

# 基于 Flex、Red5 和 MongoDB 的视频直播、录制及存储系统设计

郑静静, 叶焱\*, 刘太君, 代成, 王鸿来

(宁波大学信息科学与工程学院, 浙江宁波 315211)

(\*通信作者电子邮箱 yeyan@nbu.edu.cn)

**摘要:**为改善以往网络视频直播或点播过程中播放不流畅的情况和寻找大量视频数据的存储策略,提出了一种实时的视频直播、录制及存储系统的整体设计方案。利用开源的流媒体服务器 Red5 结合富互联网技术 Flex 实现直播和录制;通过开源 NoSQL 数据库 MongoDB 将录制的视频数据进行存储。实际测试结果表明,该平台能满足系统在多用户访问和数据存储方面的要求。

**关键词:** Flex; 流媒体; Red5; 视频直播; MongoDB

**中图分类号:** TP311.1; TP399 **文献标志码:** A

## Design of live video streaming, recording and storage system based on Flex, Red5 and MongoDB

ZHENG Jingjing, YE Yan\*, LIU Taijun, DAI Cheng, WANG Honglai

(College of Information Science and Engineering, Ningbo University, Ningbo Zhejiang 315211, China)

**Abstract:** In order to improve the conventional situation that network video does not play smoothly during live or on-demand and find storage strategy of mass video data, this paper presented an overall design scheme of a real-time live video recording and storage system. The open source streaming media server Red5 and the rich Internet application technology Flex were utilized to achieve live video streaming and recording. The recorded video data would be stored in the open source NoSQL database MongoDB. The experimental results illustrate that the platform can meet requirements of multi-user access and data storage.

**Key words:** Flex; streaming media; Red5; live video streaming; MongoDB

## 0 引言

视频直播是网上直播的一种形式,最初是将文本信息通过互联网实时播放,而现在在各种多媒体技术的支持下,可实现将活动现场的实时音频和视频信号经过压缩,送到特定的多媒体服务器以供授权用户就近收听或观看。这样,不仅可以把各式各样的社会活动借助 Internet 这个工具向世界发布,而且也可以利用这种技术构建各种各样的系统平台,如学术交流平台<sup>[1]</sup>、网络教育系统平台<sup>[2]</sup>和视频会议直播系统<sup>[3]</sup>等。这些系统平台构建的过程中采用的方案不外乎两种模式——客户端/服务器端(C/S)和浏览器/服务器(B/S)。C/S架构的缺点是部署、更新困难;B/S架构的缺点是受限于HTML页面的形式,无法像C/S那样使用丰富的效果来展示数据,用户体验比较糟糕。富互联网应用(Rich Internet Application, RIA)的出现给我们带来了在客户端和服务器的平衡机会<sup>[4]</sup>。RIA属于新一代的网络应用程序,同时具备多种性能,如Web程序的方便、高效,桌面应用程序的最佳页面功能和多媒体技术网络信息的实时快捷。另一方面,以往当用户观看媒体内容时,必须先将整个文件下载到本地然后

才能观看,这样既耗时又占用存储空间,而流媒体的出现缓解了上述问题,客户端可以采用边传边播的形式<sup>[5]</sup>。与文献[6]不一样的是,对于录制过程产生的视频数据使用MongoDB数据库进行存储,而相对于MySQL数据库,MongoDB又存在一些新的特性。

本文旨在通过在前端表现层使用支持RIA的Flex框架,后端采用开源的流媒体服务器Red5对实时视频流进行处理,实现实时视频的浏览和录制,构成一个用户体验良好、部署成本低和更新简单的视频直播系统,并且采用开源的MongoDB数据库对系统运行中产生的视频数据进行存储,实现一套完整的、经济适用的视频直播系统。

## 1 Flex 前端表示层

由于服务器端采用的是Red5流媒体服务器,Red5采用了类似于FMS(Flash Media Server)的接口,所以前端使用Flex来开发以获得与服务器的配合。Flex是Adobe公司开发的支持RIA开发和部署的技术产品,借助于Flex强大的功能可以开发出更富有交互性的用户界面。利用Flex构建的RIA能够整合实现服务器端复杂业务逻辑的Java平台技术。Flex

**收稿日期:** 2013-07-24; **修回日期:** 2013-09-22。 **基金项目:** 国家自然科学基金资助项目(61171040); 浙江省自然科学基金资助项目(Y1101270); 浙江省大学生科技成果推广项目(2012R405055)。

**作者简介:** 郑静静(1988-),女,湖北钟祥人,硕士研究生,主要研究方向:智能仪器远程控制、实时监控; 叶焱(1966-),女,云南昆明人,副研究员,主要研究方向:智能仪器、高效射频功放设计; 刘太君(1965-),男,四川绵阳人,教授,博士生导师,博士,主要研究方向:物联网、云计算、功放线性化; 代成(1987-),男,湖北随州人,硕士研究生,主要研究方向:云计算、分布式海量数据存储; 王鸿来(1990-),男,浙江杭州人,主要研究方向:流媒体数据存储。

应用程序使用 MXML 和 ActionScript 语言编写,可运行在 Adobe Flash Player 中或者桌面的 Flash 运行环境中,可以把 Flex 应用看作动画一样,其开发主要解决三个方面的问题:表现层界面展示和人机交互、客户端数据操作以及与服务器进行交互整合<sup>[7]</sup>。前端表现界面部分针对音视频流操作等功能进行基本的框架设计。

### 1.1 实时视频录制功能

本文中录制功能由 Flex 与 Red5 共同完成,并且由于录制产生视频文件,需要增加与数据库交互的逻辑。在录制功能中,与数据库的交互发生在录制结束之后。当录制结束时,客户端状态从录制状态转移到已连接状态,并且通过远端调用流媒体服务器中的函数,使得流媒体服务器将视频文件保存到 MongoDB 数据库中。

### 1.2 播放功能

由于有一些文件没有存储在流媒体服务器的缓存里,所以不能直接播放文件。若需要播放,客户端首先发送一个视频请求,服务器接受请求,将数据库中的视频数据写到缓存,然后发送参数到客户端,此时客户端就可以通过预设的函数播放所需要的视频<sup>[8]</sup>。

## 2 Red5 流媒体数据处理层

随着 Flash 播放器(Flash Player)在各个平台浏览器上得到支持,Flash 技术得到了迅速的普及,大量的流媒体视频系统采用了 Flash 平台作为客户端技术。本文采用目前兼容 Flash 播放器的流媒体服务器 Red5,将传统的流媒体功能与灵活的开发环境结合起来,为最广泛的群体创建和提高创新的、交互式媒体应用<sup>[9]</sup>。流媒体(Streaming Media)是应用流技术在网络上传输多媒体文件,如视频、音频、动画等,而流技术就是把连续的影像和声音信息经过压缩处理后放到服务器,让用户一边下载一边观看、收听,而不需要等到整个压缩文件下载到本机之后才能观看到的网络传输技术。流媒体服务器是流媒体技术应用的核系统,是运营商向用户提供视频服务的关键平台,其主要功能是对媒体内容进行采集、缓存、调度和传输播放<sup>[10]</sup>。目前流行的流媒体服务器有 Adobe 公司的 Flash Media Server, Microsoft 公司的 Windows Media Server, Real 公司的 Real Media, Apple 公司的 QuickTime,此外还有一些类似的开源服务器。FMS 是商业化的多媒体支持软件,代码和通信协议保密且价格昂贵。Red5 是一个新兴的开源项目,用于实现 Flash 与服务器之间的流媒体服务,使用开源技术 Spring Apache Mina, Jetty, 实现了实时消息传送协议(Real-Time Messaging Protocol, RTMP), RTMPT(用 http 包装后的 RTMP), AMF( Action Message Format), HTTP 通信协议。RTMP 协议首先是由 Macromedia 公司设计开发的一种高效的实时流媒体传输协议,能够迅速地传播流媒体数据,并且可以根据客户端带宽情况进行适配性的操作。RTMP 协议就像一个用来盛放数据包的容器,这些数据可以是 AMF 格式的数据,也可以是 FLV 中的音/视频数据<sup>[11]</sup>。Red5 与 Flex 前端界面以及 MongoDB 数据库的交互如图 1 所示。

## 3 MongoDB 数据存储层

MongoDB 是可扩展、高性能、开源、模式自由和面向文档

的数据库,旨在为 Web 应用提供可扩展的高性能数据存储的解决方案<sup>[12]</sup>。在海量数据存储中,主要运用复制和分片功能,实现数据库的横向伸缩扩展,以此存储动态增长的海量视频数据。

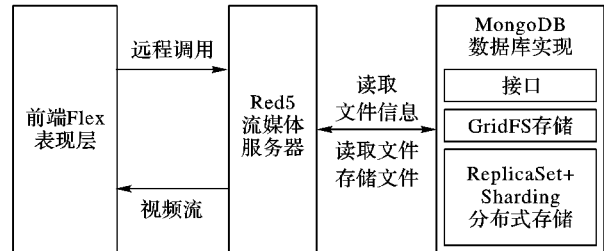


图1 服务器交互技术框图

MongoDB 的复制功能使用在工业生产场合,特别是如今存储引擎已经不提供单服务器持久性功能的情况,复制功能可以完全用于故障转移或者数据一致性,也能用在更多的场合,比如分散读、热备份或者作为离线批处理的数据源。

分片(Sharding)是 MongoDB 实现规模扩展的方法,在不影响应用的情况下运行,增加更多的机器来处理日益增长的负载和数据量。分片是分割数据,将数据片分别存储到不同机器上的过程。在增加更高性能机器的前提下,机器之间分割数据有利于存储更多的数据,处理更多负载。MongoDB 支持自动分片,可以减轻手动分片的麻烦,由集群自动处理数据分片以及负载均衡<sup>[13]</sup>。

MongoDB 中的数据都存储在文件中,对于管理文件有着天然的优势。并且 MongoDB 附带了一个文件系统 GridFS,这个文件系统能够与 MongoDB 整合,透明地将大文件分割成大小相同的一个个记录,存储在 MongoDB 数据库中。这种特性更适合存储视频、音频等较大的二进制文件,而且不会产生磁盘碎片。GridFS 的实质就是数据库中的集合,所以能够通过数据库的分布式设置,将文件分布存储到数据库集群中,从而构成一个分布式的文件系统。本实验使用了两台服务器,四个 MongoDB 服务器进程,创建了两个复制集用作分片,实现了一个最小的分布式系统,构架如图 2 所示。

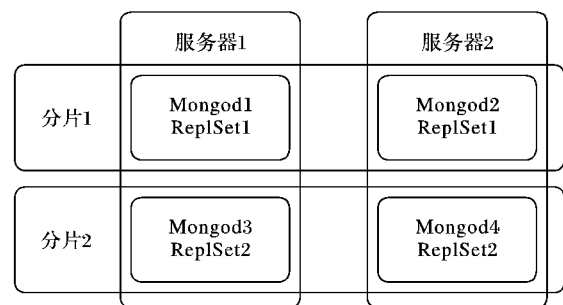


图2 最小的分布式架构

图 2 中有服务器 1 和服务器 2 两台服务器,分别运行了两个 MongoDB 数据库进程(Mongod1、Mongod3 和 Mongod2、Mongod4),为了实现数据的冗余备份,创建了两个复制集(ReplSet1 和 ReplSet2),并且将两个复制集的两个进程分别放在不同的服务器上面,在这种情况下,如果其中一台服务器出现故障,由于复制集分布在两台服务器上,另一台服务器就有复制集中的所有数据,能够不间断地提供服务并且将另一

台服务器从错误中恢复。实现两个复制集之后,在这两个复制集上创建两个分片(分片1和分片2),能够将数据均衡地分布在这两个复制集中,实现了系统的分布式特性<sup>[14]</sup>。按照系统中不同文件的存取要求,文件被分布到不同的服务器中,这样既分布了存储,也分布了计算。

## 4 系统框架的实现

### 4.1 Red5 与 Flex 客户端通信的实现

Red5 与客户端的通信实现了 Flex 应用所要完成视频的录制播放功能<sup>[15]</sup>。在平台运行时, Flex 客户端与 Red 之间有以下几种状态:未连接状态、已连接状态、播放状态和录制状态。具体实现功能的过程如下:

1) 连接 Red5 服务器获取视频列表。Red5 启动时会自动从 MongoDB 获取并保存文件列表,当 Flex 连接 Red5 后,会调用 Red5 端的 getFLVs() 函数,从而获取文件列表显示在界面上,此时, Flex 与 Red5 之间的状态从未连接转移到已连接并一直保持连接状态。

2) Flex 端播放视频。点击列表中的文件名后,如果当前状态为播放或者录制,那么 Red5 立即停止视频流的传输,然后 Flex 调用 Red5 的 ready(String arg) 函数,检查该文件是否存在于服务器的缓存中。若文件存在,就直接播放;否则,就先将视频数据从 MongoDB 数据库中写到服务器的缓存中,并给出相应的写数据提示,当用户再次点击时就能够观看到视频了。

3) Flex 端录制视频。与播放类似,点击录制按钮后,录制的视频开始向 Red5 传输。点击录制停止按钮后,在 Red5 的 streams 目录下就会生成一个指定文件名的 flv 视频文件,该文件即为刚录制的文件。同时,录制结束后, Flex 会调用 Red5 的 addFLV(String arg) 函数,将视频存储到 MongoDB 数据库,并且更新视频文件列表。

### 4.2 Red5 与 MongoDB 数据库通信的实现

Red5 流媒体服务器与 MongoDB 数据库的通信需要实现 3 个功能:读取文件列表、读取文件和存储文件。因为 Red5 流媒体服务器是由 Java 语言开发,在对 MongoDB 操作时需要导入 Java 对 MongoDB 操作的一系列类库,这个类库是由 MongoDB 开发的<sup>[16]</sup>。使用这个类库,实现以下 3 个模块:1) 初始化并且读取文件信息;2) 读取文件;3) 存储文件并且更新文件信息。其中初始化和读取文件的函数在 Red5 刚开始运行时被执行,初始化与 MongoDB 的连接,并且把文件列表缓存到 Red5 上,这样做的好处是能够减少流媒体服务器对数据库的访问。就单个流媒体服务器而言,这样做是没有问题的,当流媒体服务器有多个时,每个流媒体服务器就必须定时访问数据库以检查其他流媒体服务器对数据库的更新。以上 3 个模块对应上述 Flex 与 Red5 通信完成功能的三个函数——getFLVs()、ready(String arg) 和 addFLV(String arg)。由于服务器在初始化时已经生成了文件列表,因此获取列表只是间接地访问 MongoDB 数据库;准备播放函数则将要播放的视频文件从 MongoDB 数据库中缓存到服务器,在大规模的应用当中,缓存的视频可能会越来越多,为了节约空间需要删

掉一些缓存的视频数据;添加视频函数将通过录制的视频存储到 MongoDB 数据库中。后两个函数如同文件的映射,将 MongoDB 数据库与 Red5 流媒体服务器结合起来,实现了分布式的存储特性。

## 5 项目测试及分析

客户端应用程序在 Flash Builder 4.6 中开发。将文件的输出路径指向 Red5 服务器安装的 webapps 目录下,经过编译生成的 swf 文件嵌入在 html 网页文件中,通过浏览网页就可实现相关操作。服务端利用 Red5 提供的通用模板进行系统开发<sup>[17]</sup>,模板位于[Red5\_Home]/doc/templates 目录。

### 5.1 Red5 流媒体服务器并发连接数目测试

并发连接数是流媒体服务器类似于大多数其他用途的服务器的性能指标,即一个服务器同时能够承载连接的客户端的数量,根据测试的数据可以计算出具体某个应用所需的流媒体服务器的数量<sup>[18]</sup>。本文通过测试 Red5 能够承载的最大同时播放视频流的客户端数目来测试它的并发连接数,这项测试能够大致测出流媒体服务器的并发性和吞吐量。以下是本测试所使用的硬件环境:

中央处理器: Intel Core i3 M380 2.53 GHz 双核四线程;  
内存: 2 GB DDR3 1333 MHz;  
硬盘: 5400 转/分钟;  
操作系统: 32 位 Windows 7 操作系统。

通过将上文已经完成的客户端 Flex 程序修改为播放多个视频的模式,模拟多个用户对服务器的访问。启动 Red5 流媒体服务器时,所占用的内存为 48 MB;使用测试客户端打开 10 个连接,服务端内存占用量为 55.7 MB,用来测试客户端的 Adobe Flash Player 进程显示为 167.3 MB;然后,分别对 20、30、40 和 60 个连接的时候服务器端和客户端所占用的内存与 CPU 进行了测试,得出连接数与服务器端的内存资源占用量的关系如图 3 所示。

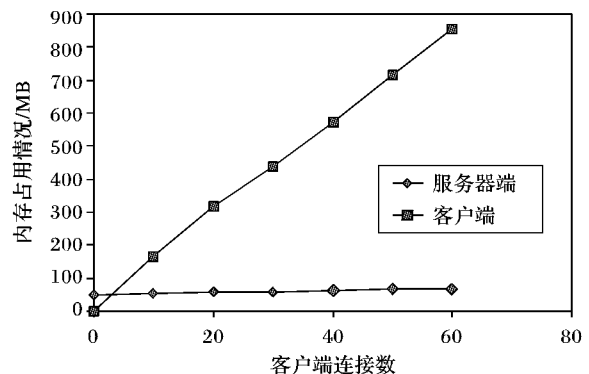


图3 内存资源占用与客户端连接数的关系

从图3可看出:客户端的资源占用量飞速上升,而服务器端所占用的资源量却没有大幅度的上升。当连接数量在 60 个左右时, Red5 流媒体服务器占用内存约 70 MB, 占用 CPU 7% 左右。由此推断,一台服务器能够承载的并发连接数超过 500 个,对于一般的应用,这种并发数量足以满足需求。

### 5.2 测试 MongoDB 的文件读写速度

同样使用上述的测试环境,两台 MongoDB 服务器安装在

上述 PC 的两台虚拟机上,虚拟机的环境如下:

虚拟软件:Oracle VM VirtualBox;  
处理器:单核,主频为 2.53 GHz;  
内存:256 MB DDR3 1333 MHz;  
硬盘:动态分配的 vdi 文件;  
操作系统:32 位 Linux 操作系统。

在 Windows 7 系统中准备了一个 675 MB 大小的文件,通过 mongofiles 工具将文件存储到分布式的 MongoDB 数据库中进行测试。这个文件在系统中拷贝大约耗时 23.6 s;将该文件存储到 MongoDB 数据库耗时 94.8 s,是文件系统中的 4 倍。考虑到文件系统的虚拟和动态分配,网络的虚拟桥接和单核的性能限制,这个结果在可接受范围内。接着再从 MongoDB 中把数据取出来,这个过程花费了 37 s,只比文件系统多出 14 s。在如此恶劣的条件下有这样的表现,足以证明 MongoDB 在文件存取方面的优越性能。而流媒体服务大多以读取文件为主,所以证明,这样的构架能够解决流媒体服务器上的数据的分布式存储与访问的问题。

## 6 结语

本文通过前端表现、流媒体服务器和分布式数据库三个层次阐述了一个新的视频直播、录制及流媒体数据存储的方案,通过对系统连接数和文件存取性能的测试,得出的结果表明本文方案在工程实践中应用是可行的。

### 参考文献:

- [1] SHI X, DONG J, FU Y. Application the video living system in the live telecast of academic exchange [J]. Computer Programming Skills & Intenance, 2013(12): 153 - 154. (史晓翠,董家集,傅扬. 视频直播系统在学术交流中的应用[J]. 电脑编程技巧与维护, 2013(12): 153 - 154.)
- [2] LIN J, YU L. Streaming media technology and its application in Web-based education system [J]. Journal of Computer Applications, 2006, 26(S2): 303 - 304. (林建华,俞利君. 流媒体技术及其在网络教育系统中的应用[J]. 计算机应用, 2006, 26(增刊 2): 303 - 304.)
- [3] WU W. Discussion on the design and application of live video conferencing system[J]. China Computer & Communication: Theory, 2013(5): 185 - 186. (吴文权. 论视频会议直播系统的设计及应用[J]. 信息与电脑: 理论版, 2013(5): 185 - 186.)
- [4] PANG Z. Real-time voice communication platform for research and implementation based on Flex and Red5 technologies [D]. Beijing: Beijing University of Posts and Telecommunications, 2011. (庞振. 基于 Flex + Red5 的实时语音交流平台的研究与实现[D]. 北京: 北京邮电大学, 2011.)
- [5] LI L, DUAN J. Research and application of streaming media technologies[J]. Fujian Computer, 2013, 29(4): 68 - 69. (李兰兰, 段军利. 流媒体技术应用与研究[J]. 福建电脑, 2013, 29(4): 68 - 69.)
- [6] LI J, WANG J, WU W, et al. Design and implementation of Web video conferencing system based on Red5[J]. Radio Communications Technology, 2012, 38(6): 56 - 58. (李俊, 王静, 吴巍, 等. 基于 Red5 网页版视频会议系统设计与实现[J]. 无线电通信技术, 2012, 38(6): 56 - 58.)
- [7] DONG L, XIAO N. Road to master of Adobe Flex[M]. Beijing: Publishing House of Electronics Industry, 2009: 4 - 13. (董飞龙, 肖娜. Adobe Flex 大师之路[M]. 北京: 电子工业出版社, 2009: 4 - 13.)
- [8] ZHOU Y. Red5 tutorial [EB/OL]. (2010-06-02)[2013-02-02]. <http://wenku.baidu.com/view/dcf9c1fdc8d376eeaea31c9.html>. (周英科. Red5 入门教程[EB/OL]. (2010-06-02)[2013-02-02]. <http://wenku.baidu.com/view/dcf9c1fdc8d376eeaea31c9.html>.)
- [9] SUN C. The extended design, test and optimization of streaming media server Red5[D]. Shanghai: Shanghai Jiaotong University, 2009. (孙超. 流媒体服务器 Red5 的扩展设计、测试与优化[D]. 上海: 上海交通大学, 2009.)
- [10] SHI G, LI Z, XU Z. Design and implementation of real-time video streaming system based on Red5[J]. Instrumentation Technology, 2010(6): 13 - 15. (石刚, 李子平, 徐志立. 基于 Red5 的流媒体实时视频系统的设计与实现[J]. 仪表技术, 2010(6): 13 - 15.)
- [11] PARMAR R, THORNBURGH M. Adobe's real time messaging protocol [EB/OL]. [2013-06-15]. [http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf).
- [12] LIU C, XU P. The design and implementation of cloud database service based on MongoDB [EB/OL]. [2011-12-14]. <http://www.paper.edu.cn/releasepaper/content/201112-359>. (刘长延, 徐鹏. 基于 MongoDB 实现云数据库服务的设计与实现[EB/OL]. [2011-12-14]. <http://www.paper.edu.cn/releasepaper/content/201112-359>.)
- [13] CHEN H. Research on distribution storage of mass astronomical image based on NoSQL [D]. Kunming: Kunming University of Science and Technology, 2012. (陈慧英. 基于 NoSQL 数据库的海量天文图像分布存储研究[D]. 昆明: 昆明理工大学, 2012.)
- [14] CAI L Q. Design and implementation of cloud monitoring with MongoDB [D]. Beijing: Beijing Jiaotong University, 2011. (蔡柳青. 基于 MongoDB 的云监控设计与应用[D]. 北京: 北京交通大学, 2011.)
- [15] FU Z J. Design and implementation of Web video surveillance system based on Flex [J]. Fujian Computer, 2011, 27(3): 1 - 2. (傅卓军. 基于 Flex 的 Web 视频监控系统的设计与实现[J]. 福建电脑, 2011, 27(3): 1 - 2.)
- [16] CHODOROW K, DIROLF M. MongoDB: the definitive guide [M]. CHENG X, translated. Beijing: Posts & Telecommunications Press, 2011: 145 - 149. (CHODOROW K, DIROLF M. MongoDB 权威指南[M]. 程显峰, 译. 北京: 人民邮电出版社, 2011: 145 - 149.)
- [17] LIU L, DONG X. Research on flash server Red5 [J]. Net Security Technologies and Application, 2009(6): 78 - 79. (刘璐, 董小国. Red5 Flash 服务器研究[J]. 网络安全技术与应用, 2009(6): 78 - 79.)
- [18] BIAN Y. Development of Web video, audio system based on Red5 [D]. Qingdao: Qingdao University, 2012. (卞园渊. 基于 Red5 的 Web 视频、音频系统开发[D]. 青岛: 青岛大学, 2012.)