

基于热点影片的峰值服务保障服务器集群系统

何统洲¹, 王卫东², 黄浩²

(1. 鄱阳师专物理与电子工程系, 丹江口 442700; 2. 华中科技大学计算机学院, 武汉 430074)

摘要:通过研究用户对热点对象的访问模式, 优化传统的保障热点影片峰值服务的调度算法, 设计一种对热点影片具有峰值服务保障能力的高性能视频服务器集群系统, 论述系统体系结构和各模块的功能, 给出软件实现的基本过程。实验表明, 热点影片的并发量阈值设定在 80~100 之间较为合适。

关键词: 视频服务; 集群系统; 热点影片; 峰值服务

Server Cluster System of Peak Service Ensurance Based on Hot Video Objects

HE Tong-zhou¹, WANG Wei-dong², HUANG Hao²

(1. Department of Physics and Electronic Engineering, Yunyang Teacher's College, Danjiangkou 442700;

2. Department of Computer, Huazhong University of Science and Technology, Wuhan 430074)

【Abstract】 This paper optimizes the traditional scheme to ensure the peak service of hot video objects by researching the clients' accessing pattern to popular video objects, and designs a high performance cluster video server to ensure the peak service of hot video objects. It discusses the architecture and module functionality of the system, and provides the processing of software implementation. Test results show that the hot video concurrent threshold between 80 and 100 is appropriate.

【Key words】 video service; cluster system; hot video objects; peak service

1 概述

视频点播业务是未来宽带网络中最典型的应用之一^[1]。设计高性能的流媒体服务器系统是目前面临的关键问题, 许多公司和研究机构都提出了可扩展的高性能服务器集群系统。这些集群系统大多通过一个前端机将外来的网络服务请求调往一组后端的真实服务器, 在数据链路层、网络层或应用层进行调度, 以实现负载均衡^[2]。LVS(Linux Virtual Server)是一个基于Linux操作系统的集群调度软件, 它通过扩展IP协议栈, 对不同的网络服务提供多种请求调度方案^[3]。LVS为构建高扩展性和高可用性的大规模服务器集群系统提供一个基本架构, 研究人员可以在此架构的基础上进行相应的扩展, 开发符合特定需求的高性能服务器集群系统, 基于此架构开发了前端机上的重定向模块和内核连接模块。同时, 该系统的设计目标是在最热门的电影达到峰值并发量时, 系统仍有足够的服务能力, 因此, 在实际系统的实现中, 常预留单台服务器承担最热门电影在峰值阶段的点播任务, 研究发现, 任何一部最热门电影发布后, 其点播热度将遵循一定的热度曲线^[4], 为此, 要对传统的保障热点影片峰值服务方法进行改进。测试结果表明, 该系统具有很高的吞吐能力和良好的扩展性和可用性。

2 视频服务器集群系统体系结构

如图 1 所示, 从硬件配置上看, 视频服务器集群系统由一个前端机和若干后台真实服务器及一个磁盘阵列组成, 它们之间通过 Hub 或交换机互连。客户端发送的请求到达前端机后, 采用 DR(Direct Routing)方式的负载均衡策略被转发到

相应的真实服务器, 真实服务器将应答消息直接发送给客户端。

从软件实现的角度考虑, 整个系统分为 3 个模块:

(1)重定向模块。该模块实现于 Load Balance 上, 负责对用户请求选择适当的真实服务器, 并记录映射关系, 完成对用户请求的重定向。

(2)内核连接模块。该模块被加入到 IPVS(IP Virtual Service)调度策略的 Linux 内核模块中, 按照重定向模块中记录的映射关系完成服务连接。

(3)热点服务器维护模块。根据优化的峰值服务保障算法, 维护热点服务器的内存映射记录。

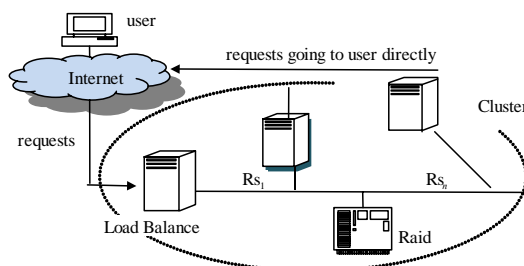


图 1 服务器集群体系结构

基金项目: 国家自然科学基金资助项目(605731120)

作者简介: 何统洲(1968 -), 男, 副教授、硕士, 主研方向: 集群与分布式计算, 存储技术; 王卫东, 博士研究生; 黄浩, 教授、博士
收稿日期: 2007-09-30 **E-mail:** ztust@yahoo.com.cn

3 优化的峰值服务保障算法

本文要求：在最热门的电影达到峰值并发量时，系统仍有足够的服务能力。常用的方法是预留单台服务器承担最热门电影在峰值阶段的点播任务^[5]。研究发现，任何一部最热门电影发布后，其点播热度将遵循一定的热度曲线(见图 2)，因此，就要对传统的峰值服务保障算法进行优化。

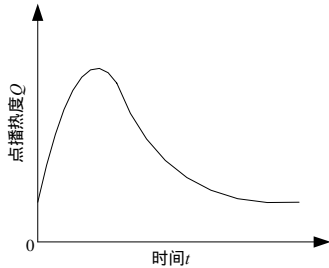


图 2 热度曲线

一部新片发布之后，点播热度会由初始值很快上升到峰值，然后开始回落并缓慢趋于平衡。可以预见，影片保持峰值并发量的时间不会很长，所以，空出一台服务器来提供峰值流量的时间也不必太长。

基于这种考虑，对原来的峰值服务保障方法进行优化：

(1)基本保持原有策略，初始状态下各台真实服务器为空，由 Raid 提供用户第一次点播，并向真实服务器传送影片(今后同样影片的点播将由真实服务器提供)，Balance 和 Raid 每次选择负载最轻的真实服务器进行传送。

无明显大片出现时，各台真实服务器的负载状况应该差不多，如图 3 中的状态 a，阴影部分为各台服务器的负载。

(2)当一部大片发布之后，承担这部电影点播任务的服务器(不妨设为 Rs_1)的并发量将会明显上升，如图 3 状态b中的B区域。

设定阈值 H_x ，当影片 X 的并发量超过 H_x ，认定 X 为当前最热门影片，需要腾出 Rs_1 ，为峰值并发量预留空间。为了平滑切换过程，使已连接到 Rs_1 的用户服务不被中断，Balance将对图 3 状态b中 Rs_1 上的A区域影片作标记(使已连接到 Rs_1 的用户仍由 Rs_1 提供服务)，之后，若再有用户点播A区域的影片，将由Raid提供服务，并向其他 3 台服务器中负载最轻者传送。在影片 X 并发量达峰值时，各服务器负载状况如图 3 中的状态c。

(3)当最热门大片 X 发布一段时间，热度达到峰值后开始回落，设定阈值为 H_x ，当 X 的并发量低于 H_x 时，Balance把 Rs_1 重新设定为正常状态，并开始为其他影片点播提供服务，各服务器负载状况又重新回到图 3 状态a中的情况。

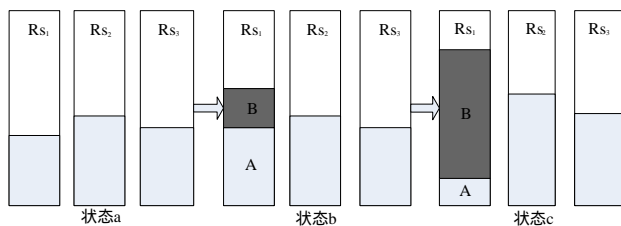


图 3 真实服务器的负载状况

4 系统实现

4.1 重定向模块的实现

在 Load Balance 上装有 Apache 服务器(Linux 主机)，能

够接收并处理 Http 请求。对于来自用户的请求，将其转到重定向模块进行处理。实现重定向模块的流程见图 4。

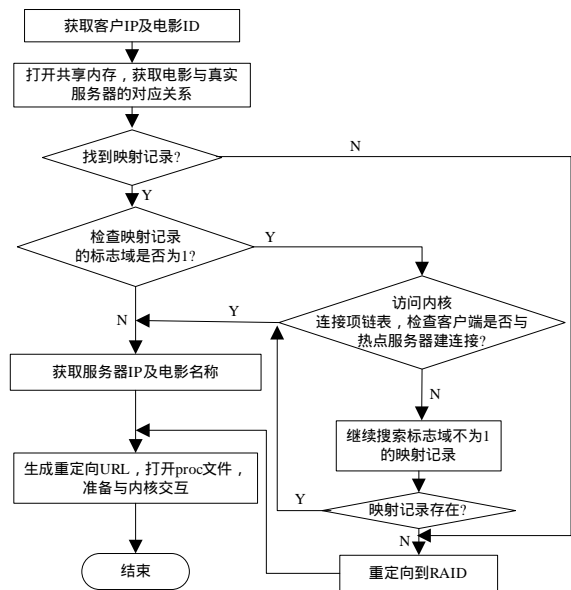


图 4 重定向模块的实现流程

4.2 内核连接的实现

内核连接模块需要被加入到 IPVS 调度策略的 Linux 内核模块中，使用 IPVS 需要重建 Linux 内核，以加入对 IPVS 的支持。这里，使用当前最新的 Linux 2.4.27 内核，其中已包含对 IPVS 的支持，只需要进行相应的配置即可。接下来要实现符合 IPVS 接口的负载均衡策略内核模块，当 IPVS 发现一个新的服务请求后，会调用与此服务绑定的负载均衡策略内核模块(绑定用 ipvsadm 进行)，得到真实服务器的 IP 地址。负载均衡策略内核模块的均衡策略函数原型为：struct ip_vs_dest *ip_vs_schedule(struct ip_vs_service *svc, struct iphdr *iph)。实现内核连接模块的具体步骤如下：

步骤 1 处理写入内核的源(真实地址信息)，并把它加入到连接项链表中。

步骤 2 创建 proc 文件系统的入口，并设置 proc 的读函数、写函数。

步骤 3 设置定时器处理函数，将超时的连接映射关系删去，重置定时器。

步骤 4 当 IPVS 中的服务绑定到本均衡策略时，进行相应的初始化处理。

步骤 5 当 IPVS 收到与本均衡策略相关的服务请求时，在连接映射表中定位客户端的 IP 地址，并返回真实服务器的 IP 地址。

步骤 6 当 IPVS 中的服务解除绑定到本均衡策略时，进行相应的清理工作。

4.3 优化的峰值服务保障方法的实现过程

设并发量阈值为 H_x ，如果一台真实服务器中的某部影片 X 的并发量达 G_x (每部影片的并发量由业务统计程序采集)，那么：

(1)当 $G_x > H_x$ 时，打开共享内存，将该服务器中除 X 外所有影片在共享内存中的映射记录的标志域置为“1”(表明该服务器只对访问影片 X 和已经与该服务器建立连接的用户提供服务)。

(2)当 $G_x < H_x$ 时，打开共享内存，检查共享内存中的映

射记录,将映射记录的标志域置为“0”,使该服务器为其他影片提供服务。对于热点服务器中不存在的影片,可以由Raid传送至该服务器,也可以将其他真实服务器中的影片传至该服务器,在影片传递结束后,应在共享内存中做相应的更新处理。

5 系统性能测试

5.1 实验环境

采用100 Mb/s的局域网,测试系统共由5台服务器组成,其中一台为前端机,前端机的操作系统为Linux2.4.27,该内核已经包含了对IPVS的支持;还有一台服务器为存储服务器(Raid),在存储服务器上安装了apache服务器,并对samba服务器进行了配置。Raid上存储了20部影片,这些影片ID为1,2,...,20,其他为真实服务器,真实服务器的操作系统为Windows 2003 server,各真实服务器的ID号分别为a,b,c,在各真实服务器上还安装了HelixServer9.0.2流媒体服务器软件。服务器的硬件配置基本相同,包括1个Pentium4 3 GHz的CPU,256 MB内存和80 GB的硬盘。各服务器的IP配置如下:Balance服务器的IP为202.103.52.66,192.168.2.14,Raid服务器的IP为192.168.2.2,服务器a的IP为192.168.2.3,服务器b的IP为192.168.2.4,服务器c的IP为192.168.2.5。

Balance上有2块网卡,其中一块配置成集群向外部提供服务的公网IP,这个IP也是真实服务器上要添加的集群虚拟服务器IP(VIRTUAL_SERVER_IP);另一块网卡(与真实服务器通信)配置成与各真实服务器的真实网卡为同一网段的IP地址。

5.2 实验方法

采用常用的Web测试工具httpperf,httpperf运行在客户端,每秒发送600个连接请求,设1号影片为热点影片。研究表明,用户对影片访问服从Zipf分布,将影片按照“流行热度”的大小降序排列,则有: $P_i = C/i^{1-\alpha}$, $1 \leq i \leq 20$,其中, $C = (\sum_{i=1}^{20} 1/i^{1-\alpha})^{-1}$,可以计算出用户对热点影片访问概率为0.191,根据这个平均访问概率,拟合一个热度曲线:

$$Q(t) = \begin{cases} 0.1t & 0 < t < 5 \\ 0.5 & 5 < t < 10 \\ -0.084t + 1.34 & 10 < t < 15 \\ 0.08 & 15 < t < 30 \end{cases}$$

(上接第131页)

负载均衡是指相对的均衡,最优的负载均衡是一个NP完全问题,很难实现全局最优,在渲染时也只能视具体的应用折中处理,因此,该算法仍须在实际应用中不断完善。

参考文献

[1] Rajkumar B R. 高性能集群计算:结构与系统[M]. 郑纬民,石威,汪东升,译. 北京:电子工业出版社,2001.

该曲线用于系统性能测试。为了采集每台服务器上的连接数、资源消耗和点播影片的ID号等信息,用Visual C++6.0作为开发工具,基于Microsoft Media Services 9 Series SDK开发工具包开发了一个业务统计程序,该程序运行在每台真实服务器上,每隔1s将收集的信息返回给前端服务器,前端服务器可以把这些信息以文本的形式保存下来,表1中的内容就是根据这些文本文件得到的统计结果,这里,主要给出了在不同H值下热点服务器的资源消耗、吞吐率和热点影片的延迟时间,表1中的H值是客户端在热点影片上的当前连接数,其他值是在不同H值下,由统计结果计算得到的平均值,统计时间为30min。

表1 热点服务器性能统计表

H值	内存占用率/(%)	CPU利用率/(%)	吞吐率/(Mb·s ⁻¹)	延迟时间/ms
30	67.6	61.8	96.2	407.3
50	73.3	69.2	114.8	458.7
80	81.5	79.7	146.3	893.1
110	93.4	90.2	182.6	1139.4

6 结束语

从实验数据可以看到,该系统的性能与H值的设置密切相关,因此,选择恰当的阈值很重要,这可以借鉴以往电影票房收入曲线、各种网站影片点击频率曲线来分析热度曲线,结合具体点播环境的指标参数,给出一个设定的阈值。如何从理论上给出H值的求解方法,从而实现H值的自动配置,这是今后研究的重要内容。

参考文献

[1] Liu Jiangchuan, Li Bo. Optimal Stream Replication for Video Simulcasting[J]. IEEE Transaction on Multimedia, 2006, 8(1): 1-7.
 [2] 杨 灿, 卢正鼎, 邹雪城. 可伸缩分布式VOD系统的设计与实现[J]. 华中科技大学学报, 2005, 33(1): 28-31.
 [3] 章文嵩, 吴婷婷, 金士尧. 一个虚拟Internet服务器的设计与实现[J]. 软件学报, 2000, 11(1): 122-125.
 [4] Nilsen E. Analysis of News on Demand Characteristics and Client Access Patterns[D]. Oslo, Norway: Department of Informatics, University of Oslo, 2005.
 [5] Mundur P, Simon R, Sood A K. End-to-end Analysis of Distributed Video on Demand Systems[J]. IEEE Transactions on Multimedia, 2004, 6(1): 129-141.

[2] Bryhni H, Klovning E, Kure Q. A Comparison of Load Balancing Techniques for Scalable Web Servers[J]. IEEE Network, 2000, 14(4): 58-64.
 [3] Haglin D J, Ford R W. The Message-minimizing Load Redistribution Problem[J]. Journal of Universal Computer Science, 2001, 7(4): 291-306.