

## 多控制器集群流媒体缓存代理的研究

杨波 廖建新 朱晓民

(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

**摘要:** 为满足用户分散的大规模视频流媒体应用需求, 该文提出了一种基于多控制器集群缓存代理结构(Multi-controller based Cluster Streaming Cache Proxy, MCSCP), 该体系由一组控制器构成控制子系统, 由多个内容存储器构成分布式存储子系统。对控制子系统的的关键问题进行了深入研究: 采用图论思想给出控制器组的选取算法, 设计了一种基于优先级的主控制器选举和在线切换协议(Priority based Master-controller Election and Handover, PMEHL)。建立数学模型分析了多控制器体系对系统可靠性的改善程度, 讨论了控制器选取算法的性能, 研究了协议中报文发送间隔变量与系统开销的关系, 并使用仿真实验给出了该参数的最佳取值范围。

**关键词:** 集群缓存代理; 协同控制; 主控选举和切换; 性能优化

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2007)08-1979-05

## Research on Multi-controller Based Cluster Streaming Cache Proxy

Yang Bo Liao Jian-xin Zhu Xiao-min

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** To satisfy the requirement of large-scale streaming application with dispersive users, a novel system architecture of Multi-controller based Cluster Streaming Cache Proxy (MCSCP) is proposed. Two subsystems are introduced in this architecture: a control subsystem consists of multiple candidate cooperative controllers, and a storage subsystem is composed of content storage group. Key techniques in the control subsystem are discussed deeply: based on the graph theory, an algorithm for the selection of candidate controllers is presented; a protocol of Priority based Master controller Election and Handover (PMEHL) is designed. The mathematical models are achieved to analyze the improvement in system reliability by using cooperative controllers architecture, to discuss the performance of controller election algorithm and to evaluate the relationship between the system cost and the variable of message interval. Simulation experiment is also carried out to obtain the optimal range for the value of this variable.

**Key words:** Cluster caching proxy; Cooperative control; Master controller election and handover; Performance optimization

### 1 引言

视频流媒体技术是宽带IP网络和多媒体技术共同发展的产物<sup>[1]</sup>。近年来, 围绕媒体内容的生成、存储、传输、缓存和接收, 已经开展了大量研究。流媒体缓存的作用是根据用户对视频节目访问模式所具有的集中特性, 利用位于用户网络中的代理服务器缓存热门节目, 以节省网络带宽和视频服务器处理能力, 为用户提供有质量保证的媒体服务。

目前对缓存代理的研究大多沿袭WEB缓存思路, 集中在单缓存服务器环境下的缓存策略<sup>[2, 3]</sup>、替换算法<sup>[4]</sup>、存储方案<sup>[5, 6]</sup>等方面。流媒体业务具有大容量、长服务时间等特性,

单个代理服务器在网络I/O带宽、磁盘存储容量等方面存在限制, 使其不适用于大规模应用。Soam<sup>[7]</sup>提出了一种分布式代理体系MiddleMan, MiddleMan是一个分布存储、单点控制的多机集群系统。这种系统的特点是控制功能简单有效、容易实现, 但也存在明显的缺点: 协同器是系统中的关键点, 其功能庞杂、负荷繁重, 可能成为系统的瓶颈; 协同器(Coordinator)是系统的单控制点, 其功能故障会导致整个系统的失效, 限制了系统的可靠性和扩展性。文献[8]提出了一种适用于移动环境的分布式缓存结构MobileCache, 并对缓存策略、缓存增益等进行了深入研究, 但并未探讨缓存控制器选择、系统可靠性与性能等问题。

针对上述情况, 本文提出一种基于多控制器的集群流媒体代理结构MCSCP(Multi-controller based Cluster Streaming Cache Proxy), 一组以主备方式工作的控制器构成控制子系统, 多个内容缓存器构成存储子系统; 设计了一

2005-11-23 收到, 2006-05-23 改回

国家杰出青年科学基金(60525110), 新世纪优秀人才支持计划(NCET-04-0111), 高等学校博士学科点专项科研基金(20030013006) 电子信息产业发展基金项目(基于 3G 的移动业务应用系统) 电子信息产业发展基金重点项目(一代网络核心业务平台)资助课题

种基于优先级的主控制器和切换协议PMEH(Priority based Master-controller Election and Handover), 使用在线监控和故障切换机制提高了控制子系统的可靠性, 增强了流媒体系统的稳定性和扩展性。

本文的组织结构如下: 第2节提出了基于多控制器的集群流媒体代理结构MCSCP; 第3节使用图论思想给出了一种控制器组选取算法, 从集群系统中选择合适的代理服务构成控制子系统; 第4节设计了基于优先级的主控制器和切换协议PMEH, 以实现主控制器选举和在线故障切换功能; 第5节建立数学模型对该体系结构的性能进行分析, 包括: 评估多控制器集群对系统可靠性的改善, 讨论了控制器选取算法的性能, 研究协议中报文发送间隔参数与系统开销的关系, 并使用仿真实验估计了该参数的最优取值范围; 第6节总结了MCSCP系统的特点和本文的贡献。

## 2 基于多控制器的集群流媒体缓存代理结构

基于多控制器的集群流媒体缓存代理结构MCSCP是对MiddleMan的改进, 用一组控制器代替原有的单一控制器, 分担了控制器的负荷, 增强了系统的可靠性。在MCSCP中, 代理服务器同时具备控制功能和存储功能两种部件, 根据应用的需要激活其中一种或两种功能, 将激活了控制功能的代理服务器称为控制器, 将激活了存储功能的代理服务器称为内容存储器。控制器和内容存储器是逻辑概念, 根据应用环境和网络结构可动态地将一台代理服务器指派为控制器或内容存储器。据此, 将MCSCP系统分为两个逻辑的子系统: 所有控制器构成控制子系统; 所有内容缓存器构成存储子系统。若将系统中的代理服务器集合记作 $PROXY = \{Proxy_1, Proxy_2, \dots, Proxy_x\}$ , 存储子系统记做STR, 控制子系统记做CTRL, 则有 $CTRL \cup STR = PROXY$ 。在如图1所示系统中,  $PROXY = \{Proxy_1, Proxy_2, Proxy_3\}$ ,  $CTRL = \{Proxy_1, Proxy_2\}$ ,  $STR = \{Proxy_1, Proxy_3\}$ 。

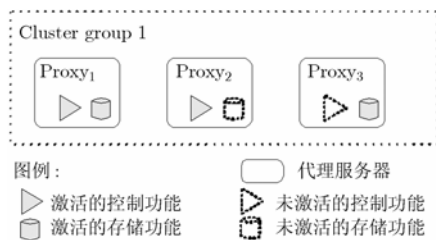


图1 代理服务器的功能划分

控制子系统的控制器分为主控制器和后备控制器两种角色, 其中仅有一个主控制器(master controller), 其余作为后备控制器(backup controller)。主控制器的功能包括: (1)对用户请求进行鉴权和接纳控制; (2)根据用户请求查找缓存映射表, 将请求重定位到相应内容存储器上; (3)实时监控各控制器和内容存储器的运行状态和负载情况, 以便对内容存储器实施负载均衡和故障隔离; (4)参与主控制器身份的竞选和维持; (5)与后备控制器定期同步缓存信息和控制信息

表。后备控制器的功能包括: (1)在主控制器的控制下, 执行缓存替换、复制、迁移策略等后台工作; (2)定期与主控制器同步缓存信息和控制信息表; (3)参与主控制器的竞争和在主服务器失效后迅速切换为主控制器; (4)定期向主控制器发送运行状态报告。主控制器和后备控制器之间通过PMEH协议交互, 实现基于优先级的主控制器选举和在线故障切换。

在MCSCP系统中, 使用PMEH协议从控制器组中选举产生主控制器, 并在其发生故障时在线切换。这种设计的优点体现在: (1)通过有效的故障检测和在线切换机制, 加强了控制子系统的稳定性, 避免系统出现单点故障; (2)将控制任务按优先级、实时性要求分类, 将非实时业务分配到后备控制器上执行, 降低了主控制器的负荷, 提高了主控制器的处理效率; (3)将缓存查找和缓存替换策略分离, 降低了系统的耦合度, 增强了系统的扩展性; (4)良好的自组织性, 网络失效导致原有的单控制区分裂为多个控制区域时, 每个区域能够重新选举主控制器构成新的集群。

存储子系统由内容存储器构成, 内容存储器在控制子系统的操纵下缓存视频内容, 向用户提供内容服务, 并定期向主控制器报告其运行参数(如: 用户请求强度、负载状况、可用资源数量等)。限于篇幅, 本文不将存储子系统作为研究重点。

在一个流媒体系统中, 各代理服务器具有不同的性能参数(如: 处理能力、I/O带宽、存储容量、服务强度), 各代理服务器之间的链路具有不同的性能参数(如: 网络带宽、传输时延、代价), 能否选择合适的代理服务器作为控制器, 将直接影响系统的处理效率和运行代价。如何从PROXY集合的 $x$ 个代理服务器中选取 $z$ 个控制器构成控制子系统CTRL是第3节讨论的内容; 如何从 $z$ 个控制器中选取主控制器, 并在主控制器失效后及时选举新的主控制器, 是第4节研究的问题。

## 3 控制器组的选取算法

控制器组的选取问题, 可以归结为图论中完全赋权图(Complete Weighted Graph, CWG)重心(Gravity Center, GC)的求解。

### 3.1 完全赋权图的重心

定义1 (完全赋权图CWG)

图 $G = (V, E, M, L) \in CWG$ , 其中: 顶点集 $V \triangleq V(G) = \{v_1, v_2, \dots, v_n\}$ ; 边集 $E \triangleq E(G) = \{e_1, e_2, \dots, e_p\}$ ; 顶点的权值集 $M \triangleq M(G) = \{m_1, m_2, \dots, m_n\}$ , 顶点 $v_i$ 的权值为 $m_i$ , 泛指质量; 边的权值集 $L \triangleq L(G) = \{l_1, l_2, \dots, l_p\}$ , 边 $e_j$ 的权值为 $l_j$ , 泛指长度; 则图 $G$ 称为完全赋权图。

定义2 (完全赋权图的重心GC)

设图 $G = (V, E, M, L) \in CWG$ , 设 $d_{i,j}$ 为 $v_i, v_j$ 两点间的距离(即两点间路径上边的权值和),  $S_j(G) \triangleq$

$\sum_{i=1}^n (m_i(G) \cdot d_{i,j})$  定义为点  $v_j$  对应的功,  $SM_j(G) \triangleq \min_{v_j \in V(G)} (S_j(G))$  定义为  $v_j$  功的最小值, 则  $MP(G) \triangleq \{v_k | v_k \in V(G), S_k(G) = SM_k(G)\}$  定义为图  $G$  的重心。

**定理 1** 完全赋权图的重心, 通常位于图  $G$  的某个顶点上<sup>[9]</sup>。

**算法 1** (完全赋权图的重心求法——矩阵验算法<sup>[9]</sup>)

令  $D(G) = (d_{i,j})_{n \times n}$ ,  $(v_i, v_j \in V(G))$  为完全赋权图  $G$  的最短距离矩阵, 定义  $W(V(G)) \triangleq M(G) \cdot D(G)$ , 则图  $G$  的重心  $MP(G)$  是  $W(V(G))$  中最小元对应的顶点。

### 3.2 控制器组的选取算法

根据 3.1 节所述的原理, 可以将集群代理服务器系统抽象成完全赋权图: 顶点  $v_i$  由各代理服务器构成, 每个顶点上的用户请求强度和性能用顶点的权  $m_i$  表示; 边  $e_j$  由代理服务器之间的网络链接组成, 边的权  $l_j$  表示链路的传输代价和负载状况。控制器的选择问题就转化成了完全赋权图的重心求解。完全赋权图的重心对应的代理服务器将是总体代价最小的节点。由定理 1 可知, 不用增加新的节点, 即可从网络中选择合适的节点作为控制器。为避免控制节点发生单点故障, 我们从代理服务器组中优选一组 ( $z$  个) 控制器, 构成控制子系统。

**算法 2** (控制器组的选取算法)

**输入** 分布式代理缓存网络

**输出**  $z$  ( $z \geq 2$ ) 个适合作为控制器的代理服务器

**步骤 1** 将网络结构抽象成完全赋权图: 省略网络中除了代理服务器以外的所有设备, 以代理为顶点、代理之间的逻辑链路作为边, 形成连通图  $G_0$ ; 根据链路带宽和使用状况的统计值为各边赋权  $l_j$ , 根据各代理服务的访问强度和处理能力为各顶点赋权  $m_i$ , 构成完全赋权图  $G$ ;

**步骤 2** 构造完全赋权图  $G$  的邻接矩阵  $Adj(G)$ ;

**步骤 3** 使用 Floyd 算法, 根据邻接矩阵  $Adj(G)$  求解图  $G$  的最短路径矩阵  $D(G)$ ;

**步骤 4** 使用算法 1, 计算  $W(V(G)) \triangleq M(G) \cdot D(G)$ , 从中选择代价值的最小值  $w(v_k)$  对应的顶点  $v_k$  作为控制器, 将  $v_k$  加入控制子系统:  $CTRL = CTRL + \{v_k\}$ , 若控制子系统中控制器的数量  $|CTRL| \geq z$ , 则结束算法; 否则转步骤 5;

**步骤 5** 在图  $G$  中删除节点  $v_k$  以及与其相连的边:  $G = G - \{v_k\}$ , 若生成的图  $G$  依然连通, 则转步骤 2; 否则结束算法。

设系统中代理服务器数量为  $n = |V(G)|$ , 则由邻接矩阵求最短路径矩阵的 Floyd 算法时间复杂度为  $O(n^3)$ , 故算法 2 总的复杂度为  $O(n^3)$ 。

算法 2 的调度间隔 (即: 连续两次调度算法 2 的时间之差) 直接影响系统的性能。实时在线执行算法 2, 虽然能够及时反映系统的重心变化情况, 但存在如下问题: (1) 过高频率地采集和上报用户的请求强度, 耗费代理服务器大量的处理能

力; (2) 算法 2 计算复杂度大, 每次选取控制器组需要大量的系统开销; (3) 频繁更换控制子系统成员可能导致主控制器的改变, 产生系统震荡; (4) 由于切换期间无法接入新的用户请求, 频繁变更主控制器导致系统吞吐量下降。为了折中系统性能和实时性之间的矛盾, 我们设计了一种调度机制, 以下条件发生时执行一次算法 2: (1) 用户请求强度变化: 内容存储器每隔  $\sigma$  时间上报一次负载状况信息, 当发现超过比率  $\eta$  的内容存储器上的用户请求强度变化率超过设定的门限  $\tau$  时; (2) 网络拓扑变化: 当增加代理服务器数量或改变链路路由时; (3) 时间触发: 若上述两种条件均未发生, 则经过一段时间  $\Delta T$  后调度一次该算法。

## 4 主控制器选举和切换协议

3.2 节使用算法 2 选择了  $z$  个代理服务器构成控制子系统。按照选出顺序, 参照各代理服务器的处理能力, 为每个控制器设置竞选优先级  $P_i$  ( $0 < P_i < 1$ ), 并保证当  $i > j$  时  $P_i < P_j$ ; 竞选优先级是各控制器参与竞争并取得主控制器身份的决定因素。

在PMEH设计中, 我们采用IP组播思想: 控制器加入特定的组播组, 通过发送和接收IP组播报文参与选举和切换; 其优势体现在: (1) 降低了配置管理负担, 实现了集群的自发现和主控制器选举; (2) 避免了  $z$  个控制器间端到端  $z^2$  数量级的通信连接, 使用单个组播报文代替了多份单播报文的发送, 节省了通信开销; (3) 简化了协议实现的复杂度, 增加了系统的扩展性。

图 2 给出了基于优先级的主控制器竞选和切换协议 PMEH 的有限状态自动机描述。在该自动机中控制器具有 3 个状态: 后备、竞争和主控, 其中后备和主控状态都是稳态, 竞争态是临时状态; 定义了 3 个定时器: 竞选激活定时器  $T_{active}$ , 竞选报文发送定时器  $T_{elect}$  和竞选获胜检测定时器  $T_{win}$ 。我们用术语“复位定时器”表示将一个定时器的计时值清除为 0, 以触发定时器超时事件; 用“设置定时器”表示将该定时器的计时设置为指定的初始值。

下面以自动机的状态为主线描述主控制器选举和主备

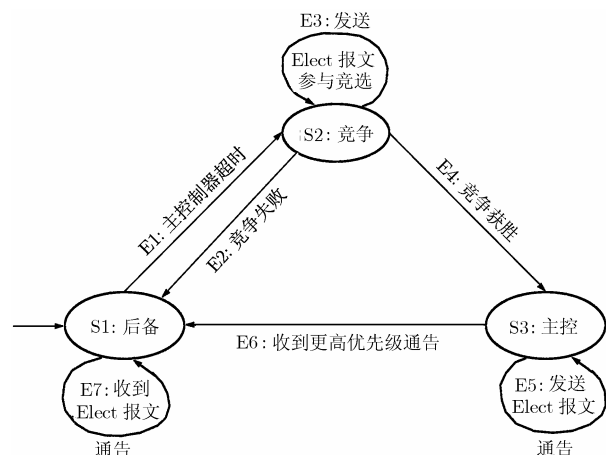


图 2 基于优先级的主控制器竞选和切换协议 PMEH 的自动机模型

切换协议:

(1)初始状态 每个控制器启动后的初始状态为“S1:后备”,并将竞选激活定时器  $T_{active}$  复位为 0;

(2)后备状态 处于“S1:后备”状态的控制器维护竞选激活定时器  $T_{active}$ ,  $T_{active}$  用于监控当前的主控制器活跃状态,接收到来自主控制器的 Elect 报文后设置  $T_{active}$  定时器;  $T_{active}$  的超时意味着当前主控制器的失效和新一轮竞选的开始,  $T_{active}$  超时后,当前控制器以组播方式在控制区域内发送主控制器竞选报文 Elect,并转移到“S2:竞争”状态,参与主控制器竞选流程。竞选报文格式为 Elect(HostId, Priority),其中 HostId 为本机的网络标识,可以是主机 DNS 域名或者 IP 地址; Priority 为竞选优先级;

(3)竞争状态 处于“S2:竞争”状态的控制器启动两个定时器:竞选报文发送定时器  $T_{elect}$  和竞选获胜检测定时器  $T_{win}$ ;定时器  $T_{elect}$  超时后,继续发送 Elect 报文参与竞选,并重新设置  $T_{elect}$ ;定时器  $T_{win}$  超时表明本控制器在当前控制区域具有最高优先级,即本控制器获得竞选胜利,转状态“S3:主控”。处于“S2:竞争”状态的控制器监听主控竞选报文,收到 Elect 后将报文中的优先级  $P_R$  与本机的候选优先级  $P_L$  作比较。若  $P_R > P_L$ ,则停止定时器  $T_{elect}$  和  $T_{win}$  以避免 Elect 报文的继续发送,并转到“S1:后备”状态;若  $P_R < P_L$ ,则继续保持本地定时器  $T_{elect}$  并设置定时器  $T_{win}$ ,准备发送 Elect 报文继续参与竞选;

(4)主控状态 处于“S3:主控”状态的主控制器使用  $T_{elect}$  定时器定期发送 Elect 报文,以告知其它控制器自己的存活状态;当收到的 Elect 报文优先级  $P_R > P_L$  时,表明网络中出现了新的主控制器,转移到“S1:后备”状态。

为了防止报文丢失导致主控制器角色动荡,我们设定  $T_{win} = mT_{elect}$ ,  $T_{active} = nT_{elect}$ ,  $3 \leq m < n$ 。对  $T_{elect}$  初始值的选择要考虑两方面因素的平衡:较小的  $T_{elect}$  值可缩短系统的收敛时间,但选择过小的  $T_{elect}$  意味着协议报文发送频率的加快,这将带来大量网络带宽消耗和控制器处理开销,影响系统性能。

从上述流程可以看出,只要控制器集合 CTRL 中存在一个以上控制器,一个选举程序总能选出本区域内优先级最高的控制器作为主控制器;主控制器失效时,在  $(m+n)T_{elect}$  时间内必能选出新的主控制器接替其工作,且新的主控制器在系统当前网络状态下的最优控制点;结合定期的主控数据备份机制,可实现主控制器上运行中业务的在线平滑切换。

## 5 模型与性能分析

### 5.1 系统可靠性

引入多控制器的集群代理结构,在线主控选举和切换机制增强了控制子系统的可靠性:设单台控制器的可靠性为  $r$ ,集群中控制器的数量为  $z$ ,则控制子系统的可靠性  $R = 1 - (1 - r)^z$ 。为了使系统达到 99.99%以上电信级软件的可靠性要

求,对单控制器可靠性  $r=0.9$  的系统,需要 4 台控制器构成控制子系统;  $r=0.7$  的子系统需要 8 台;  $r=0.5$  时则需要 14 台。

### 5.2 控制器组选取算法的性能

首先通过仿真实验测量算法 2 的调度间隔。从 300 个代理服务器中选取 8 个控制器构成控制系统;取  $\Delta T = 120\text{min}$ ,  $\sigma = 5\text{min}$ ,  $\eta = 20\%$ ,  $\tau = 10\%$ ;用户请求强度的变化模型来自文献[10]的统计数据。经过仿真时间 60 h 后,统计到算法 2 的平均调度间隔为 71.3 min。此外,我们分析了校园网上一台 VoD 服务器 1 周内的访问日志,以 72 min 为调度区间,相邻调度区间用户访问强度变化率的平均值为 5.42%。因此,认为对算法 2 实施非实时的调度机制是可行的。

然后测量 3.2 节的算法 2 的执行时间。我们在 3 种平台分别运行算法 2 测量其实际执行时间,这 3 个平台分别是:IBM R52 笔记本电脑、普通个人电脑、HP 小型机。从 300 个代理服务器中选取 8 个控制器构成控制系统。从测试结果看出,算法的每次执行时间在 2-4min 之间,与算法的平均调度间隔(71.3min)比较,其计算开销和滞后对系统的影响并不明显。

因此,3.2 节提出的非实时调度机制适合于算法 2,可以满足实际流媒体应用系统的需要。

### 5.3 消息发送间隔的选择

如 4.2 节所述,消息发送间隔参数  $T_{elect}$  的选择需要综合考虑系统收敛时间和网络开销两方面因素:选择较小的  $T_{elect}$  有助于降低主控制器竞选时延,加快网络的收敛速度;但较小的  $T_{elect}$  会增加协议报文的发送频度,提高网络代价和其它代理的处理开销。我们定义如下的规范化增益函数  $G_i$  评价  $T_{elect}$  参数对系统的总体影响:

$$G_i = \alpha \cdot \frac{T_{\max}^{cvg} - T_i^{cvg}}{T_{\max}^{cvg}} + \beta \cdot \frac{L_{\max} - L_i}{L_{\max}} \quad (1)$$

其中  $T_i^{cvg}$  表示第  $i$  次仿真试验测得的收敛时间,  $T_{\max}^{cvg}$  代表收敛时间的最大值,  $T_{\max}^{cvg} = \max_i(T_i^{cvg})$ ;  $L_i$  代表某次选举流程的代价,包括网络传输代价和节点处理代价两部分,  $L_{\max}$  是代价的最大值,  $L_{\max} = \max_i(L_i)$ 。  $\alpha$  和  $\beta$  分别代表两种增益的权值  $0 < \alpha, \beta < 1$ ,  $\alpha + \beta = 1$ ,在本系统中我们更注重缩短收敛时间,因此我们取  $\alpha > \beta$ 。

选择 8 个节点构成控制子系统,针对 3 种典型的拓扑结构进行仿真试验,  $T_{elect}$  分别选取从 1 到 30s 之间所有整数,每种情况进行 20 次仿真实验,测得收敛时间和系统开销数据的平均值。然后对式(1)中的  $\alpha$  分别取 0.9, 0.8, 0.7, 计算  $T_{elect}$  与  $G$  之间的关系。图 3 给出了其中一种拓扑下的报文发送间隔  $T_{elect}$  与规范化增益  $G$  的关系图,可以看出  $T_{elect}$  的最优取值区域应该在 3s-6s 之间。针对其它两种拓扑结构的仿真实验获得完全相同的结论。

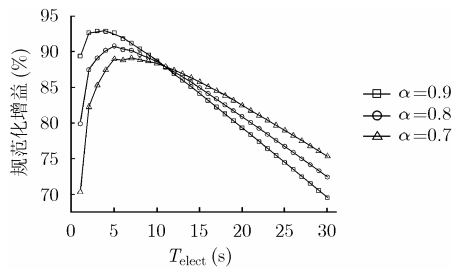


图3 参数  $T_{\text{elect}}$  与规范化增益  $G$  的关系

## 6 结束语

本文提出的基于多控制器集群的流媒体缓存代理结构 MCSCP, 采用控制器之间的协同工作、基于优先级的主控制器选举和在线切换、存储与控制分离等机制, 提高了系统的可靠性和伸缩性, 对大规模分布式流媒体系统具有一定的参考价值。后续工作中, 我们将研究存储子系统中分布式缓存策略, 主要包括: 缓存替换、缓存复制和缓存迁移等方面问题。

本文的贡献在于: (1)提出了一种基于多控制器的流媒体缓存代理结构 MCSCP, 控制子系统与存储子系统分离增加了系统的伸缩性, 基于优先级的主控制器选举和在线切换机制提升了系统可靠性; (2)使用图论思想设计了一种选取一组代理服务器构成控制子系统的算法, 降低了系统的通信代价; (3)提出了控制子系统中主控制器基于优先级的选举和在线故障切换协议 PMEHL, 该协议具有较强的灵活性和鲁棒性, 适应于网络结构的动态变化; (4)建立数学模型分析了 MCSCP 结构对系统可靠性的提高程度, 讨论了协议中的关键参数  $T_{\text{elect}}$  对系统性能的影响, 并使用仿真实验给出了该参数的最优取值范围。

## 参考文献

- [1] 钟玉琢, 向哲, 沈洪. 流媒体和视频服务器. 北京: 清华大学出版社, 2003: 83-110.
- [2] Hua K A, Cai Y, and Sheu S. Patching: A multicast technique

for true video-on-demand services. Proc. of ACM Multimedia '98, Bristol UK, 1998, 9: 35-43.

- [3] Wang B, Sen S, and Adler M, *et al.* Optimal proxy cache allocation for efficient streaming media distribution. *IEEE Trans. on Multimedia*, 2003, 6(2): 366-374.
- [4] Rizzo L and Vicisano L. Replacement policies for a proxy cache [J]. *IEEE/ACM Trans. on Networking*, 2000, 8(2): 158-170.
- [5] Wu K L, Yu P S, and Wolf J L. Segment based proxy caching of multimedia streams. Proc. of the 10th International Conference on World Wide Web, Hong Kong China, 2001,5: 36-44.
- [6] Chen S, Shen B, and Wee S, *et al.* Adaptive and lazy segmentation based proxy caching for streaming media delivery. Proc. of the 13th international workshop on Network and operating systems support for digital audio and video, Monterey USA, 2003, 6: 22-31.
- [7] Acharya S and Smith B. MiddleMan: A video caching proxy server. Proc. of the 10th International Workshop on Network and Operating System Support for Digital Audio and Video, Chapel Hill USA, 2000, 6: 163-172.
- [8] 郭常杰, 向哲, 钟玉琢. 一种新的基于分区的多媒体代理协作管理策略. *计算机研究与发展*, 2002, 39(11): 1505-1512.
- [9] 杜端甫. 运筹图论. 北京: 北京航空航天大学出版社, 1990: 259-269.
- [10] 向哲, 钟玉琢. 流调度算法验证平台的设计与实现. *小型微型计算机系统*, 2000, 21(12): 1237-1239.

杨波: 男, 1973年生, 博士生, 研究方向为宽带通信网、流媒体技术.

廖建新: 男, 1965年生, 教授, 博士生导师, 主要研究方向为移动智能网、宽带智能网、下一代网络技术.

朱晓民: 男, 1974年生, 副研究员, 中国电子学会高级会员, 中国通信学会高级会员, 主要研究方向为智能网、下一代业务网络、流媒体技术.