基于资源约束的流媒体服务 QoS 策略

蔡贤资[1] 李吉桂[2]

(华南农业大学理学院 广州 510642)[1] (华南师范大学计算机学院 广州 510630)[2]

摘 要 对于流媒体服务,端表现质量是衡量系统性能的关键指标。本文对流媒体服务瓶颈资源进行了分析并提出了基于资源约束的策略来保证端服务质量。通过仿真测试,验证了 QoS 策略的有效性。

关键词 流媒体,QoS,负载均衡

QoS Strategy of Streaming-Media Service Based on Resource Constraint

CAI Xian-Zi[1] LI Ji-Gui[2]

(College of Sciences, South China Agricultural University, Guangzhou 510642)[1]

(Faculty of Computer, South China Normal University, Guangzhou 510630)[2]

Abstract End present-quality is the key performance scale of streaming-media service system. This paper analyzes its key resources and proposes a strategy based on resource-constraint to guarantee the QoS. Simulation test results verify the effectiveness of the strategy.

Keywords Streaming media, QoS, Load balancing

1 引言

网络技术的发展使在线多媒体服务逐渐兴起,特别是面向音视频流媒体的服务。端表现质量是衡量系统性能的关键指标,在网络环境下,影响端表现质量的因素包括网络带宽、服务器性能、中间节点性能和终端机器性能等,在高质要求的流媒体应用服务中对服务质量就更加敏感。要保证系统性能,流媒体服务系统必须采用一定的服务策略来保证 QoS。

2 问题分析

流媒体服务主要是各种音视频服务应用,而各种音视频格式对速率的要求也不一样,常见的音视频格式的码流如图 1 所示。

Application	Speed Requirement
Telephone	16 kbps
Audio—conferencing	32 kbps
CD—quality audio	128—192 kbps
Digital music(Qos)	64—640 kbps
H.261	64kbps—2 Mbps
H.263	<64 kbps
DVI video	1.2—1.5 Mbps
MPEG-1 video	1.2—1.5 Mbps
MPEG-2 video	4—60 Mbps
HDTV(compressed)	>20 Mbps
HDTV(uncompressed)	>1 Gbps
MPEG-4 video-on-demand(Qos)	250—750 kbps
Videoconferencing(Qos)	384kbps—2 Mbps

图 1 流媒体码流

音视频流特别是高质视频流对服务系统要求很高,一般单个视频节目传输时间比较长,从几分钟到一两个小时不等,当并发数增加时,占用系统资源相当高。而在许多应用场合,流媒体服务对媒体表现质量要求甚高,客户端要求能达到数据平滑,对数据抖动(jitter)容忍有限。这就对系统提出了一个问题就是如何在一定的硬件平台下,更高效地提供服务并有效地保证服务质量。

影响系统性能的主要参数包括网络带宽、磁盘 I/O 带宽、服务器性能(主要是内存和 CPU)。我们可以把相应的参数量化,并视作系统资源。任何一个并发连接,都将消耗相应的资源。当并发数达到一个临界阈值时,系统资源耗尽,流服务将处于不稳定状态。一旦超过这个阈值,某些连接流量将不稳定,甚至连接失败,无法保障端表现质量。每个资源并不是一个恒定的数值可以平均分配确定并发数量,它们之间相互影响,某个资源的耗尽会影响其他资源的可用度量甚至导致系统崩溃,因此有必要对相应的资源进行并发控制,避免猝发数据导致网络抖动过大引起某些连接数据饥饿来保障 QoS。以下是针对这几项资源在无 QoS 策略的性能测试,测试平台是:服务器 P4 2.8G, 512M 内存 ,硬盘组 IBM 80G * 4 通过 promise Ultra133 连接成加速阵列,200 个视频流文件为 mpeg2 格式,DVD 质量,平均存放在 4 个硬盘,客户端为赛扬 1.8G,256M 内存,交换机为 DES1024+,网络带宽为 100M,操作系统是 win2K。

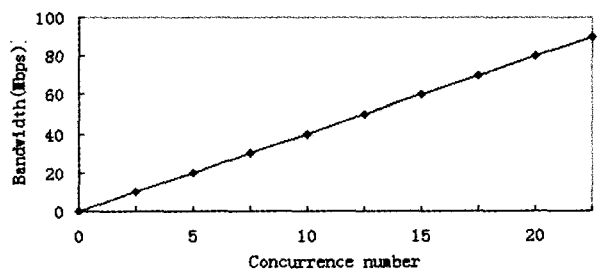

图 2 并发数与带宽使用

蔡贤资 讲师,主研方向:网络多媒体应用;李吉桂 教授,主研方向:计算机网络。

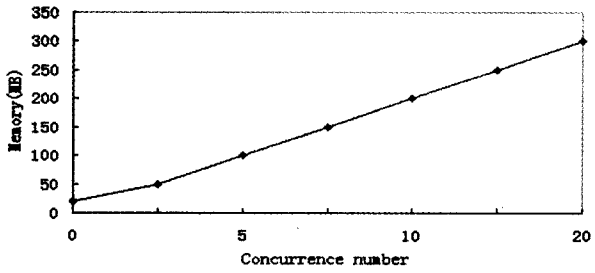

图 3　并发数与内存使用

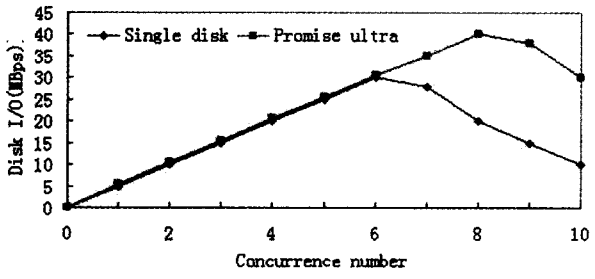

图 4　并发数与磁盘 I/O

由测试得知，在单纯的流媒体服务应用的情况下，带宽和服务器内存在没有消耗尽的前提下基本和并发数呈正比关系。而同一逻辑磁盘的数据 I/O 则会随并发数增长到某一个峰值后急剧下降，不管是同一物理磁盘或阵列磁盘。原因是此时大量的处理时间会消耗在磁盘的寻道定位上。因此对以上几项资源进行准入控制，是保证流媒体服务表现质量的保证。

文[1～3]对带宽和磁盘分别进行策略控制但没有综合考虑系统资源的整合和相互间的优化从而保证 QoS。本文主要讨论基于有限资源的 QoS 策略如何更有效地使用系统资源并保证用户端表现质量，QoS 过程的目标是设定可度量的相关参数控制以达到质量要求和资源约束。本文将针对关键的几个系统资源约束提出解决方案。

3　QoS 策略

很明显，带宽资源、内存资源和磁盘资源是影响服务的几大因素，设为 R_b、R_m 和 R_d。系统为每个连接提供流服务，若每连接需消耗资源为 r_b、r_m、r_d，相应每类资源可支持的连接数 C 存在关于 (R,r) 的函数关系，即满足下式：$C_m=f_m(R_m,r_m)$；$C_d=f_d(R_d,r_d)$；$C_b=f_b(R_b,r_b)$，则可支持的最大连接数为：

$$C_{max}=\min\{C_m,C_d,C_b\} \tag{1}$$

策略目的就是使得 C_{max} 在满足 QoS 前提下尽可能大。由测试得知，R_m 支持的并发数大于前两者，并且 R_m 增加的成本不高，因此主要考虑的资源约束是 R_b 和 R_d。

3.1　基于资源的准入机制

实时流媒体服务向客户端提供平滑实时的媒体流数据，典型应用就是视频点播。相对音频数据而言，视频数据码流较大，对系统影响也较大。在有限带宽的使用上碰到的问题就是网络的抖动、过载或欠载问题。在只考虑流媒体应用的前提下，由于连接的带宽消耗是线性的，我们只需要对有限带宽 R_b 进行增量计算和控制。设流文件 i 的平均码流为 p_i，系统为维护此连接所需要的额外调度控制开销为 e_i，由于属性相近的流连接维护开销相近，可以近似地认为是相同的一个量 e，则当前 m 个并发连接的带宽分配如下：

$$B_u=\sum_{i=1}^{m}p_i+m\times e \tag{2}$$

在总带宽 R_b 一定的情况下，$R_b\geqslant B_u+B_r$，B_r 是为保证传输的预留带宽。

流媒体文件一般都比较大，所以通常需要大容量的磁盘组进行存储。磁盘的数据传输率跟每个磁盘为数据访问提供的服务的响应时间有关。服务响应时间 T_{srv} 主要由定位延时 T_l 和数据传输时间 T_d 组成，其中定位延迟还包括寻道延迟 T_s 和旋转延迟 T_r，即 $T_{srv}=T_s+T_r+T_d$。磁盘 I/O 很明显还跟并发流连接数有关，在多个并发连接都访问同一个物理硬盘或阵列盘，对于每个连接访问的定位延迟会大幅上升。如图 4 所示，我们可以看到并发流连接对磁盘 I/O 的影响。一般而言，磁盘或阵列的服务能力是远大于每个流媒体应用的需求，但当并发数增加到一定数量时，由于产生阻塞，某些连接访问的响应时间将无法满足应用请求，流媒体系统希望磁盘能支持尽量多的并发连接同时必须保证每个连接的 QoS，这是一对矛盾，我们只能找到一个相对最佳的平衡点，即最大并发阈值。计算此阈值通常采用最差或平均服务时间方法，如(3)、(4)式所示：

$$f=\sum_{i=1}^{n}W(T_{srv})_i/mT_i \tag{3}$$

$$f=\sum_{i=1}^{n}A(T_{srv})_i/mT_i \tag{4}$$

其中 $W(T_{srv})_i$ 是最差访问服务时间，$A(T_{srv})_i$ 是平均访问服务时间；T_i 是流连接访问周期；$m\geqslant 1$，为阵列中磁盘数；$n\geqslant 0$，是并发连接数量。阈值就是满足 $f\leqslant 1$ 的最大 n 值。DVD 质量的 Mpeg2 流的平均码流为 4 至 5MBps，要保证 QoS，就要求分配服务时间不小于满足此码流时间。通过计算，我们得出每磁盘(组)在最差服务时间方式下 n 约为 7，在平均服务时间方式下 n 约为 10。其中 $W(T_{srv})_i$ 和 $A(T_{srv})_i$ 可以通过厂商提供的磁盘参数获取。由此，可计算出所有磁盘或阵列的资源值 $R_d=R_{d1}+R_{d2}+\cdots+R_{dm}$，$m$ 为磁盘组数。表 1 是依照测试磁盘参数基于 TV 播放质量的不同码流计算结果。

表 1　不同码流的并发数

	采用最差服务时间	采用平均服务时间
Mpeg1	16～33	25～38
Mpeg2	5～10	8～12
Mpeg4	25～50	40～60

由于内存与并发连接基本是线性关系，因此控制处理只需要判断增量后是否超越系统阈值即可，即满足 $R_m>nM_i+M_r$，其中 n 是并发数，M_r 是为保证系统运作的保留内存。由于内存增长成本较低，在目前的情况下，主要瓶颈是带宽和磁盘 I/O，内存资源 R_m 的容量还是能保证。结合以上约束条件，基于资源的准入许可算法如下：

1. Initialize R_b、R_m R_d；
2. Initialize B_r、B_u、(Du)、M_r；$(Du)_i$ 是第 i 个磁盘或阵列的当前已有访问数；
3. 访问进入，序号为 i，码流为 p_i；
4. if not $R_m\geqslant\sum M_i+M_r$ then goto 10；
5. if not $R_b\geqslant B_u+B_r+p_i$ then goto 10；
6. 定位访问的磁盘或磁盘组序号 m；
7. if not $(R_d)m\geqslant(Du)_m+1$ then goto 10；
8. 发出访问许可，建立连接；
9. $B_u=B_u+p_i$；$(Du)_m=(Du)_m+1$；
10. 拒绝接入，转入队列管理。

由于即使是单纯流媒体应用环境也无法完全避免系统抖动，偶然的抖动会产生连锁反应，无论 CPU 时间，磁盘处理时

间，网络处理时间都大大增加，产生阻塞。阻塞后处理是：当阻塞发生，降低某个磁盘(组)的读写许可数，降低一个并发连接许可数，以求快速恢复。至于取消哪一个连接，处理的方式可以有 FIFO 和 LIFO 两种，其中 FIFO 可以采用服务时间最长或最接近完成两种选择。此外亦可考虑优先权控制。在稳定后可用试探法进行恢复，以达到资源的最大使用率。

设定两组阈值 $T=\{T_b,(T_d)_i\}$ 和 $P=\{P_b,(P_d)_i\}$，T 是资源带宽和磁盘组阻塞恢复时间阈，P 是资源阻塞概率，i 是资源序号。当资源在单位时间 T 内没有发生阻塞，则资源的负荷值增加 1，如在指定时间区间内资源发生阻塞的概率大于 P 则资源不再允许增加负荷值。算法描述如下：

1. 阻塞进入；
2. 按指定方式选择相应连接挂起，相应序号 i 的资源 R_b、R_m 或 $(R_d)i$ 减 1；
3. 登记相应资源号 i，$(T_{block})_i$ 进入阻塞计时，若继续阻塞，转 1；
4. if $(T_{block})_i>T_b$ 或 $(T_{block})_i>(T_d)_i$ 则转 6；
5. $(T_{block})_i$ 继续计时，转 4；
6. if $(P_{block})_i>P_b$ 或 $(P_{block})_i>(P_d)_i$ 则转 8；
7. 恢复阻塞连接，相应资源 R_b、R_m 或 $(R_d)_i$ 加 1，重新计算 $(P_{block})_i$；
8. 设置资源 i 不可增加，取消(Tblock)i 计时。

3.2 基于资源的动态缓存协作的策略

由于多媒体流数据特别是视频流的码流不稳定，在某些极端的情况下变化较大，即使是对服务资源进行了准入控制，也有可能在少数特殊的情况下连接数据饥饿造成客户端回放不平滑，此时即使资源 R_b 尚未到达所设阈值也会影响 QoS。如何在不稳定的网络带宽中保证回放质量，避免在网络带宽抖动的时候影响表现质量是很关键的一个问题。造成此情况主要原因是媒体流数据抖动较大，端用户的回放缓冲数据使用完毕而新数据尚未到达。端操作系统对多媒体数据本身也有缓冲设置，但对较大码流的媒体流特别是高质视频数据流，缺省缓冲是不够的。为了尽量减少这种情况，就需要增大缓冲数据，我们设置了一个协作式的动态缓存机制来处理这个问题。动态流缓冲的作用是尽量利用现有资源延长端用户的缓冲时间，增强系统的抗抖动能力。

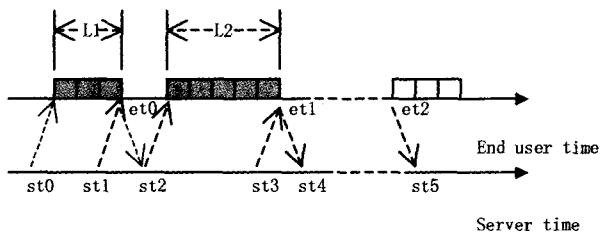

图 5 基于协作的动态缓存

处理描述如图 5 所示，步骤如下：

1. 服务端在 st0 处收到客户端请求建立连接并开始开始发送数据，客户端建立缺省长度 L1 的缓冲，在缓冲填满 L1 长度后在 et0 处开始播放。
2. 服务端在 st2 处接到客户端的 L1 通知，如果此时系统资源 R_b、R_m 和 R_d 满足增长条件，则作增长标记，继续传送数据到客户端直到到达缓冲边界 L2。
3. 客户端在 et1 处达到 L2 边界，停止缓冲填充，并通知服务端，服务端在 st4 处停止发送直到缓冲边界小于 L2。此时，流服务和端回放处于平衡状态。
4. 客户端在 et2 处由于数据饥饿，流回放将产生停顿，服务出现阻塞，服务端在 st5 处进行阻塞处理即需要对阻塞率 P 和阻塞点 T 进行登记，并进行 3.1 节的阻塞后处理，同时暂停满载方填充，恢复所有缓冲边界为 L2 的连接到 L1。
5. 在重新获取用户端边界通知时候，恢复正常，直到当前流服务结束。

由于在产生阻塞的时候客户端通知有可能产生掉包现象，因此还需要一个超时检测机制。设当前平均码流为 p_i，上次缓冲边界确认到当前时间为 T 已发送流量为 S，当前缓冲边界量为 L，缓冲单位量为 m，则当 $(L*m+S)/(T*p_i)$ 小于 1 时就可基本认为已经发生阻塞，因为在此时间内还应包括网络的时延。

对于缓冲的设置是否越大越好呢？这并不一定。一来客户端缓冲受系统限制不是可以无限增大，二来流媒体服务讲究时效性，客户在请求服务后对服务提供时间有个忍耐度，一般缓冲时间我们需要设定在端用户可接受的时间范围内。按我们的经验，高质流媒体服务应用中，影片类用户可忍耐时间可以十几秒以上，歌曲类特别是娱乐 KTV 点歌类只有几秒左右，如果缓冲时间太长，本质上也影响了终端表现质量。

3.3 基于分布支持的资源负载均衡

媒体流服务系统如视频点播系统中，流文件的服务率是有所差异的(如点播系统中的点播率)。按服务频次分布，一般可以认为其概率分布满足参数 $\theta=0.271$ 的 Zipf 分布[3]。其分布函数为：

$$f_i=\frac{c}{i^{1-\theta}},c=[\sum_{i=1}^{n}\frac{1}{i^{1-\theta}}]^{-1},i=1,2,\cdots,n \qquad (5)$$

其中 n 为流媒体节目数，f_i 是频次，i 为频次序号。根据 Zipf 分布特点，用户请求有高度集中并在通常情况下满足 20～80 规则。由上文讨论可知，每个磁盘(组)保证 QoS 前提下可以支持的并发连接有限，因此把热点流文件分布到不同的磁盘(组)就能增加有效连接数。每个流服务器的负荷能力也有限度，通常采用的是多服务器并且每个服务器带若干磁盘(组)的分布服务方式，实现最大限度的负载均衡，使总的并发支持数 D_{total} 尽量趋向于 $\sum(R_d)_i$ 是我们的目标。在进行分布处理时我们还需要考虑另外一个因素就是流节目文件长度。每个流节目长度不一，用户在观看某些长节目如影片的时候，往往会尝试看一部分然后再决定是否继续，也就是说流媒体特别是视频流内部各段也存在类似的访问频率差异。文[4]中对这种用户退出行为做了分析，如图 6 所示。

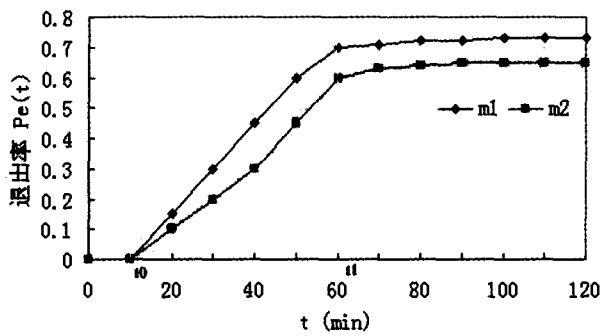

图 6 用户退出率描述

如果采用退出概率 P_e 对这种用户行为进行描述，由图 1 结果可以得出退出率描述函数如(6)式所示：

$$Pe(t)=\begin{cases}0, & 0<t\leqslant t_0\\ E(t-t_0) & t_0<t\leqslant t_1\\ T & t\geqslant t_1\end{cases} \qquad (6)$$

E 值与 T 值会相应不同的节目而有所差异，当 $0<t\leqslant t_0$ 时用户退出率为零，当 $t_0<t\leqslant t_1$ 的时候，用户退出会比较明显，在 $t\geqslant t_1$ 的时候，用户很少退出，基本可以看作是一个常数 T 了。这样如果把节目分割成若干段，其相应的访问率 f_i 就变成 $f_i(1-P_e)$ 了。

综上所述在进行节目分布支持的时候，可以考虑把长节目按一定粒度分割，每片段当作一个相对独立的流节目，赋予

(下转第 112 页)

4.3 服务器端

基于J2EE,我们开发了服务器端的系统。当服务器端的逻辑层servlet接收用户数据后,从数据库中查询出用户的密码,然后用上述的解密算法得到明文。最后,根据标志符进行分类处理,如存储用户的会议密码、人数、旁听密码等等。

5 系统安全性

由于密钥生成器的参数(即公式(1),(2)中的a、x_0、p、y_0)的值在给定的区间内有较大的取值范围,故密钥空间较大;基于KSSL的HTTP协议,保证了数据的完整性。由于采用了反馈加密,使得密文具有良好的扩散性。该方案具有抗穷举攻击能力,这是因为当进行密码分析时,只有获得了上述4个参数才能得到正确的密钥生成器的信息。加之混沌序列的选取是随机的,若用直接的穷举攻击,在理论上是行不通的(因为遍历16个序列值需要验证次数为128^{16},数量级为10^{32})。该方案也具有抗选择明文攻击性:混沌序列中的x_n与x_{n+1},y_n与y_{n+1}之间不是直接的迭代关系,它们是在一个较大的范围内随机地选取的,因而攻击者无法通过简单的运算得出混沌系统的参数[10]。

结论 本文提出了一个将混沌加密与HTTPS协议相结合并应用于移动商务的安全方案。它利用了混沌系统的初值和系统参数的秘密性、序列值选取的随机性,并采用了密钥集中管理、动态更新。算法速度快,加密效果好,能抵抗混沌密码系统的相空间重构攻击、统计分析攻击。该方案适用于客户端/服务器(C/S)模式。

参 考 文 献

1 Park Nam-Je, Song You-Jin. M-Commerce security platform based on WTLS and J2ME. In: Proc. of 2001 IEEE International Symposium on Industrial Electronics, Pusan, Korea, 2001. 1775～1780
2 朱从旭,陈志刚. 一种适于移动计算的快速组合混沌加密方法. 计算机工程, 2005, 31(1): 138～140
3 Lorenz E N. Deterministic nonperiodic flow. J. Atmospheric Sciences, 1963, 20(2): 130～141
4 Eckmann J P, Ruelle D. Ergodic theory of chaos and strange attractors. Rev Modern Phys, 1985, 57: 617～656
5 Jakimoski G, Kocarev L. Chaos and cryptography: block encryption ciphers based on chaostic maps. IEEE Transactions on Circuits and Systems-I, 2001, 48(2): 163～169
6 Lenz H Obradovic D. Global Control of Lorenz Chaos. Proc. of IEEE Conference on Decision and Control, 1977, 2: 1486～1487
7 崔光亮,冯正进,胡国杰,等. 基于参数跳动和扰动的混沌保密系统的理论设计. 上海交通大学学报, 2004, 38(11): 1818～1821
8 Staff M. Introduction of MVC structure in J2ME client (see website: http://www.motorola.com)
9 周红. 有限精度混沌系统的m序列扰动实现. 电子学报, 1997, 25(7): 95～97
10 孙克辉,张泰山. 基于混沌序列的数据加密算法设计与实现. 小型微型计算机系统, 2004, 25(7): 1368～1371

(上接第81页)

相应服务率,然后按服务率分布存储于不同的磁盘(组)和不同的服务器,提高系统的并发处理能力。关于服务率的确定,虽然可以依据(5)式得到,但在不同的应用场合用户可能有不同的偏好,因此对于相应排列我们可以采用基于统计的方式获取。每次重分布的统计我们引入了一个依赖因子λ,并按照依赖因子重新计算所有流节目的服务率及其排列,如(7)式所示:

$$f(i)=\lambda f(i-1)+(1-\lambda)s(i) \quad i=1,2,\cdots,N \qquad (7)$$

其中,i是节目编号,S是本次统计的服务率。依赖因子的引入平衡了每次统计,避免重新分布的过度抖动。

4 测试与分析

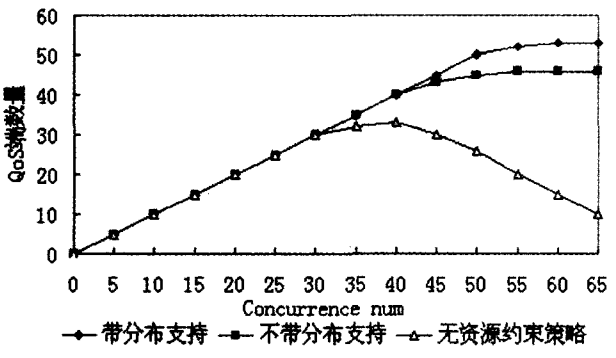

图7 策略控制比较

为了验证上文所述策略的有效性,采用了两台浪潮英信Np350服务器,每台使用promise Ultra133加速阵列分别连接4个IBM80G硬盘,1000个视频流文件为mpeg2格式,DVD质量,平均码流基本相同。其中100个为影视文件,其他为短节目(≤10min)。长节目按(6)式分割,其中t_0=10min,t_1=60min,E取平均值0.4,T=0.6,依赖因子取0.3。按(5)式赋初始服务率值给各个流文件再按概率平均存放在2台服务器8个硬盘,然后按20～80规则,访问相应流节目。实验结果如图7所示。

在无资源约束策略下,系统是尽可能为所有连接提供服务,在连接数增多的情况下,客户端的QoS将不能保证。在有约束策略的支持下,QoS端数量将在达到一定水平后保持稳定,此时多出的并发请求将被拒绝接入并进入队列管理。测试同时也显示使用服务率分布支持系统能负荷的并发数有所提高。实验证明策略对保证客户端表现质量有效,并能较好利用系统资源支持尽量多的并发。

结束语 在大规模的高质流媒体服务系统中,流应用对QoS更为敏感,系统的关键资源也是系统性能的瓶颈,如何更有效地分配和使用这些资源,是保证QoS的关键。本文从资源约束的角度出发,针对系统瓶颈提出一套流媒体服务系统QoS保障的策略,并通过测试验证了策略的有效性。在基于统计的服务率排列计算中,重分布对系统性能也是有一定的影响,这也是下一步的研究方向。

参 考 文 献

1 Li X, Ammar M. Bandwidth Control for Replicated-stream Multicast Video Distribution [C]. In: Proc. of the Fifth IEEE Intl Symposium on High Performance Distributed Computing, Syracuse, NY, 1996, 8: 45～56
2 Wolf J L, Yu P S, Shachnai H. Disk Load Balancing for Video-on-Demand Systems [J]. Multimedia Systems, 1997, 5: 358～370
3 Flynn R, Tetzlaff W. Disk striping and block replication algorithms for video file servers [A]. Multimedia Computing and Systems. In: Proc. 3rd IEEE Int Conf [C]. IEEE Press, 1996. 590～597
4 Wu Song, Jin Hai. Symmetrical pair scheme: a load balancing strategy to solve intra-movie skewness for parallel video servers [A]. In: Parallel and Distributed Proc Symposium, Proc. Int [C], IEEE Press, 2002. 126～132
5 Laoutaris N, Zissimopoulos V, Stavrakakis I. On the optimization of storage capacity allocation for content distribution. [J]. Computer Networks, 2005, 47(2): 409～428
6 李宇辉,李吉桂. 多媒体流间同步参数计算[J]. 计算机科学, 2001, 28(8): 50～52
7 钟玉琢,向哲,沈洪. 流媒体和视频服务器[M]. 清华大学出版社, 2003