

RTSP 流媒体服务器性能测试工具^①

杜彬^{1,2}, 王淑玲², 杨海波^{1,2}

¹(中国科学院研究生院, 北京 100049)

²(中国科学院沈阳计算技术研究所, 沈阳 110171)

摘要: 基于 RTSP 协议的流媒体应用日益增加, 如何评测 RTSP 流媒体服务器的服务性能成为一个亟待解决的实际问题。本文在分析流媒体业务性能指标的基础上, 设计和实现了一种 RTSP 协议流媒体服务器性能测试工具。该工具通过读取测试用例脚本设定参数建立测试场景, 利用多线程机制生成多实例伪终端向流媒体服务器发送 RTSP 信令、接收和解析 RTP 包, 但不解码, 最终测定服务器的相关性能指标值。验证实验表明, 该测试工具实用高效, 能准确反映被测系统的整体性能。

关键词: RTSP 流媒体服务器; 性能测试; 性能指标; 媒体流; 测试工具

Performance Testing Tool for RTSP-Based Streaming Server

DU Bin^{1,2}, WANG Shu-Ling², YANG Hai-Bo^{1,2}

¹(Graduate University, Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110171, China)

Abstract: The attention on the streaming media applications based on RTSP protocol is growing, so the main issue we address in this paper is how to evaluate the serving performance of RTSP streaming media server. With comprehensive analysis of the performance metrics of streaming applications, we propose an approach to design and implement a Performance Testing Tool for RTSP-based Streaming Server. The tool creates the testing environment by obtaining parameters from the script of test case, utilizes multi-thread mechanism to create multiple pseudo-terminal instances to simulate a certain number of concurrent users for sending RTSP signals, receives media flow, analyzes RTP packets without decoding, and calculates the related performance metric values of the server finally. Experiments validate the efficiency and accuracy of the tool.

Keywords: RTSP streaming media server; performance test; performance metrics; media flow; testing tool

1 引言

随着 3G 移动通信网络在国内的全面启动和发展, 各运营商也充分利用 3G 网络为客户提供丰富的个性化业务, 流媒体成为其中重要内容。开放的 RTSP 协议被用于多种流媒体应用中, 作为流媒体系统中必不可少的环节, 各种各样满足用户需要的流媒体服务器应运而生, 如何评测 RTSP 流媒体服务器的服务性能成为一个亟待解决的实际问题。

与传统 Web 服务不同, 流媒体业务传送流媒体文件, 强调实时传输、智能控制、具有多播和实况转播

等特点。显然, Web 服务器性能测试方法不再适用于流媒体服务器, 而国际上各大测试系统提供商对流媒体产品测试的研究还十分有限, 相应的测试工具也非常稀少, 工业级负载测试工具 HP LoadRunnerTM 也仅对 Real Networks 公司的 Real 协议以及 Microsoft 公司的 MMS 协议提供了测试服务, 并未真正涉及 RTSP 协议; 国内至今也没有相应工具的开发。所以, 开发这样一种测试工具对广大 RTSP 流媒体服务器系统的研发者和流媒体服务提供商都具有十分重要的意义。

本文提出了一种基于 RTSP 协议流媒体服务器性

^① 收稿时间:2010-07-02;收到修改稿时间:2010-08-05

能测试工具的设计和实现方法,这种测试工具不但可以测试服务器 RTSP 信令的交互能力,同时更加重点考量流媒体服务器在面对大量用户并发情况下处理媒体流传输的能力。测试工具生成的多个伪终端实例随机访问流媒体服务器上已正确存在的多个 RTSP 地址,服务过程更加真实;另外,为了支持任何 codec 的媒体流传输,工具未进行解码;同时,采用一种专门的传输机制使伪终端只发送和接受 RTSP 信令,而由特定的 IP 地址和端口接收 RTP 媒体流,能更好地支持并发传输,很大程度上缓解了测试工具的压力。

本文首先介绍了相关流媒体协议,并给出性能测试方法和相关性能指标,然后详细阐述测试工具的设计和实现方法,最后利用它对开源流媒体服务器 VLC 进行了测试实验。

2 流媒体相关协议介绍

RTSP(Real Time Streaming Protocol)即实时流协议,是一个用以控制具有实时属性的数据传递的应用层协议,它能够实现音视频数据受控、按需的实时传输,提供播放、停止、快进、快退等控制功能,本身并不传送数据流。RTP(Real Transport Protocol)协议是实时传输协议 RTP 与实时传输控制协议 RTCP 的统称,用于单点和多点之间音视频等多媒体数据的实时传输。RTP 报文中携带了包括数据负载类型(Payload Type)、时间戳标识(TimeStamp)、序列号(Sequence Number)、和同步源标识符(SSRC)等重要字段,运用这些信息进行流同步,同时也为定量分析流媒体服务器性能指标提供了数据来源,它并不保证 QoS 或可靠性传输;RTCP 主要提供数据传输质量的反馈,通过周期性地向服务器发送控制信息包封装发送端和接收端的统计会话信息,根据这些动态统计信息,终端对传输速率等传输参数进行合理的调整和修改,它与 RTP 协议紧密结合使得终端可以向服务器动态反馈传输信息并监测 QoS。

3 性能测试方法及度量指标

3.1 性能测试方案

性能测试主要是测试被测实体在预期用户量或是大用户量情况下系统的稳定性、可靠性和响应时间。性能测试的方法是针对系统的性能指标制定性能测试方案,执行测试用例,运行测试场景,分析并得出测

试结果来验证系统的性能指标是否满足既定值^[6]。对一个 RTSP 流媒体服务器系统进行测试的目的就是为了把握被测系统响应情况以及并发传输媒体流的能力,可以通过逐渐增加并发流数目的方式观察各个性能参数的变化从而得出服务器的最大服务能力。

3.2 性能指标参数

目前,关于流媒体服务器性能评价没有形成固定的标准,本文借鉴 Web 服务器的性能标准并结合流媒体服务器自身的特点给出反映流媒体服务器性能和各种资源利用情况的参数^[4,7]:

- 1) 最大并发流数目:流媒体服务器可以长时间稳定地支持的最大客户端数;
- 2) 聚合输出带宽:服务器向外发送数据流时所能达到的最大发送带宽;
- 3) 请求响应延迟:从用户发出 RTSP 点播信令开始到接收到第一个 RTP 数据包之间的时间间隔;
- 4) 服务器端丢包率:服务器丢弃 RTP 数据包的情况;
- 5) 延迟抖动:两个数据包之间发送与实际到达的时间间隔;
- 6) CPU 占用率:随着并发流数目的增加,流媒体服务器提供服务时所占用 CPU 资源的情况;

3.3 典型的流媒体请求与传输过程

图 1 是一个典型的流媒体请求及媒体流传输流程,由终端播放器向流媒体服务器发出播放请求,服务器端返回成功响应,随后服务器向终端发送 RTP 媒体流,期间终端可以执行暂停、快进、快退、结束会话等控制操作。

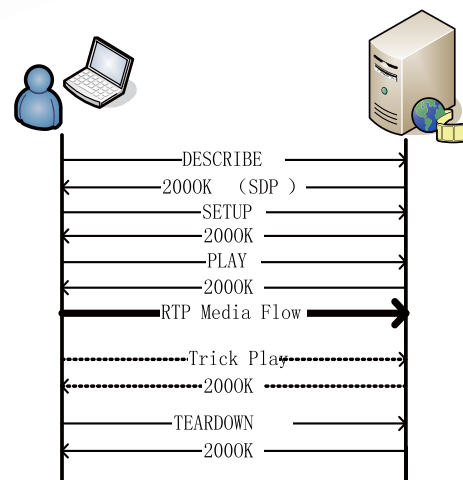


图 1 典型的媒体点播及媒体流传输流程

4 测试工具的设计与实现

4.1 测试工具的访问模式

从本质上来说，测试工具首先是一个请求终端的行为，但它能同时建立多路请求，另外需要提供被测系统的性能指标值。为了更加真实地反映流媒体服务器提供服务的过程，本文采用多线程机制同时生成多个实例的伪终端随机地向流媒体服务器中已正确存在的 RTSP 访问地址发起请求，而限制于多个实例只对一个流媒体文件进行访问的模式^[6]。但很显然，这也引入了实际问题。我们知道服务器的性能状况与媒体文件的码率有直接的关系——码率越大，最大支持的并发流数目越少，丢包、抖动等问题随之加重。所以如果选择的媒体文件码率相差较大，而伪终端实例又只是随机地对其进行访问，那么测试结果就会存在很大的不确定性。为了避免这种情况，本文设计的测试工具需要在测试前统一媒体文件的码率与文件播放时间。

另外，测试工具与真正的流媒体终端最大的区别还在于它未进行解码工作。在一个完整的流媒体系统中，编解码器是重要的组成部分，强大稳定的解码器可以为终端用户提供更好的服务。但因为这里关注的是流媒体服务器在面对大量并发用户请求下传输媒体流的能力，而不是强调音视频在终端的最后呈现，所以测试工具无需完成解码，那么理论上测试工具可以支持任何 codec 的媒体流。

4.2 测试工具工作原理

流媒体服务器性能良好与否不仅体现在对信令的响应能力上，而更重要的是它对大量并发媒体流的处理。但如果流媒体服务器向所有发出请求的伪终端发送真实的媒体流数据，那么伪终端需要具有很强的处理媒体流的能力，在大量的媒体流并发的情况下，仅它自身就会占用大量的系统资源，导致测试过程难以维系，出现分析不准确甚至错误。所以，本文采用一种新的媒体流传输机制，使伪终端实例只发送和接收与流媒体服务器之间的 RTSP 交互信令，而真实的媒体流则通过一定的映射方法被发送到专门的地址并得到相应的处理^[1]。测试工具工作原理如图 2 所示。

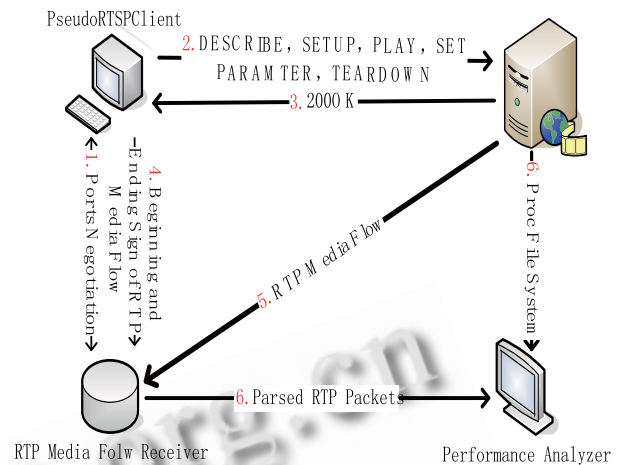


图 2 工作原理

4.3 测试工具的体系结构和功能模块

测试工具从逻辑上包括三个应用程序，一个是只负责发送和接受 RTSP 信令的伪终端，一个是接收从服务器端发送过来的 RTP 数据流的媒体流接收器，它在接收媒体流的同时对 RTP 包进行解析获取一定的头域信息，另一个是分析计算出上文中列出的各性能参数值的统计分析器。

测试工具体系结构如图 3 所示^[5]：

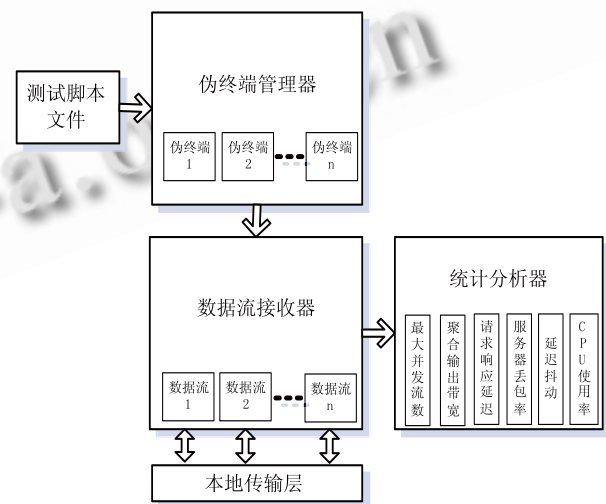


图 3 测试工具体系结构图

1) 测试脚本文件：测试场景的建立以脚本文件的输入参数为基础，按照下面几个方面建立测试场景：

① 启动方式选择并发用户同时发起请求还是以线性增长的方式逐渐达到设定并发用户数；

② 终端实例执行的应用类型采用单播还是多播;

③ RTP 数据流基于 UDP 还是 TCP;

④ 终端实例发送 TEARDOWN 消息是在请求媒体播放完毕后向服务器还是限定媒体文件播放一段时间后;

通过这些设置模拟真实的流媒体请求和数据流传输过程。

2) 伪终端管理器: 伪终端管理器提供了整个测试流程的管理。测试工具从本质上来说首先是一个多实例的 RTSP 终端, 每一个终端实例就是一个线程, 各自执行自己的 RTSP 请求, 维持着自身的状态。伪终端管理器负责终端实例的生成和撤销, 它维护一个终端实例队列, 由活跃的终端实例组成。伪终端管理器把生成的伪终端实例插入队列, 同时检查每一个终端实例的状态, 把完成会话或意外终止的终端实例撤出队列。

3) 伪终端: 伪终端通过 RTSP 协议与流媒体服务器进行通信, 通过分析测试脚本中设定的并发流数, 伪终端建立相应数目的实例, 它同时支持基于 UDP 和 TCP 的 RTP 流, 也支持单播和组播应用。为了确保服务器向正确的地址发送 RTP 媒体流, 伪终端与数据流接收器需要事先协商好正确接收媒体流的端口信息; 同时, 为了保证 RTP 数据流在伪终端与服务器正常会话期间到达数据流接收器端, 而不是因为出现错误被开始或终止, 终端实例要在开始接收和停止接收 RTP 流前给数据接收器发送一个相应的标识。

4) 数据流接收器: 与伪终端相呼应, 数据流接收器也采用多线程机制, 每一个线程负责处理从相邻两个监听端口上接收的一路 RTP 媒体流。将监听端口设定为一系列连续的端口, 由于每一路数据流占据 2 个端口 (RTP/RTCP), 则监听端口从设定的起始值 InitialPort 依次增至 InitialPort+2*N-1 或 InitialPort+4*N-1 (N 为测试脚本中给定的并发流数, 视频文件需要分别建立音视频两个媒体通道, 一路流共需 4 个端口); 同时数据流接收器要解析 RTP 数据包, 获得 TimeStamp、SeqNumber 等信息为后面定量分析响应延迟、服务器端丢包率、延迟抖动等性能指标提供直接依据。

5) 统计分析器: 作为一个评测工具, 在实现多路请求媒体流的模拟基础上, 必须提供一定的分析数据

来反映被测系统的性能, 这个工作主要由统计分析器完成。每一路媒体流都会包含自身的状态信息, 包括带宽、延迟、抖动等, 统计分析器统计各路媒体流的性能指标值得出各性能参数的平均值, 为服务器性能分析提供量化依据。

4.4 测试工具的实现要点

伪终端运行后主界面如图 4:

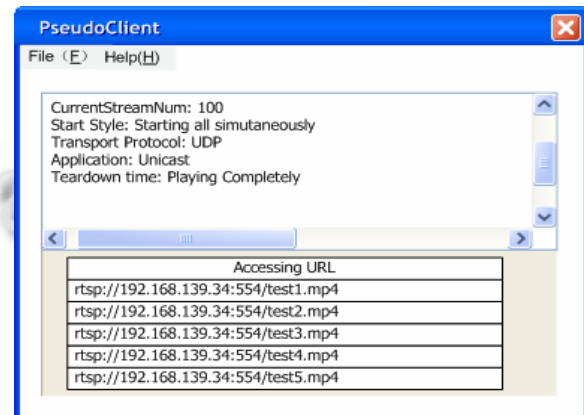


图 4 伪终端主界面

1) 测试工具是一个密集的多线程应用, 多线程的伪终端实例就意味着例如共享内存等更多开销。伪终端的线程结构基于 pthreads 线程模型, 使用头文件 <pthread.h>, 创建线程调用 pthread_create(), 终止线程调用 pthread_exit()/pthread_join(), 调用 pthread_cancel()实现一个线程对另一个线程的 cancel, 对互斥信号量的访问调用 pthread_mutex_init/pthread_mutex_lock/pthread_mutex_unlock/pthread_mutex_destroy 等。

伪终端线程 pseudoClient_thread 的数据结构:

```
struct pseudoClient_thread
{
    //Global properties
    bool is_eof;
    bool is_parsing;
    bool is_can_pause;
    bool is_can_rate_control;
    bool is_can_pace_control;
    bool is_abort;
    //Playtime configuration and current time
    int i_start; //start-time
```



```

int i_stop; //stop_time
int i_run; //have run time;
int i_time; //current time
//Pending actions
pthread_mutex_t lock_ctrl;
pthread_cont_t wait_ctrl;
}
    2) RTP 媒体流的数据结构:
struct RTPReceivedInstance
{
    unsigned int32 fSSRC;
    unsigned fTotalNumPacketsRecv;
    int fTotalBytesRecv_high, fTotalBytesRecv_low;
    unsigned fBaseSeqNumRecv;
    unsigned fMaxSeqNumRecv;
    unsigned fPreviousPacketRTPTimestamp;
    double fJitter;
    struct timeval fLastPacketReceptionTime;
    unsigned fBandwidth;
};
    
```

为了减少程序运行的平台硬件本身比如网卡、内存、硬盘等对测试工具造成的影响，接收器每次解析过 RTP 包取出有效信息后就将其“丢弃”，即释放内存，缓解接收端压力。通过在接收器内部设置若干个定时器来记录请求响应的时间间隔以及整个呼叫的时长；记收到的第一个 RTP 包的序列号为 fBaseSeqNumRecv，最大的序列号为 fMaxSeqNumRecv，共收到的 RTP 数据包的个数为 fTotalNumPacketsRecv，则服务器端丢包率计算如下：

$$\text{Packet loss rate} = 1 - \frac{fTotalNumPacketsRecv}{(fMaxSeqNumRecv - fBaseSeqNumRecv)}; \quad (1)$$

若服务器发出的 RTP 包的时间戳记录 Si 即数据包的发送时间，Ri 为实际到达终端的时间，则延迟抖动指标^[7]：

$$\text{Jitter}(i,j) = (R_j - R_i) - (S_j - S_i) \quad (2)$$

另外，“proc”是 Linux 操作系统的伪文件系统，它自动收集系统的 CPU 及内存等资源的使用信息，统计分析器在不同的并发流请求下通过读取服务器端 /proc/cpuinfo, /proc/meminfo 文件获得它们的使用情况。

5 实验测试

为了验证测试工具的有效性，本文对开源 VLC1.0.5 版本的流媒体服务器的业务性能进行了测试，由于测试平台受限，测试只在局域网环境下实施。实验硬件配置如表 1 所示：

表 1 服务器端配置

CPU	Pentium(R) Dual-Core E2210 @ 2.20HZ
内存	4GB
硬盘	16T 磁盘阵列
OS	Ubuntu 8.04, EXT3 文件系统
网卡	TP-LINK TG-3269C 10/100/1000M

测试用例：整个测试过程服务器提供服务的所有数据都从硬盘中获取^[2,3]，采用固定码率方式传输媒体文件，同一时间启动设定数目的伪终端实例，RTP 媒体流基于 UDP 传输，应用单播，且等待媒体文件播放完毕再终止传输媒体流，选用 10 个其视频码率为 1Mb/s，音频码率为 128kbit/s 的 MP4 格式文件。表 2 显示了随着设定并发流数目的增多 CPU 占有率、聚合输出带宽、丢包率的变化。

表 2 各性能指标值随着并发请求数目的增多而变化

并发流	聚合输出 带宽(Mb/s)	丢包率(%)	CPU 利用率 (%)
50	60	0	22
60	72	0	25
80	95	0	30
100	121	0	33
120	142	0.8	44
140	165	1	56
160	194	1.3	68
180	218	2	78
200	238	2.5	86
220	264	2.8	92
240	282	7	95
250	281	10	96
260	280	12	96

可以看到，CPU 利用率并非随着并发流数目的增多而呈线性增长，在请求的并发流数目较少时，系统资源充足，服务器工作流畅，CPU 利用率增加缓慢，也几乎没有丢包现象，聚合输出带宽基本呈线性增长，每一路分到的带宽平均为 1.17Mb/s；而当并发流数

目大于 100 时, 由于系统调度策略的影响等, CPU 利用率开始显著增长, 并开始有丢包, 此时聚合输出带宽仍然平滑增长; 最后, 当并发流数目升至 240 时, CPU 利用率达到 95%, 丢包率到达流媒体服务最大允许的丢包阈值 7%, 聚合输出带宽也达到峰值 280Mb; 随后再增多并发流数目, 服务器已经不能正常提供服务。

6 结论

随着支持 RTSP 协议的流媒体服务器和终端产品的相继推出, 如何全面评测这些产品的性能成为一个迫切需要解决的问题。本文提出了一种基于 RTSP 流媒体服务器测试工具的设计和实现方法, 测试工具能够根据具体需要设定测试用例, 并统计得出该测试过程中服务器各性能指标值, 其本身不能完全代替分析人员, 但在很大程度上为测试结果的确定提供了最直接客观的量化依据。需要说明的是, 该工具只针对服务器单个的基本性能参数进行的量化分析, 而不同的流媒体服务提供商因具体业务的不同而对服务器各个性能指标的要求程度也不同, 下一步的工作可以建立一个分析服务器整体性能的数学模型, 通过模型更加全面的量化各种流媒体服务器的综合性能。

(上接第 105 页)

- 4 林新棋. 基于模糊理论的电影情感内容识别. 北京: 北京邮电大学信息与通信工程学院, 2009.
- 5 Dai XW, Cai ZP, Zhong GY. Video Shot Detection Method Based on Wavelet Transform and Wavelet Entropy. *Journal of Southwest Jiaotong University*, 2008,43(3):314-318.
- 6 Wang XJ, Ding HT, Chen HX. A Shot Clustering Based Approach for Scene Segmentation. *Journal of Image and Graphics*, 2007,12(12):2127-2131.
- 7 Rasheed Z, Sheikh Y, Shah M. On the use of computable features for film classification. *IEEE Trans. Circuits Systems for Video Technology*, 2005,15(1):52-64.
- 8 Wu MS(translated). *Probability and Information*. Shanghai Science and Technology Press, 1964. 29-88.
- 9 Xu M, Chia LT, Jin J. Affective content analysis in comedy and horror videos by audio emotional event detection. *Proc. of IEEE International Conference on Multimedia and Expo, Amsterdam, Netherlands*, 2005,7:621-625.
- 10 Su JH, Huang YT, Yeh HH, et al. Effective content-based video retrieval using pattern-indexing and matching techniques. *Expert Systems with Applications*, 2010,37(11):5079-5085.
- 11 Lin XQ, Wen XM, Lu ZM, et al. Video Affective Content Recognition Based on Film Grammars and Fuzzy Evaluation. *Proc. of the 2008 International Conference on MultiMedia and Information Technology*. Three Gorges, China, 2008:264-267.

参考文献

- 1 Lee YJ, Min OG, Kim HY. Performance Evaluation Technique of the RTSP based Streaming Server. *Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05)*, 2005,14-16:414-417.
- 2 Cherkasova L, Staley L. Building a Performance Model of Streaming Media Applications in Utility Data Center Environment. *Third IEEE International Symposium on Cluster Computing and the Grid (CCGrid'03)*, doi:10.1109/CCGRID.2003.1199352.
- 3 茅旭峰,李俊,吴刚.流媒体服务器服务能力基准实验与性能模型. *小型微型计算机系统*,2007,2(2):204-209.
- 4 李学森,李俊,奚宏生.流媒体服务器性能评价模型. *计算机工程*,2009,35(10):270-272.
- 5 张静,雷为民.一种支持媒体流的 SIP 性能测试工具的设计与实现. *小型微型计算机系统*,2006,27(10):1831-1836.
- 6 林浒,王阳,雷为民,于东海. SIP 协议测试方法和测试工具的研究. *小型微型计算机系统*,2008,29(3):406-410.
- 7 曾彬,张大方,张玫.基于主动测试的流媒体业务性能测量工具设计与实现. *计算机应用*,2008,28(4):832-836.
- 8 Schulzrinne H, Lanphier R, Rao A. Real time streaming protocol (RTSP). RFC 2326, Internet Engineering Task Force, Apr.1998.
- 9 Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: a transport protocol for real-time applications. RFC 3550, Internet Engineering Task Force, July 2003.