

Compute

Java ASP.NET PHP Oracle PostgreSQL MySQL

博客园 首页 新随笔 联系 订阅XML 管理

随笔 - 406 文章 - 2 评论 - 21 trackbacks - 1

≤	2016年4月							≥
日	一	二	三	四	五	六		
27	28	29	30	31	1	2		
3	4	5	6	7	8	9		
10	11	12	13	14	15	16		
17	18	19	20	21	22	23		
24	25	26	27	28	29	30		
1	2	3	4	5	6	7		

昵称: [Dufe王彬](#)
园龄: 8年
粉丝: 26
关注: 3
[+加关注](#)

搜索

找找看

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

最新随笔

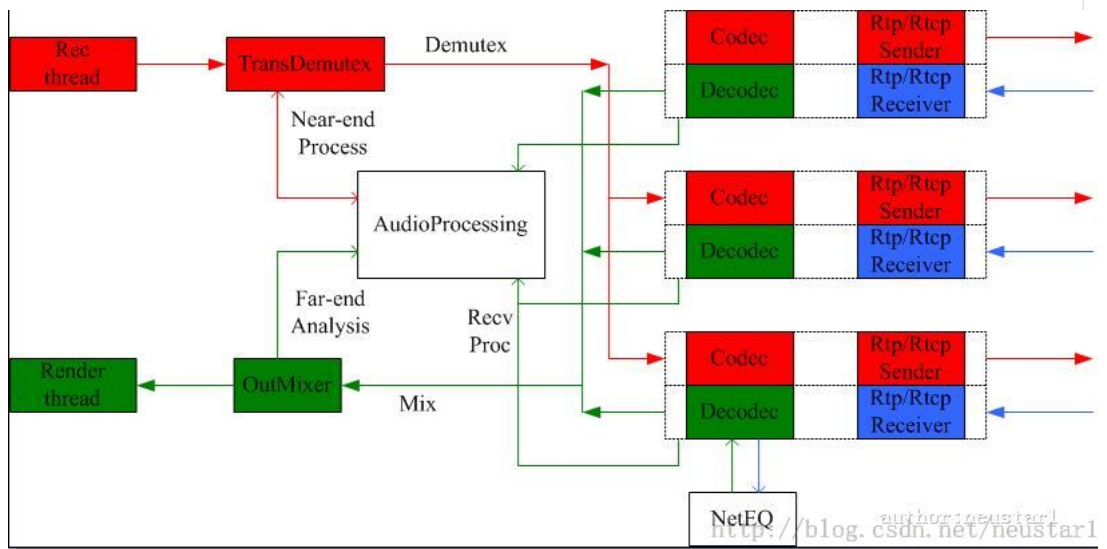
- [1. Lua中调用C函数](#)
- [2. C++ 用libcurl库进行http通讯网络编程\(转\)](#)
- [3. SkipList 跳表](#)
- [4. ntohs, ntohs, htons, htonl 的比较和详解【转】](#)
- [5. SQLITE3 使用总结\(转\)](#)
- [6. 软件测试人员必备Linux命令\(初、中、高级\)](#)
- [7. epoll用法【整理】](#)
- [8. 智能指针-C++](#)
- [9. 关于std::auto_ptr std::shared_ptr std::weak_ptr](#)
- [10. CentOS6.5 一键安装vpn + 添加账号](#)

随笔分类(398)

- [ADO.NET\(1\)](#)
- [Android\(12\)](#)
- [AOP\(1\)](#)
- [ASP.NET\(6\)](#)
- [C Programming\(7\)](#)
- [C#.Net\(6\)](#)
- [C++\(16\)](#)
- [CodeSmith\(1\)](#)
- [DIV CSS\(41\)](#)
- [Enterprise Library\(1\)](#)
- [ERP分类\(1\)](#)
- [Flex\(4\)](#)
- [FreeBSD\(5\)](#)
- [html5\(1\)](#)
- [ipad](#)
- [iphone](#)
- [JAVA\(10\)](#)
- [JS\(30\)](#)
- [Linux\(44\)](#)
- [lua\(2\)](#)
- [MySQL\(21\)](#)
- [NOSQL\(1\)](#)

WebRTC源码分析：音频模块结构分析

一、概要介绍WebRTC的音频处理流程，见下图：



webRTC将音频会话抽象为一个通道Channel，譬如A与B进行音频通话，则A需要建立一个Channel与B进行音频数据传输。上图中有三个Channel，每个Channel包含编解码和RTP/RTCP发送功能。

以一个Channel而言，应用程序中将包含三个活动线程，录音线程，音频接收线程和播放线程。

1) 录音线程：负责麦克风音频的采集，见图中红色路径，采集到音频后，缓存到一定长度，进行音频处理，主要包括EC，AGC和NS等。然后送到Channel，经过音频

Codec模块编码，封装成RTP包，通过Socket发送出去；

2) 接收线程：见蓝色路径，负责接收远端发送过来的音频包，解封RTP包，解码音频数据，送入NetEQ模块缓存。

3) 播放线程：负责耳机声音播放，见绿色路径。播放线程去OutMixer中获取要播放的音频数据，首先依次获取参与会话的Channel中NetEQ存储的音频帧，可以对其做AGC和NS处理；然后混合多个Channel的音频信号，得到混合音频，传递给AudioProcessing模块进行远端分析。最后播放出来。

如下为本地回环录音和播放代码：

```
VoiceEngine* ve = VoiceEngine::Create();
VoEBase* base = VoEBase::GetInterface(ve);
base->Init();
int chId = base->CreateChannel();
base->SetSendDestination(chId,3000,"127.0.0.1",4000);
base->SetLocalReceiver(chId,3000,3001,"127.0.0.1");
base->StartPlayout(chId);
base->StartReceive(chId);
base->StartSend(chId);

//....sleep...wait....

base->StopSend(chId);
base->StopReceive(chId);
base->StopPlayout(chId);
base->Terminate();
```

本文介绍WebRTC音频模块组成和结构，详细介绍音频引擎的配置和启动，相信看完本文后，很多人可以利用WebRTC完成一个音频通话程序开发。

一、对外接口

音频部分的对外主要接口如下，各个接口之间的关系如图1所示。

- 1) VoiceEngine：负责引擎的所有接口查询，存储共享数据信息ShareData。
- 2) VoEBase：负责音频处理的基本操作。
- 3) VoEAudioProcessing：音频信号处理接口，设置各个音频处理项的参数。
- 4) VoECodec：音频编解码接口，提供支持的编解码器查询，音频编解码设置。

[Oracle\(2\)](#)
[P2P\(1\)](#)
[Perl\(1\)](#)
[PHP\(63\)](#)
[PostgreSQL\(31\)](#)
[Project Management\(1\)](#)
[SHELL\(7\)](#)
[SQL\(13\)](#)
[UML\(3\)](#)
[Unity3D\(4\)](#)
[webrtc\(4\)](#)
[WebService\(2\)](#)
[xcode\(2\)](#)
[大型系统架构\(9\)](#)
[翻译分类](#)
[管理_营销\(5\)](#)
[加密解密\(5\)](#)
[架构设计\(5\)](#)
[软件测试\(7\)](#)
[软件开发\(1\)](#)
[设计模式](#)
[数据结构与算法\(1\)](#)
[水晶报表\(1\)](#)
[搜索引擎\(5\)](#)
[网页技术\(2\)](#)
[项目管理\(12\)](#)

随笔档案(406)

[2016年3月 \(3\)](#)
[2016年1月 \(3\)](#)
[2015年12月 \(4\)](#)
[2015年11月 \(1\)](#)
[2015年10月 \(4\)](#)
[2015年9月 \(1\)](#)
[2015年8月 \(1\)](#)
[2015年7月 \(2\)](#)
[2015年5月 \(1\)](#)
[2015年4月 \(8\)](#)
[2014年8月 \(2\)](#)
[2013年11月 \(1\)](#)
[2013年10月 \(3\)](#)
[2013年3月 \(1\)](#)
[2013年2月 \(1\)](#)
[2013年1月 \(1\)](#)
[2012年2月 \(3\)](#)
[2012年1月 \(1\)](#)
[2011年12月 \(7\)](#)
[2011年11月 \(2\)](#)
[2011年10月 \(3\)](#)
[2011年9月 \(19\)](#)
[2011年8月 \(4\)](#)
[2011年7月 \(1\)](#)
[2011年6月 \(3\)](#)
[2011年5月 \(3\)](#)
[2011年4月 \(2\)](#)
[2011年3月 \(2\)](#)
[2011年2月 \(17\)](#)
[2011年1月 \(11\)](#)
[2010年12月 \(24\)](#)
[2010年11月 \(31\)](#)
[2010年10月 \(4\)](#)
[2010年9月 \(14\)](#)
[2010年8月 \(4\)](#)
[2010年7月 \(3\)](#)
[2010年6月 \(1\)](#)
[2010年5月 \(6\)](#)
[2010年4月 \(23\)](#)
[2010年3月 \(1\)](#)
[2010年2月 \(2\)](#)
[2010年1月 \(3\)](#)
[2009年12月 \(3\)](#)
[2009年11月 \(2\)](#)
[2009年10月 \(2\)](#)
[2009年8月 \(13\)](#)
[2009年7月 \(4\)](#)
[2009年6月 \(32\)](#)
[2009年5月 \(16\)](#)
[2009年4月 \(29\)](#)
[2009年3月 \(13\)](#)
[2009年2月 \(8\)](#)
[2008年11月 \(3\)](#)

5) VoEHardware: 音频硬件设备接口, 负责音频硬件设备的设置。

其它的接口还有VoENetEqStats, VoENetwork, VoERTP_RTCP, VoEVideoSync, VoEVolumeControl, VoEFile, VoECallReport, VoEDtmf, VoEMeidaProcess和VoEEncryption。

WebRTC使用继承实现接口转换和查询, 接口之间的数据共享是通过ShareData完成, 首先VoiceEngineImpl继承各个对外接口的实现, 所以可以从VoiceEngineImpl很容易获取其他对外接口。而VoiceEngineImpl本身也继承ShareData, 当从VoiceEngineImpl获取其他对外接口的同时, 隐式的传递了ShareData指针, 因此各个接口可以很方便的获取到ShareData的数据信息。因此虽然类与类之间的关系看起来比较混乱, 但是使用上比较方便。

利用VoiceEngine获取对外接口: VoEInterfaceXX* pInterf = VoEInterfaceXX::GetInterface(pVoiceEngine);

二、模块组成

主要由五大模块组成: AudioDeviceModule音频设备模块, AudioProcess音频处理模块, AudioCodingModule音频编码模块, AudioConferenceMixer混音模块和RtpRtcp传输模块。

ShareData用于粘合各个模块之间的关系, 负责管理全局的对象, 包括AudioDeviceModule, TransmitMixer, OutputMixer, ChannelManager和AudioProcess。

录音流程: AudioDeviceWinCore负责采集音频数据, 传递到AudioDeviceBuffer中缓存, AudioDeviceBuffer则将数据送入TransmixMixer, 首先交给AudioProcess进行近端音频处理, 完成后分发到各个Channel中, Channel则通过AudioCodingModule进行编码, 编码后再交付到RtpRtcp中经由RTPSender发送出去。

接收流程: RTPReceiver负责接收音频RTP包, 接收到RTP包后交给Channel, Channel转交给AudioCodingModule中的ACMNetEQ模块, 进行解码缓存。

播放流程: Channel从ACMNetEQ模块中取出缓存的解码音频数据, 如果需要进行远端数据处理的话, 传递给AudioProcess处理。最后所有Channel都汇入到OutputMixer中进行混音, 混音后再传递到AudioProcess进行远端音频分析。最后送入AudioDeviceModule中的AudioDevceWinCore播放。

三、配置

1、音频引擎创建与删除

```
VoiceEngine*pVoeEngine = VoiceEngine::Create();

VoiceEngine::Delete(pVoeEngine);
```

2、音频收发

1) 音频通话链路创建

WebRTC中的Channel, 为一路音频。作为网络语音通信, 至少要创建一路音频Channel。

Channel没有提供对外接口, 是有VoEBase来管理的, 通过索引号来选定对应的Channel。

```
VoEBase*base = VoEBase::GetInterface(pVoeEngine);

int ch0 =base->CreateChannel();
```

2) 网络端口设置

音频通过RTP和RTCP发送出去, RTP和RTCP使用UDP实现, 需要配置网络端口和地址。

```
//设置发送给.2机器的3000端口

base->SetSendDestination(ch0,3000,"192.168.8.2");

//在本机的3000端口接收RTP包

base->SetLocalReceiver(ch0,3000);
```

3) 音频编码选择

VoECodec负责编解码的配置。

```
VoECodec*codec = VoEBase::GetInterface(pVoeEngine);

设置Channel的编码类型之前, 要查询支持的编码列表。

CodecInstnst;

Intnum = codec->NumOfCodecs();

for(int i=0; i<num; ++i)

{

    Codec->GetCodec(I,inst);

    //打印编码信息

}

//设置编码0

Codec->GetCodec(0,inst);

Codec->SetSendCodec(ch0,inst);

WebRTC自动识别编码类型, 因此解码不需要设置。
```

4) 启动

启动播放: base->StartPlayout(ch0);该操作含义是将通话ch0进行混音输出。

启动接收: base->StartReceive(ch0);开始接收后, 每增加一路通话, 引擎会将音频进行混音再输出。

启动发送: base->StartSend(ch0);启动发送的时候, 会检查是否正在录音, 如果已经开启录音, 则不再开启; 否则会执行音频

2008年10月 (14)

2008年9月 (6)

2008年8月 (12)

2008年7月 (18)

文章分类(1)

大型系统架构(1)

相册(3)

Photo(3)

JS

《快品牌》博客！金错刀看商业2.0

《快品牌》博客！金错刀看商业2.0

mvext

最新评论

1. Re:简单的PHP+SMARTY分页类

下一页这个地方page+1不是-1...

--Curtis

2. Re:简单的PHP+SMARTY分页类

哥 借用你这个总算搞定了那个smarty的分页，这个里面类里面有个bug，当点击下一页的时候出现的！

--Curtis

3. Re:C++ 16进制转10进制

@xyqing525我也是菜鸟，学点东西，就记上，以后用就看下，要不然不做就忘了。。^_^...

--Dufe王彬

4. Re:C++ 16进制转10进制

向技术大牛学习,看到你的随笔档案,08-15年间的随笔记录,真的膜拜!

--xyqing525

5. Re:mysql自动备份脚本

您好，请教个问题:1.根据所设置的路径，如：/root/，默认保存那个盘符里面的？2.根据您贴出的脚本，根据自己数据库密码等设置后（好像没有设置数据库名的），执行后出现一个文件\$LogFile，用记.....

--CreateKang

阅读排行榜

1. Linux shell脚本的字符串截取(14486)

2. PgSql备份pg_dump与还原手记pg_restore (转) (7707)

3. C# 10进制与16进制相互转换(3831)

4. 在sql中取系统时间?日期?年?(2947)

5. php soap实例讲解(2556)

评论排行榜

1. 电脑前的朋友注意坐姿 -详图解说(6)

2. 客观公正地评价MySQL和PostgreSQL优劣(转)(2)

3. 简单的PHP+SMARTY分页类(2)

4. C++ 16进制转10进制(2)

5. Linux+Nginx+Php架设高性能WEB服务器(1)

设备录音操作。

3、音频处理的配置

VoEAudioProcessing负责音频处理的配置。

VoEAudioProcessing*pAudioProc = VoEAudioProcessing::GetInterface(pVoeEngine);

//启动AGC功能

pAudioProc->SetAgcStatus(true);

4、音频设备的配置

VoEHardware接口可以查看录音和播放设备，可以选择指定的设备进行音频通话。

VoEHardware*pHardware=VoEAudioProcessing::GetInterface(pVoeEngine);

Int numin =pHardware->GetNumOfRecordingDevices();

For(int i=0;i<numin; ++i)

{

pHardware->GetRecordingDeviceNames(...)

//打印录音设备

}

//选择设备0作为录音设备

pHardware->SetRecordingDevice(0);

播放设备配置类似。

分类: [webrtc](#)

好文要顶

关注我

收藏该文

[Dufe王彬](#)
[关注 - 3](#)
[粉丝 - 26](#)
[+加关注](#)

00

(请您对文章做出评价)

« 上一篇: [webrtc 的回声抵消\(aec、aecm\)算法简介\(转\)](#)

» 下一篇: [WebRtc VoiceEngine代码解析](#)

posted on 2015-04-28 14:10 [Dufe王彬](#) 阅读(559) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

[【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库](#)

[【推荐】融云即时通讯云 - 豆果美食、Faceu等亿级APP都在用](#)

微软Dynamics
在用什么开发工具？

马上了解

最新IT新闻:

- [揭秘：微软要用聊天机器人取代app？](#)
- [比办公桌还小的涡轮机：能为1万户家庭供电](#)
- [Ubuntu GNOME 16.04 LTS发布](#)
- [欢迎来到隐形革命的时代](#)
- [给16GB容量iPhone用户的一枚彩蛋](#)

» [更多新闻...](#)

90%的开发者在用

极光推送

不只是稳定

最新知识库文章:

- [架构漫谈（九）：理清技术、业务和架构的关系](#)
- [架构漫谈（八）：从架构的角度看如何写好代码](#)
- [架构漫谈（七）：不要空设架构师这个职位，给他实权](#)
- [架构漫谈（六）：软件架构到底是要解决什么问题？](#)
- [架构漫谈（五）：什么是软件](#)

» [更多知识库文章...](#)