup>;OpenFlow 控制器负责依据迁移策略生成流表,并下发给交换机;OpenFlow 交换机则根据流表完成数据包的修改和转发。

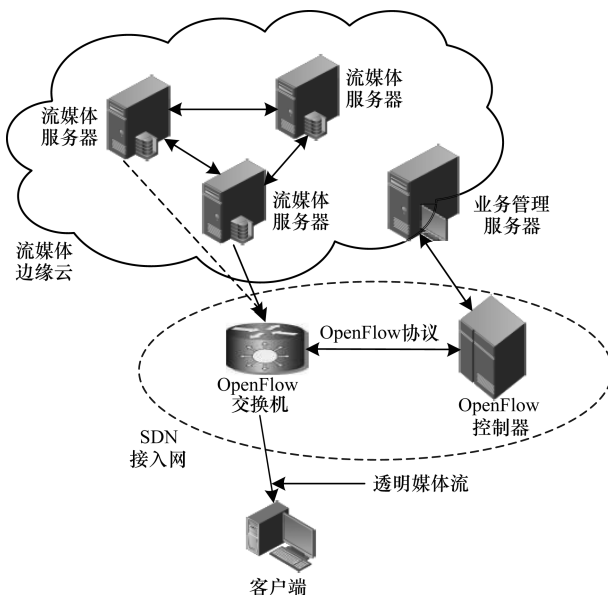


图 1 基于 OpenFlow 的流媒体边缘云架构

通过引入 MEC 架构,应用层的流媒体业务处理和网络层的转发路径优化得以分离,实现了视频服务的透明性。

## 2 问题描述

本文借助会话迁移来提高 MEC 的资源利用。由于实时性的资源调度将带来过高的迁移代价,如何处理迁移效果和代价的权衡是资源优化的关键。

假设  $\{v_i\}_{i=1}^I$  是 MEC 所提供的  $I$  种不同视频内容,每种视频均为恒定码率编码,以相同码率提供服务;  $p_i, i=1, 2, \dots, I$  表示不同视频的流行度。假设 MEC 流媒体服务器总数为  $J$ , 第  $j$  台服务器用  $M_j$  表示。假设会话分配矩阵表示为  $A$ , 大小为  $I \times J$ , 元素  $a_{ij}$  表示视频  $v_i$  在服务器  $M_j$  上所分配的会话数阈值与系统总服务能力的比值。定义  $D$  作为视频部署矩阵, 大小为  $I \times J$ , 元素  $d_{ij} \in \{0, 1\}$  表示  $M_j$  上是否部署了  $v_i$  的副本。为了简单起见,假设所有服务器都同构,并且单个服务器最多可同时提供  $C$  个流媒体会话,以及最多可存储  $S$  份视频。

假定流行度  $p_i, i=1, 2, \dots, I$  是已知的;依据 MEC 所记录的服务信息,利用一些预测算法<sup>[9-10]</sup>可准确获得各类视频的流行度。这样,本文研究问题可描述为:在流行度动态变化下,实时地调整会话分配矩阵  $A$ , 以较小的迁移代价接纳更多的请求。

## 3 基于动态阈值分配的会话迁移策略

### 3.1 会话迁移策略

为了充分利用流行度及负载分布信息,本文提

出一种基于动态阈值分配的会话迁移策略,以通过预算的会话分配阈值来指导新请求的接入。

首先,根据流行度分布,确定每类视频在各服务器上的会话分配  $\mathbf{A}$ 。然后,对于新视频请求,若同类视频的会话总数未超过分配阈值,则在会话数低于分配阈值的服务器中,按照最小负载优先接入;否则,说明流行度已产生波动,需重新确定  $\mathbf{A}$ 。

另外,考虑到实际情形下的许多局限,本文采用了一种“Lazy 迁移机制”,将迁移实际执行延迟至请求视频的部署节点都已满载时,而非在迁移计算之后立即执行。为了增强迁移执行效果,本文基于分支限界法还提出一种多步迁移执行算法。

### 3.2 迁移计算模型

会话迁移模型作用是:根据给定的流行度分布,重新计算会话分配矩阵  $\mathbf{A}$ ,以获得最优接入效果,并且保证较低迁移代价。由于元素  $a_{ij}$  反映了  $M_j$  对  $v_i$  的请求接入数,因此优化目标可表示为:

$$\max \left\{ \sum_{i=1}^I \sum_{j=1}^J a_{ij} \right\} \quad (1)$$

即最大化请求接受率。在基于 SDN 的 MEC 架构中,迁移代价可用被迁移会话数来表示。假设  $K$  表示各服务器上不同视频的实际会话数,大小为  $I \times J$ ,元素  $k_{ij} \in [0,1]$  表示视频  $v_i$  在服务器  $M_j$  上的会话数占总服务能力的比例,则迁移代价可表示为  $\sum_{i=1}^I \sum_{j=1}^J \max(k_{ij} - a_{ij}, 0)$ 。另外,本文规定了单次迁移代价最大阈值  $M$ ,即  $\sum_{i=1}^I \sum_{j=1}^J \max(k_{ij} - a_{ij}, 0) \leq M$ ;其中,  $M$  值通常由 OpenFlow 的流表处理能力决定。

在负载不均衡分布下,对于已满载的服务器,只有将部分会话迁出,才能继续接入新请求。因此,负载是否均衡将间接地影响迁移代价。在实际情形下,由于请求分布的波动性,各类视频请求并非严格地按照流行度而到达,上述优化接受率的方案,很容易造成负载不均衡,从而导致代价增加。因此,引入 2 个关于负载均衡维持的目标。

1) 对于即时到达的新请求  $r$ ,由于分配阈值的指导性,  $r$  只能接至“会话数小于分配阈值”的节点上,这样只有使最小负载节点的分配阈值大于会话数(差值至少为 1),才能最大程度地降低负载不均衡。因此,增加了如下新优化目标:

$$\min \left\{ \sum_{i=1}^I \sum_{j=1}^J \max(a_{ij}^r - a_{ij}, 0) \right\} \quad (2)$$

其中,  $\mathbf{A}^r$  是一个常量矩阵,计算方法是:对于  $v_r$  (即  $r$  的请求视频),假设  $M_j$  是部署  $v_r$  且负载最小的节点,则对应元素  $a_{rj}^r = k_{rj} + 1$ ;当然,其余节点上的对应元素  $a_{ij}^r = k_{ij} + 1, i \neq r$ 。另外,对于其余视频  $v_i, i \neq r$ ,

其对应元素分别为  $a_{ij}^r = k_{ij}, \forall j \leq J$ 。

2) 对于后续到达的所有请求而言,为使其都接至最小负载节点,需保证各分配阈值都大于会话数;另外,考虑到请求到达的连续性及随机性,分配阈值与会话数的差值,还应与请求到达情况等因素相关。因此,又增加了如下的新优化目标:

$$\min \left\{ \sum_{i=1}^I \sum_{j=1}^J \theta_i \times \max(a_{ij}^* - a_{ij}, 0) \times \gamma_j \right\} \quad (3)$$

其中,  $\mathbf{A}^*$  是一个常量矩阵;  $\theta_{1 \times I}$  和  $\gamma_{J \times 1}$  为权值向量,计算方法是: (1)  $\mathbf{A}^*$ , 假设  $R_i$  表示  $v_i$  仍未到达的请求数,其可近似表示为  $R_i = \max(JCp_i - \sum_{j=1}^J k_{ij}, 0)$ 。对于部署  $v_i$  的节点集合  $N_i = \{j | d_{ij} = 1, j \leq J\}$ , 对应元素  $a_{ij}^* = k_{ij} + R_i / \sum_{j=1}^J d_{ij}$ , 其中,  $\sum_{j=1}^J d_{ij}$  是  $v_i$  总副本数。对于其余节点,对应元素  $a_{ij}^* = 0$ 。 (2)  $\theta_{1 \times I}$  和  $\gamma_{J \times 1}$ , 考虑热门视频更容易影响负载分布,则权值  $\theta_i$  可取  $p_i$ ; 考虑到轻载节点更应分配较大阈值,权值  $\gamma_j$  可取  $(1 - l_j/C)$ , 其中,  $l_j$  是  $M_j$  的实际会话总数。

从效果上看,节点负载越小,其分配阈值可大一点,以承担更多的请求接入;但由于 MEC 启动后便采用此优化策略,各节点负载基本均衡,因此采用上述平均分配处理仍可到达理想效果。

3) 除了对单次迁移代价的限制外,还应考虑如下约束:

(1) 每台服务器的服务能力限制:在服务器同构的假设条件下,最大服务能力可表示为  $1/J$ 。

(2)  $a_{ij}$  取值范围限制:只有  $M_j$  部署  $v_i$  时,才能够提供该视频服务,即当  $d_{ij} = 0$  时,  $a_{ij}$  只能取零;这点可巧妙地使用  $a_{ij} < d_{ij}$  进行限制。

(3) “实际会话数不减少”原则:在迁移前后,实际会话数保持不变,因此每类视频在系统中所分配的总阈值必须足够容纳会话数,即满足  $\sum_{j=1}^J a_{ij} \geq \sum_{j=1}^J k_{ij}$ ;特别地,  $\sum_{j=1}^J a_{ij}$  还应包括即时到来的新请求  $r$ 。

(4) 流行度分布约束:引入  $\sum_{j=1}^J a_{ij} \leq \alpha p_i$  来使各类视频的分配阈值与其流行度相匹配;其中,负载因子  $\alpha$  反映总体负载情况,理论上该值应在 1 附近上下浮动。

综上所述,以会话分配矩阵  $\mathbf{A}$  作为决策变量,迁移计算模型可表示如下:

$$Obj 1: \max \left\{ \sum_{i=1}^I \sum_{j=1}^J a_{ij} \right\}$$

$$Obj 2: \max \left\{ \sum_{i=1}^I \sum_{j=1}^J \max(a_{ij}^r - a_{ij}, 0) \right\}$$

$$Obj 3: \max \left\{ \sum_{i=1}^I \sum_{j=1}^J \theta_i \times \max(a_{ij}^* - a_{ij}, 0) \times \gamma_j \right\}$$

where,  $\mathbf{A}^r$  and  $\mathbf{A}^*$  are const matrix.

Subject to:

$$\sum_{i=1}^I \sum_{j=1}^J \max(0, k_{ij} - a_{ij}) \leq M$$

$$\sum_{j=1}^J a_{ij} \geq \sum_{j=1}^J k_{ij}, \forall i \leq I$$

$$\sum_{j=1}^J a_{ij} \leq \alpha p_i, \forall i \leq I$$

$$\sum_{i=1}^I a_{ij} \leq 1/J, \forall j \leq J$$

$$0 \leq a_{ij} \leq d_{ij}, \forall i \leq I, \forall j \leq J$$

### 3.3 迁移执行策略

考虑到实际情形下的许多局限,上述迁移模型仅用来计算会话分配阈值。本文提出了 Lazy 迁移机制和多步迁移算法,以负责会话迁移的实际执行。

#### 3.3.1 Lazy 迁移机制

新会话分配确定后,在会话数超过分配阈值的服务器上,若干会话应立即被迁出,以使分配阈值严格地大于会话数。为了提高迁移性能,迁移实际执行被延迟至部署请求视频的服务器都已满载时,这种处理称为“Lazy 迁移机制”。

显然,采用此机制具有以下优点:1)所有待迁移会话的实际迁移被分散到不同的时间点上,从而避免了大量并发迁移对系统性能的影响;2)由于用户请求到达以及会话退出的随机性,实际的请求分布与预测流行度并不一致,延迟迁移执行可避免因会话分布过早调整而导致的迁移代价。

#### 3.3.2 多步迁移算法

在传统的单步迁移中,待迁移会话由源服务器直接迁至目标服务器。虽然算法简单,其却无法保证最优的迁移效果。本文以分支限界法<sup>[11]</sup>为基础,采用多步迁移作为执行策略。

多步迁移算法分为图构建和迁移路径搜索两步。

1)构建连通图。如图2所示,假设图模型为  $G = (V, E)$ , 顶点表示服务器,边集合  $e_{ij}$  反映了两服务器间的共有视频。此外,每个图节点  $M_j$  还记录了各类视频的分配阈值和服务会话数等信息;依据这些信息,各节点上的视频分为超载  $V_i^+ = \{i | k_{ij} > a_{ij}\}$ 、满载  $V_i^0 = \{i | k_{ij} = a_{ij}\}$  和欠载  $V_i^- = \{i | k_{ij} < a_{ij}\}$  3类。

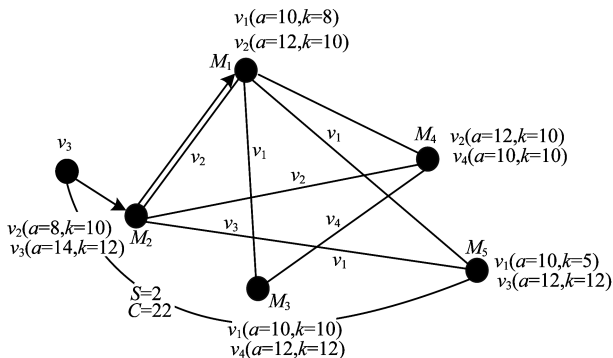


图2 多步迁移图构建示意图

2)利用上述得到的图模型,针对给定的视频请求,基于最短路径优先和最小负载次优先原则,搜索可行的迁移路径。特别地,采用分支限界法来搜索最短迁移路径,搜索树的产生过程如图3所示。首先将请求视频及其未满载的服务器作为所有可能分支;当搜索至下层节点时,按照如下方法继续产生分支:对于扩展节点上的超载视频,若其在某个相邻节点属于欠载类型,则该视频及其相邻节点就作为新分支;最后依次迭代更新,直到获取目标节点及迁移路径。显然,对于某个分支而言,前向节点就是迁出服务器,后向节点就是迁入服务器,而共有视频则对应被迁移会话。

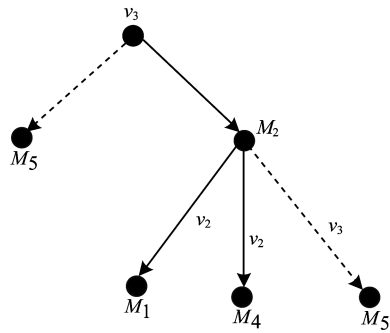


图3 单步路径生成示意图

值得注意的是,在绝大部分情况下,搜索树的深度维持在1或2,因而本文算法的复杂度并不高。

### 3.4 迁移策略工作流程

为了说明本文迁移策略原理,下面介绍新请求的处理流程。

**步骤1** 新请求  $r$  到来,首先判断系统是否还具有空闲的服务能力。若已满载,则直接拒绝。

**步骤2** 判断流行度分布是否发生变化,即视频  $v_r$  的当前会话数是否已超过总分配阈值;如果已超过,说明流行度分布发生变化,则进入步骤3;否则,进入步骤4。

**步骤3** 触发迁移计算;通过求解迁移计算模型,获取新会话分配矩阵  $A$ ;特别地,若模型无解,则直接拒绝;否则,便进入步骤4。

**步骤4** 按照分配阈值继续处理新请求。

**步骤4.1** 判断  $r$  是否可直接接入,首先寻找部署视频  $v_r$  的欠载服务器集合  $N_r$ , 即  $\{j | d_{rj} = 1, a_{rj} < k_{rj}, \forall j \leq J\}$ ; 然后判断  $N_r$  中最小负载节点是否满载;若已满载,则  $r$  无法直接接入,进入步骤4.2;否则,进入步骤4.3。

**步骤4.2** 执行多步迁移算法,创建有向图,并搜索可行的迁移路径;依据路径完成会话迁移,以及接入新请求。

**步骤4.3** 将  $r$  接至集合  $N_r$  中最小负载节点。

## 4 性能仿真与分析

### 4.1 系统参数设定

对于面向局部区域的流媒体边缘云,视频内容都是本地用户常点播的,只需选取固定数目的相对热门内容即可。在仿真测试中,设定 MEC 的视频种类数  $I = 300$ , 视频  $v_i$  的流行度服从 Zipf-like 分布<sup>[12]</sup>:

$$p_i = i^{-\psi} / \sum_{n=1}^I n^{-\psi}, \forall i \leq I \quad (4)$$

其中,  $\psi$  是倾斜系数,根据实际的视频点播规律,选取  $\psi = 0.7$ 。服务器总数  $J = 20$ ,其最大服务会话数  $C = 90$ ,最大存储副本数  $S = 30$ 。另外,选取单位时间为 1 min,请求到达是参数  $\lambda$  泊松过程<sup>[13]</sup>,并且会话时长服从文献[14]的分布,其中平均时长设定为 30 min;因此,当  $\lambda = 60(JC/30)$  个/min 时,系统达到满载。本文分别将请求接受率和总迁移会话数作为性能评价指标;仿真时长设定为 200 min。

### 4.2 仿真分析结果

首先将请求接受率作为性能指标,分别测取了在不同请求到达速率下的统计接受率。另外,为了更好地反映优化效果,采用了 2 组对比实验:无迁移策略和 DASD-dance 迁移策略<sup>[5]</sup>。实验结果如表 1 所示,其中,第 1 行表示请求到达速率,其余 3 行表示对应策略下的请求接受率。

表 1 不同策略对请求接受率的影响

请求到达速率 /(个·min <sup>-1</sup> )	本文 策略	无迁移 策略	DASD-dance 策略
55	1.000 00	1.000 00	1.000 00
56	1.000 00	1.000 00	1.000 00
57	1.000 00	0.999 73	0.999 91
58	1.000 00	0.998 09	0.999 22
59	1.000 00	0.994 46	0.995 08
60	1.000 00	0.986 01	0.986 67
61	0.985 17	0.974 67	0.974 96
62	0.970 82	0.961 61	0.961 50
63	0.956 93	0.948 41	0.949 03
64	0.943 47	0.935 16	0.935 46

可以看出,无论是在低负载( $\lambda \leq 60$ )还是在高负载( $\lambda > 60$ )情形下,对于请求接受率,本文策略及 DASD-dance 策略都优于无迁移策略,这说明会话迁移确实可以通过优化资源分配而提高 MEC 服务性能。另一方面,在高负载情形下,本文策略效果要明显优于 DASD-dance 策略;这是因为:对于 DASD-

dance 策略,只有负载不均衡度超过预设阈值时,才触发迁移;而本文策略直接面向新请求,即一旦请求无法被接入,则立即触发多步迁移执行。因此,本文策略能保证更多的新请求接入。

为了更好地反映控制效果,还采用了两组对比实验:“最小负载优先接入(Smallest Load First, SLF) + 最短路径迁移(Shortest Path Migration, SPM)”策略<sup>[15]</sup>,以及“随机请求接入(Random Request Access, RRA) + 最短路径迁移(SPM)”策略<sup>[16]</sup>。首先,分别测取了不同请求到达速率下的请求接受率;实验结果显示,3 种迁移策略的统计接受率一致。这样,在接入效果相同的条件下,详细统计了 3 种策略的总迁移会话数,具体结果如表 2 所示。其中,第 1 行表示请求到达速率,其余 3 行表示对应策略下的总迁移会话数。

表 2 不同策略对迁移代价的影响

请求到达速率 /(个·min <sup>-1</sup> )	本文策略	SLF + SPM 策略	RRA + SPM 策略
55	0	0	194
56	1	3	283
57	3	9	400
58	14	24	627
59	114	134	974
60	1 143	1 176	2 097
61	1 191	1 195	2 045
62	1 173	1 203	2 084
63	1 062	1 142	2 044
64	1 049	1 169	1 954

可以看出,无论在何种负载情形,对于迁移会话数,本文策略及“SLF + SPM”策略都低于“RRA + SPM”策略。这是因为:未指导性地将新请求接至合适的节点,“RRA + SPM”策略很容易造成负载不均衡。另外,本文策略的迁移代价还要稍低于“SLF + SPM”策略;这是因为:除负载分布外,本文策略还将流行度分布用于会话迁移,即通过会话分配阈值的接入指导性,避免随机接入的盲目性。

为了进一步验证利用分配阈值确实可降低迁移代价,在请求到达严格地遵循流行度分布下,分别测取了本文策略及“SLF + SPM”策略在低负载情形下的总迁移次数。实验结果如表 3 所示。其中,第 1 行表示请求到达速率,其余 2 行表示对应策略下的总迁移会话数。

表 3 分配阈值对迁移代价的影响

请求到达速率/(个·min <sup>-1</sup> )	本文策略	SLF + SPM 策略
57	0	0
58	0	3
59	0	4
60	0	190

综上所述,本文策略不仅考虑了负载情况,而且将流行度分布用于迁移计算和执行中;通过会话分配阈值来指导新请求的接入,既保证了最优的接入效果,又有效地控制了迁移代价。

## 5 结束语

本文提出一种基于动态阈值分配的会话迁移策略,并将流行度分布用于该策略。考虑到请求分布的动态波动,本文还将负载分布用于分配阈值的重新计算。通过综合利用流行度分布和负载信息,该策略可实现优化效果和迁移代价的合理权衡。另外,考虑到实际情形下的局限,给出了“Lazy 迁移机制”和多步迁移算法。但是会话迁移的调度效果是有限的,特别是在流行度发生剧烈变化的情况下,而动态视频部署则更适合于剧烈变化的情形。因此,将会话迁移和动态视频部署相结合将是下一步的工作重点。

### 参考文献

- [1] 北京原力创新科技有限公司. 云模式流媒体服务平台:200910091537.2[P]. 2010-01-27.
- [2] Wang Feng, Liu Jiangchuan, Chen Minghua. CALMS: Cloud-assisted Live Media Streaming for Globalized Demands with Time/Region Diversities[C]//Proceedings of IEEE INFOCOM'12. Orlando, USA; IEEE Press, 2012: 199-207.
- [3] 周俊,李文中,陆桑璐,等. 利用网格技术实现流媒体服务迁移[J]. 计算机科学, 2005, 32(8): 109-113.
- [4] Dhage S N, Meshram B B. Design and Implementation of Video Servers for VoD System[J]. International Journal of Cloud Computing, 2013, 2(1): 61-88.
- [5] Wolf J L, Philip S Y, Shachnai H. Disk Load Balancing for Video-on-demand Systems[J]. Multimedia Systems, 1997, 5(6): 358-370.
- [6] Zhao Yingqing, Kuo C J. Video-on-demand Server System Design with Random Early Migration[C]//Proceedings of 2003 International Symposium on Circuits and Systems. Bangkok, Thailand: [s. n.], 2003: 640-643.
- [7] 左青云,陈鸣,赵广松,等. 基于 OpenFlow 的 SDN 技术研究[J]. 软件学报, 2013, 24(5): 1078-1097.
- [8] 战立松,奚宏生,王子磊. 基于 OpenFlow 的流媒体云服务迁移方法[J]. 计算机工程, 2014, 40(12): 1-5.
- [9] Niu Di, Xu Hong, Li Baochun, et al. Quality-assured Cloud Bandwidth Auto-scaling for Video-on-demand Applications[C]//Proceedings IEEE INFOCOM'12. Orlando, USA; IEEE Press, 2012: 460-468.
- [10] Gürsun G, Crovella M, Matta I. Describing and Forecasting Video Access Patterns[C]//Proceedings IEEE INFOCOM'11. Shanghai, China: [s. n.], 2011: 16-20.
- [11] Lawler E L, Wood D E. Branch-and-bound Methods: A Survey[J]. Operations Research, 1966, 14(4): 699-719.
- [12] Qiu Tongqing, Ge Zihui, Lee Seung-Joon, et al. Modeling Channel Popularity Dynamics in a Large IPTV System[J]. ACM SIGMETRICS Performance Evaluation Review, 2009, 37(1): 275-286.
- [13] Yu Hongliang, Zheng Dongdong, Zhao B Y, et al. Understanding User Behavior in Large-scale Video-on-demand Systems[J]. ACM SIGOPS Operating System Review, 2006, 40(4): 333-344.
- [14] Chen Liang, Zhou Yipeng, Dah Ming Chiu. Smart Streaming for Online Video Services[J]. IEEE Transactions on Multimedia, 2015, 17(4): 485-497.
- [15] Huang Yinfu, Fang C C. Load Balancing for Clusters of VOD Servers[J]. Information Sciences, 2004, 164(1): 113-138.
- [16] Guo Jun, Wong Eric W M, Chan S, et al. Performance Analysis of Resource Selection Schemes for a Large Scale Video-on-demand System[J]. IEEE Transactions on Multimedia, 2008, 10(1): 153-159.

编辑 索书志