

# 基于流媒体的小型分布式视频点播系统研究

朱晓辉, 陈苏蓉

(南通大学 计算机科学与技术学院, 江苏 南通 226019)

**摘 要:** 针对视频点播系统中存在的高带宽、高存储、低并发性等问题, 提出了一种基于流媒体的小型分布式的视频点播系统, 通过把视频进行分段并动态分布到不同的服务器实现了分布式存储. 通过文件共享机制并在数据库中纪录和维护视频信息实现对不同视频服务器上视频资料的统一管理. 仿真测试结果表明, 该方案能大大提高视频点播系统的并发访问量, 并可动态添加视频服务器以满足日益增长的视频点播需求.

**关键词:** 流媒体; 分布式存储; 视频点播

中图分类号: TN919.85

文献标识码: A

文章编号: 1000-7180(2010)10-0069-05

## The Research of Simply Distributed Video on Demand System Based on Stream Media

ZHU Xiao-hui, CHEN Su-rong

(College of Computer Science and Technology, Nantong University, Nantong 226019, China)

**Abstract:** To the problems of high bandwidth, high storage and low concurrency, a technology of simply distributed VOD system which is based on stream media was proposed. By the means of dividing video into several sections and deploying them to different video servers, solved the issue of distributed storage. By the means of file sharing and recording the video information to database, solved the issue of unified management for video information. Experiments show that the solution can service more clients, and can add additional video servers to meet the increasing demands.

**Key words:** stream media; distributed store; Video on Demand

### 1 引言

视频点播 VOD (Video on Demand) 系统是网络多媒体中一类重要的应用, 它通过网络为用户提供包括视频、音频等媒体数据的点播业务<sup>[1]</sup>. 它是近年来新兴的一种网络传媒方式, 是计算机技术、网络通信技术、多媒体技术、数字压缩技术等多学科、多领域融合交叉的一门综合性技术. 随着 VOD 技术的发展和普及, 出现了许多像“优酷网”、“土豆网”等大型网络视频站点. 这类站点都保存了海量视频信息, 为了满足高并发量的用户点播需求, 这类大型视频网站一般会采用 CDN、NFS、P2P、负载均衡等各

种技术来提高系统的并发负载能力<sup>[2]</sup>. 然而, 采用这些技术需要昂贵的设备和雄厚的技术力量的支持, 对于一般小型的视频点播系统而言, 由于缺少资金、设备和技术的支持, 因此无法采用这些技术来有效提升视频网站的并发用户数.

对此, 提出了适用于小型视频点播系统的基于流媒体的分布式视频点播系统解决方案. 该方案实现简单, 并且只需增加很少的硬件成本, 即可有效提升点播系统的并发用户数, 并且可以根据需要动态添加视频服务器, 确保系统具有可扩展性和伸缩性<sup>[3]</sup>.

收稿日期: 2009-09-13; 修回日期: 2010-11-13

基金项目: 南通市重大科技专项基金项目(XA2008004); 上海市信息安全综合管理技术研究重点实验室开放课题(AGK2009006)

## 2 系统性能瓶颈分析

视频点播能支持的最大并发数是评价该系统优劣的一项重要性能指标,因此首先需要找出影响该性能指标的因素。这里通过 VC++ 开发了一个视频点播测试程序,该程序利用多线程技术来模拟和仿真多用户并发点播视频的情形。程序只是从视频服务器下载流媒体而不进行真正的播放,这样可以大大减小测试程序对客户端资源的占用,使得一个客户端程序可以模拟上百个客户端的并发点播情况。通过仿真测试我们发现:影响这一指标的性能瓶颈主要在两个方面,一是网络带宽,二是磁盘 I/O 性能。

### 2.1 网络带宽

视频点播系统对网络的带宽要求较高,流媒体节目的高码率特性会给网络带来很大压力,影响系统的并发性能。虽然最新的流媒体技术可以达到较高的压缩比,但网络带宽依然是 VOD 系统的主要性能瓶颈之一。通过仿真程序测试可知,利用 MPEG-4 技术在网络上传递 DVD 画质的流媒体视频需要的带宽约 700kb/s,当有 100 个用户并发访问同一台视频服务器时就需要 70Mb/s 的带宽资源。而在仿真测试过程中发现,100Mb/s 网卡带宽的实际有效利用率仅为 60% 左右,即 100Mb/s 的网卡只能负担 60Mb/s 左右的实际网络流量。因此对于高并发的视频点播系统需要采用多个视频服务器来应对用户并发访问量的不断增长,减少对单个服务器的带宽压力。

### 2.2 磁盘 I/O

用户在选择某个视频进行播放后,视频服务器需要做如下几步处理:首先从硬盘上读取视频数据并通过硬盘接口电路传送到系统总线,然后数据被送到内存中,再由 CPU 对数据进行切分、封装等处理,最后发送到网卡,由网卡对数据进行包装后发送到外部网络。这里所涉及到的设备中硬盘的数据传输速度是最慢的,因此磁盘的 I/O 性能在很大程度上也决定了视频点播服务器的性能<sup>[4]</sup>。普通 IDE 接口硬盘的稳定数据传输率一般在 30MB/s 到 45MB/s 之间。但当有多个用户并发点播不同的视频时,硬盘需要同时读取多个不同的视频信息来满足不同用户的点播需求。由于磁盘所固有的机械特性,因此很难满足这样高并发的点播需求。从仿真测试结果中也印证了这点,当有多个线程同时请求硬盘上不同视频时,硬盘的实际传输速率呈现下降趋势,线程数越多下降得越快。

## 3 系统架构设计

针对以上通过仿真测试得到的结果,系统架构的设计需要解决磁盘 I/O 和网络带宽这两个 VOD 系统的性能瓶颈问题。因此需要采用多视频服务器系统来分担磁盘 I/O 和系统带宽压力。系统架构如图 1 所示,包括了客户端、Web 服务器、数据库服务器和视频服务器组四个部分。

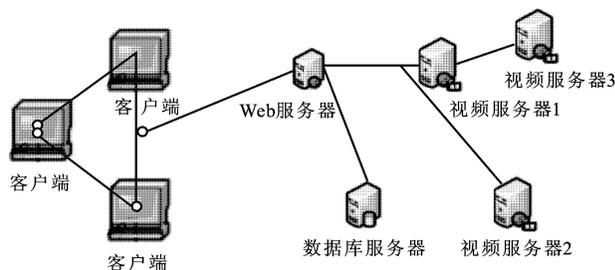


图 1 系统架构

客户端是指 IE、Firefox 等常用的客户端浏览器,用户可以通过客户端来浏览网站的视频信息,并进行视频点播操作,网站管理员也可以通过客户端上传和管理视频信息。

Web 服务器部署视频网站的 Web 页面,用于响应客户端用户的请求。当用户浏览网页时,Web 服务器响应请求并访问数据库服务器,获得相应的视频信息并生成页面返回给客户端,客户端点击某个视频链接后,Web 服务器访问数据库并分析该链接所对应的视频服务器地址,然后把对应的视频服务器地址返回视频地址给客户端浏览器,客户端浏览器开始接受视频地址并播放该视频。

数据库服务器用于纪录所有视频的 URL 地址以及所有视频服务器的地址和目录位置等信息,同时需要纪录所有视频服务器的配置及当前状态信息。为了进一步提高 VOD 系统的性能,使用 RamDisk 技术在数据库服务器上建立内存数据库系统,把数据库系统直接映射到内存中,从而极大地提升了数据库访问性能。

视频服务器组用于存储所有视频信息,视频服务器数量可以根据需要动态增加。这里采用普通 PC 机作为视频服务器,而不是昂贵的专用视频服务器。考虑到磁盘 I/O 性能对整个系统的影响,为了获得较好的性价比,存储系统采用 IDE RAID 硬盘,并组成 RAID1 的存储架构<sup>[5]</sup>。

## 4 系统实现及关键技术

用多个视频服务器组成分布式视频存储系统

后,对于客户端而言,整个视频点播系统执行过程应该仍然是透明的,无需关心视频具体存储在哪个服务器上。但后台系统需要解决以下几个问题:一是如何实现视频分布式部署,即管理员在视频上传时如何动态确定保存到哪个视频服务器;二是如何尽可能做到视频服务器的负载均衡,既要保证所有视频服务器都有均等的机会来保存视频资料又要考虑到不同服务器的硬件配置和性能差异来区别对待;三是如何把一台视频服务器上视频资料均衡保存到多个不同的物理硬盘中以便充分利用磁盘的 I/O 性能;四是如何根据性能需要和视频数量的增加实现视频服务器的动态扩充。

#### 4.1 视频服务器表及存储信息表

Web 服务器需要掌握所有视频服务器的状态和信息才能在管理员上传视频信息时动态决定把视频保存到哪一台视频服务器,因此需要在数据库中纪录所有的视频服务器信息,纪录视频服务器的表如表 1 所示。

表 1 视频服务器表

字段名	类型	备注
ServerId	int	自增字段,主键
ServerName	nvarchar(50)	视频服务器名称
IPAddress	nvarchar(100)	视频服务器 IP 地址
MaxAmount	int	可存放最多视频数
CurAmount	int	当前已保存视频数
FlgUsable	boolean	服务器是否可用

视频服务器表中的 ServerId 字段为主键自增列,唯一代表一台视频服务器。ServerName 字段纪录服务器的名称,方便管理员识别该纪录代表哪台服务器。IpAddress 表示该视频服务器的 IP 地址。MaxAmount 字段表示视频服务器能保存的最大视频数,该字段的值可以根据服务器的硬件配置和性能以及 VOD 系统的实际需要而进行动态调整。CurAmount 字段表示当前已保存的视频数,当 CurAmount  $\geq$  MaxAmount 时系统将不再把视频上传到该服务器。

考虑到磁盘 I/O 性能瓶颈问题,需要为视频服务器配置多个物理硬盘,并把视频信息尽可能均匀分布到不同的物理硬盘来分担 I/O 访问请求,因此,需要在数据库中建立一个视频服务器存储表,详细纪录每个视频服务器的所有物理硬盘信息,其格式如表 2 所示。

表 2 视频服务器存储表

字段名	类型	备注
DiskId	int	自增字段,主键
ServerId	int	视频服务器表外键
DiskName	nvarchar(50)	视频服务器磁盘名
DiskUrl	nvarchar(100)	服务器磁盘 Url
MaxAmount	int	可存放最多视频数
CurAmount	int	当前已保存视频数

存储表中的 DiskId 字段为主键自增列,唯一代表视频服务器中的一个物理硬盘纪录。DiskName 字段纪录视频服务器中物理硬盘的名称,方便管理员识别该纪录代表哪个物理硬盘。DiskUrl 字段标识了服务器上物理硬盘在 Web 服务器中的 URL 路径。MaxAmount 字段表示该硬盘能保存的最大图片数,该数可以根据硬盘的 I/O 性能进行动态调整。CurAmount 字段表示该硬盘当前已保存的视频数,当 CurAmount  $\geq$  MaxAmount 时系统将不再把视频上传到该服务器。

#### 4.2 播放视频

客户端用户通过浏览器向 Web 服务器发出查询视频信息的请求,Web 服务器从数据库服务器中获取相应的视频信息,并将该视频信息直接返回给客户端。用户选择某个视频信息,向 Web 服务器发送播放该视频的请求,Web 服务器访问数据库服务器分析该视频所在的视频服务器地址及硬盘位置,并动态转换成该视频的 URL 地址,最后把该 URL 地址返回给客户端的视频播放器,视频播放器根据该 URL 地址向相应的视频服务器发送连接请求并开始播放视频。其视频播放的算法如图 2 所示。

#### 4.3 上传视频

由于网站本身的技术限制,视频资料无法通过 Web 服务器直接上传到不同的视频服务器,因此需要在所有视频服务器上部署一个 Web Service<sup>[6]</sup>以便 Web 服务器可以通过调用不同视频服务器上的 Web Service 而执行保存或删除视频的操作。

视频上传的过程比较复杂,首先 Web 服务器接收网站管理员的视频上传请求并访问数据库,查询视频服务器表筛选出可用的视频服务器集合记作 C,集合中服务器数量记作 N。然后根据可用的视频服务器集合 C 查询视频存储表,获得所有可用的视频存储位置集合记作 P。为了尽可能减少磁盘的 I/O 压力,系统把一个完整的视频切分成多个分段,然

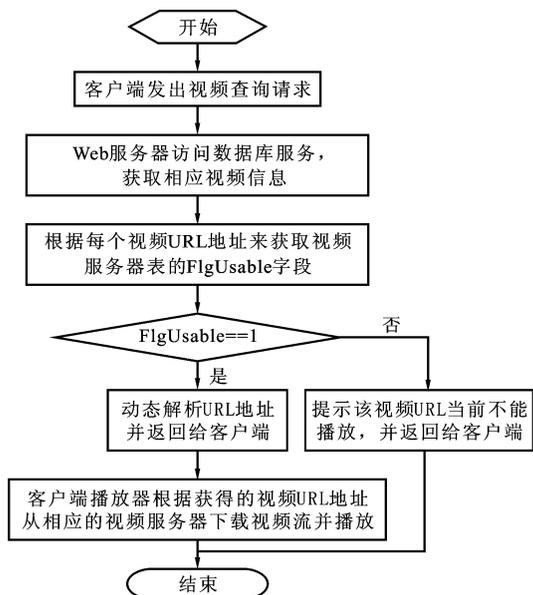


图2 视频播放算法

后把每个分段依次上传到不同的视频服务器及不同的物理硬盘上, 从而进一步提升系统的并发性能。为了把视频资料均匀分布到各视频服务器的物理硬盘, 系统采用随机函数的方式来随机选择要保存的视频服务器和物理磁盘。其基本算法是: 用随机函数产生一个随机数  $R_1$ , 并用  $R_1$  与  $N$  进行取余运算, 记作  $I = R_1 \% N$ , 则  $C[I]$  为要保存的视频服务器。再获取  $C[I]$  服务器上的磁盘信息, 并利用随机函数确定要保存的磁盘, 为了确保上传的视频文件名不重复, 以当前时间+ 随机数的形式组成视频文件名。

以上可以看出, 通过利用随机函数取值的随机性和取余运算, 使每台视频服务器以及同一台服务器上的所有物理磁盘都有均等的机会来保存视频。因此, 所有视频及其视频分段都是被随机保存到不同视频服务器的不同物理磁盘中, 很好地实现了视频的分布式部署和负载均衡。同时网站管理员也可以通过设定服务器信息表及存储信息表中的“MaxAmount”字段的值来限定整个服务器所能保存视频的最大数和单个物理磁盘能保存视频的最大数, 从而能够根据服务器的硬件配置和性能差异等因素来决定该服务器能保存的视频数量, 因此进一步提升了整个视频服务器集群的负载均衡能力。同时一个视频又被划分成多个分段, 并随机保存到不同的视频服务器及不同的物理硬盘上, 因此有效减缓了新上传的视频因为点播量过大而导致某台视频服务器过载的情况。当需要增加视频服务器时, 也只需在服务器信息表及存储信息表中增加相应的纪录, 添加

视频服务器的过程对整个网站系统的运行没有任何影响, 从而实现了视频服务器的动态增加。用户上传视频的算法如图3所示。

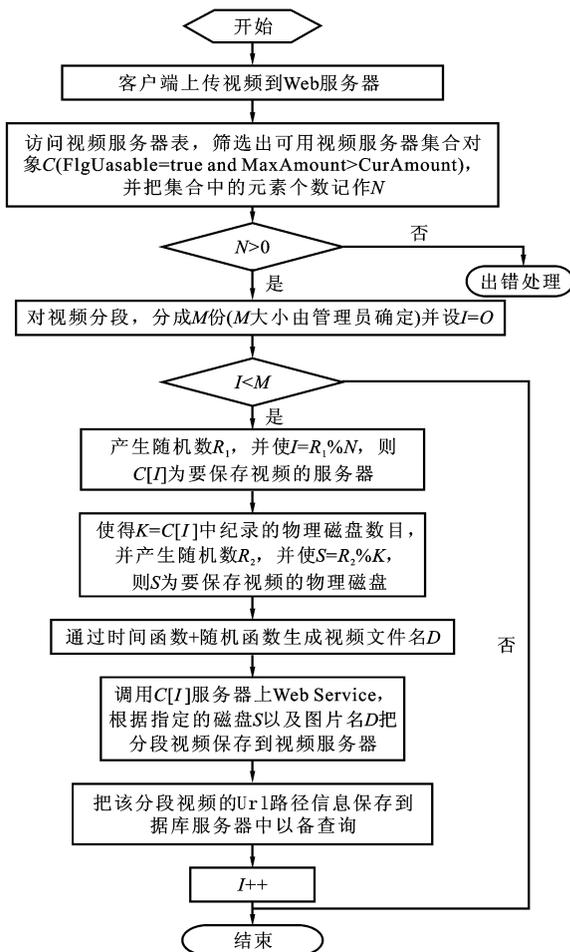


图3 上传视频算法

#### 4.4 删除视频

客户端向 Web 服务器发出删除某个视频的请求, Web 服务器接受请求并搜索数据库获取待删除视频所有分段的 URL 信息。依次将该 URL 信息通过字符串运算分隔为视频服务器的 URL 根路径  $R$ 、视频所存放的磁盘名称  $D$  和视频名称  $N$ 。再查找数据库中的服务器信息表, 获取与  $R$  匹配的纪录记作  $C$ ,  $C$  即为要删除视频的视频服务器。然后调用  $C$  视频服务器上的 WebService 方法, 并以视频名称  $N$  和视频所存放的磁盘  $D$  为参数调用该方法删除视频分段, 最后将该视频纪录从数据库服务器中删除。用户删除视频信息的算法如图4所示。

#### 5 系统仿真测试

在局域网环境中, 对采用视频服务器群组和普通情况下采用单台高性能视频服务器两种情况进行

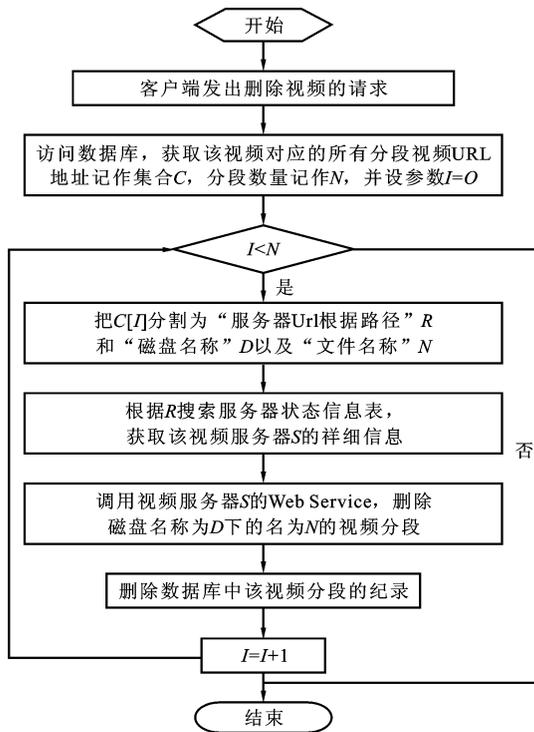


图4 删除视频算法

了性能仿真测试. 硬件配置如下: Web 服务器、数据库服务器各一台, Pentium3.0GHz, 内存 2GB, 网络带宽 100Mb/s. 客户端机器 10 台为, Pentium 3.0GHz, 内存 1GB, 网络带宽 100Mb/s. 普通视频服务器 3 台, 为普通的 PC 机, CPU: Intel 双核 P2.0GHz, 内存 2GB, 网络带宽 100Mb/s, 四块 IDE RAID 硬盘每块容量 160GB, 采用 RAID 1 存储结构. 高性能视频服务器一台, 配置为 CPU: Intel Xeon 四核 3GHz, 内存 6GB, 网络带宽 1Gb/s, SCSI 硬盘 160GB\* 4. 为了保证测试数据的精确性, 把一部时长为 113 分钟, 片源平均码率为 750kb/s 的高清流媒体电影分别复制了 500 份保存在一台高性能视频服务器及均匀分布在三台视频服务器上. 在 10 台客户端上同时运行该前文中的视频点播测试程序, 来仿真 10~ 500 个并发用户的点播请求, 分别测试系统的缓冲时间和 CPU 资源占用率, 仿真测试结果如图 5 和图 6 所示.

从图 5 和图 6 可以看出, 采用 3 台普通 PC 机作为视频服务器与采用单台高性能视频服务器相比, 在并发连接数较少时, 无论在 CPU 资源占用率还是缓冲时间都无明显的优势. 但并发连接数超过 100 后, 分布式存储的优势得到了明显体现, 视频缓冲时间大大优于单服务器模式, 3 台 PC 服务器的 CPU 平均占用率也远低于单台高性能视频服务器, 当并发连接数达到 500 个时, 单服务器模式下视频缓冲

时间大大增加 CPU 满负荷运转, 播放时频繁出现画面停顿现象, 而多服务器模式下播放画面依旧清晰稳定. 通过实验可以发现, 以上设计方案能有效提升 VOD 系统的并发用户数, 提升整个 VOD 系统的性能, 而且并发访问量越大, 性能的提升越明显, 而对于整个系统而言增加的硬件成本却很有限.

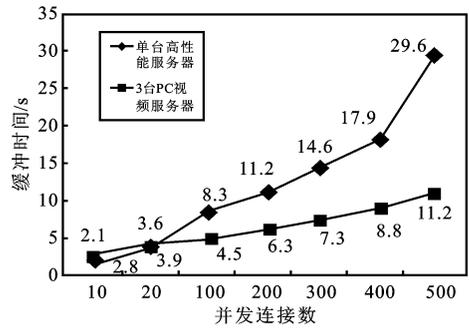


图5 缓冲时间

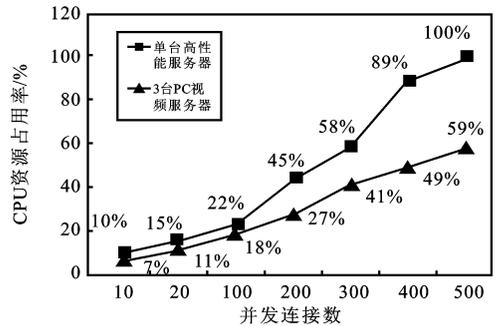


图6 CPU占用率

## 6 结束语

面对日益增长的视频点播系统的并发访问量, 设计并实现了一种基于流媒体技术的小型分布式视频点播系统. 详细论述了视频分布式存储、数据库结构设计及相关查询、播放算法等关键技术. 通过仿真测试的性能分析数据可知, 该解决方案简单易行, 只需增加很少的硬件成本即可有效提升视频网站的并发访问性能.

## 参考文献:

- [1] 张正华, 余松煜, 陈军华, 等. 一种小型视频点播系统中的多客户管理方案[J]. 微电子学与计算机, 2005, 22(8) 144-150.
- [2] Laoutaris N, Zissimopoulos V, Stavrakakis I. Joint object placement and node dimensioning for internet content distribution[J]. Information Processing Letters, 2004, 89(6): 273-279.

(下转第 78 页)

其中输入为  $u(t)$ , 输出为  $y(t)$ ,  $\theta_1, \theta_2, \theta_3, \theta_4$  是待估参数。

可以归结为对性能指标函数  $f = \sum (\hat{y}_i - y_i)^2$  ( $y_i$  为系统的真实输出,  $\hat{y}_i$  为搜索参数下的模型输出) 的优化问题, 它是一个非线性、多极值函数。根据模型特点, 参数  $\theta_1, \theta_2$  必须小于 1, 否则发散。本文在估计时, 输入  $u(t)$  用正弦函数  $\sin(t)$ , 在一个周期内采样 50 个输入输出对, 算法参数设置 Visual = 0.1, Step = 0.01, 最大迭代次数 200, 其他参数设置同前。系统参数估计结果如表 2 所示。

表 2 非线性系统参数估计

参数	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$f$
真实值	0.5	0.3	1.8	0.9	0
文献[8]结果	0.491 6	0.301 41	1.843 2	0.926 7	0.049 3
ACMAFSA 结果	0.492 2	0.302 0	1.799 9	0.925 3	6.9157e-006

由此可见 ACMAFSA 估计的结果也是相当有效的。

## 6 结束语

本文充分利用公告板中的历史最优鱼和柯西变异的优点, 提出自适应柯西变异人工鱼群算法, 用不同的测试函数对 ACMAFSA 和 AFSA 进行了仿真试验及实例应用研究, 并对收敛性和参数进行理论分析, 结果表明 ACMAFSA 提高了收敛速度、求解精度, 算法更加稳定。

## 参考文献:

- [1] 李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22(11): 32-38.
- [2] 张梅凤, 邵诚, 甘勇, 等. 基于变异算子与模拟退火混合的人工鱼群优化算法[J]. 电子学报, 2006, 34(8): 1381-1385.
- [3] 曲良东, 何登旭. 改进的人工鱼群算法及其在近似求导中的应用[J]. 微电子学与计算机, 2009, 26(5): 122-125.
- [4] 王梓坤. 概率论基础及其应用[M]. 北京: 科技出版社, 1979.
- [5] 郑小霞, 钱锋. 一种改进的微粒群优化算法[J]. 计算机工程, 2006, 32(15): 25-27.
- [6] 吴红亮, 王耀南, 周少武, 等. 双群体伪并行差分进化算法研究与应用[J]. 控制理论与应用, 2007, 24(3): 453-458.
- [7] 左兴权, 李士勇. 一种新的免疫进化算法及其性能分析[J]. 系统仿真学报, 2003, 15(11): 1607-1609.
- [8] 姜波, 王秉文. 基于遗传算法的非线性系统模型参数估计[J]. 控制理论与应用, 2000, 17(1): 150-152.

## 作者简介:

- 曲良东 男, 硕士. 研究方向为智能优化算法设计与分析和人工神经网络.  
何登旭 男, 教授. 研究方向为智能优化算法设计与分析和人工神经网络.

(上接第 73 页)

- [3] 郑海洋, 宿红毅, 李毅. 分布式视频点播网络统计调度策略及负载均衡研究[J]. 计算机工程与应用, 2005, 8(16): 165-167.
- [4] 刘楠, 付先平, 张维石. 基于 .NET 的视频点播系统的设计与实现[J]. 计算机工程, 2006, 32(24): 213-215.
- [5] 基于 P2P 的视频点播系统综述[J]. 计算机学报, 2010, 33(4): 613-621.

- [6] 毕敬, 朱志良, 铁鸣. 基于 Web services 的分布式企业信息整合模型[J]. 计算机工程, 2008, 34(12): 280-282.

## 作者简介:

- 朱晓辉 男, (1976-), 硕士, 讲师. 研究方向为计算机软件与理论.  
陈苏蓉 女, (1977-), 硕士, 讲师. 研究方向为计算机应用.