流媒体服务概要设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| **日 期** | **制定人** | **审核人** | **批准人** |
| 2014-07-07 |  |  |  |

版权所有

**版本记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **作者/日期** | **变化内容描述** | **审核人/日期** | **批准人/日期** |
| V0.1 | 2014/7/7 | 创建 |  |  |
| V0.2 | 2014/8/4 | 添加详细设计内容 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 引言

## 编写目的

本说明书基于需求说明，用于明确流媒体服务的系统架构、各个功能模块，以用模块间的关联关系，并以此为依据，制定详细技术设计方案。

本说明书的预期读者为: 研发工程师，项目经理等。

## 背景

本说明书所描述的产品大概信息如下:

软件名称: 流媒体服务

软件标识: Streaming Media Services

软件版本: 1.0

开发者:

## 1.3 定义

1. SMS: Streaming Media Services, 流媒体服务产品

## 1.4 引用文档

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标识 | 标题 | 修订版本 | 日期 | 来源 |
| 1. | 流媒体服务项目需求说明书.docx |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 概述

## 需求概述

流媒体服务平台，要求能对采集的直播流提供大批量用户并发的点播和直播服务；

具体需求及设计目标请查阅《流媒体服务项目需求说明书.docx》。

# 组成与架构设计

## 流媒体服务组成与架构

流媒体服务组成框图如下:



流媒体服务的架构图如下:



注:

1.DB 表示数据库(Database)

2.绿色箭头线条表示直播数据流；红色箭头线条表示点播数据流。

蓝色双向箭头线条表示后台管理交互数据流。

黑色双向箭头线条表示用户直播与点播播放交互数据流。

### 3.1.1硬件层

#### 3.1.1.1 PC平台: RTMP推送

PC平台，负责音视频数据的采集、处理与合成，并依据配置输出RTMP流。

#### 3.1.1.2 嵌入式平台:RTMP推送

嵌入式平台，负责音视频数据的采集、处理与合成，并依据配置输出RTMP流。

#### 3.1.1.3 PC平台&嵌入式平台素材与FTP client

PC平台和嵌入式平台都能将音视频数据采集后，进行本地化的收录并存储，在需要时，将平台作为一个FTP客户端，通过FTP协议进行指定素材的上传。

#### 3.1.2 流媒体服务系统层

#### 3.1.2.1 RTMP中继系统

作为RTMP流的中继，负责接收PC平台和嵌入式平台推送的RTMP流；并转发给其它系统。

提供统一的REST API，包括获得流接收服务器的IP；RTMP流的状态，推流端信息，拉流端信息，结点及整个系统流的计数，以及系统的状态等。

系统的参数可配置，具有自动负载均衡，状态自检等功能。

集群结点可热删除和扩展。

#### 3.1.2.2 RTMP实时转码系统

从RTMP中继系统取得RTMP流，按转码配置参数进行实时转码输出。

提供统一的REST API，包括配置转码输出参数；转码任务的启动、停止、重转；查询转码任务的状态，转码路计数，以及系统的状态；

系统内部可配置、具有自动负载均衡，状态自检等功能。

集群结点可热删除和扩展。

#### 3.1.2.3 RTMP直播分发系统

从RTMP转码系统中取得转码后的RTMP流，按配置进行RTMP直播分发；

提供统一的REST API，包括配置分发参数；分发任务的启动、停止；分发任务的状态、路数计数，以及查询系统的状态；

系统内部可配置，具有自动负载均衡，状态自检等功能。

集群结点可热删除和扩展。

#### 3.1.2.4 分布式文件系统

提供大容量的分布式文件存储空间；

包含有FTP server服务，用于接收通过FTP协议上传的素材；

包含有文件分发服务，可实现简单的文件分发功能；

系统可实现集群结点的热删除和扩展，实现容量的动态增长；

提供统一的 REST API，包括查看文件系统容量，已用空间等。

#### 3.1.2.5 RTMP实时录制系统

从RTMP中继系统中取得RTMP流后，将RTMP流存储成文件；

提供统一的REST API，包括配置录制参数；录制任务的启动、停止；录制任务的状态、路数计数，以及系统的状态；

系统内部可配置，具体自动负载均衡，状态自检等功能。

集群结点可热删除和扩展。

#### 3.1.2.6 文件转RTMP系统

将分布式文件系统中的文件转码成RTMP流，推送到RTMP中继系统；

提供统一的REST API，包括配置转码参数；转码任务的启动、停止；转码任务的状态、路数计数，以及系统的状态；

系统内部可配置，具有自动负载均衡，状态自检等功能；

集群结点可热删除和扩展。

#### 3.1.2.7 文件转码系统

对分布式文件系统中的文件进行转码再存储；

提供统一的REST API，包括配置转码参数；转码任务的启动、停止；转码任务队列的状态，任务优先级设置；转码任务的状态、路数计数，以及系统的状态；

系统内部可配置，具有自动负载均衡，状态自检等功能；

集群结点可热删除和扩展。

#### 3.1.2.8 点播与直播服务系统

支持点播和直播业务的播放请求；

实现用户的动态负载均衡；

系统内部可配置，具有状态自检等功能；

集群结点可动态扩展；

#### 3.1.2.9 后台管理系统

实现对各个子系统的配置、管理和监控；

包含有数据库，以用于基本的数据收集和存储；

提供统一的REST API， 实现对系统Web 可视化的管理界面。

#### 3.1.2.10 播放数据收集系统

收集播放器的数据，并用于用户观看数据的统计，包括用户观看数，用户观看内容，用户观看时长等；

### 3.1.3 播放层

#### 3.1.3.1 PC端播放器

PC端的直播与点播播放器，支持HTTP和RTMP传输协议；

能实现定制化的播放器效果，可方便地嵌入到播放页面；

支持用户播放数据的收集并回传、日志记录等；

#### 3.1.3.2 IOS

IOS端的直播与点播播放器，支持HLS传输协议；

能实现定制化的播放器效果，可方便地嵌入到播放页面；

支持用户播放数据的收集并回传、日志记录等；

#### 3.1.3.3 ANDROID

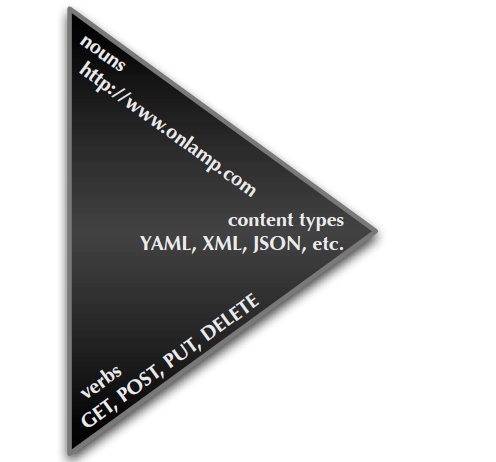
ANDROID端的直播与点播播放器，支持HLS传输协议；

能实现定制化的播放器效果，可方便地嵌入到播放页面；

支持用户播放数据的收集并回传、日志记录等；

# 4.系统详细设计

目前国内外的很多大型网站、系统都在使用RestAPI的方式来设计WebService层的接口，这样设计有很多的好处，符合面向对象的思想，能够使我们的代码具备更多的维护性，系统更灵活的扩展性等等。以下的三角图可以很好的说明Rest的原理。



在我们的流媒体系统中，存在各种操作对象，比如转码、分发、中继、后台管理等。这些都可以抽象为Rest API的操作对象，操作类型则为HTTP协议中的GET、POST、PUT、DELETE，可以分别对应为对操作对象的查询、增加、修改、删除动作，也能映射到数据库的CRUD操作。

## 4.1 RTMP中继系统

### 4.1.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3

重要技术: Tornado-4.0, CRTMPServer version1.1, Nginx version 1.4.2

### 4.1.2 REST API功能设计

1. 获得本结点的配置参数列表；

2. 修改本结点的配置参数；

3. 启动/停止/重启crtmpserver实例；

4. 获得本结点的接收RTMP的信息；

5. 获得本结点的拉取RTMP的信息；

6. 获得本结点的某路RTMP的接收与拉取信息；

7. 获得所有结点的RTMP的接收与拉取信息(仅用于中心机);

8. 获得分配的接收RTMP的IP和端口信息；

9. 获得本结点的日志信息；

### 4.1.3 节点机组成框图



### 4.1.4 集群组成框图



## 4.2 RTMP实时转码系统

### 4.2.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3

重要技术: Tornado-4.0, FFmpeg version 0.8.15, Nginx version 1.4.2

### 4.2.2 REST API功能设计

1. 获得本结点的配置参数列表；

2. 修改本结点的配置参数；

3. 启动/停止/重启本节点的crtmpserver实例；

4. 启动/停止/重启指定的某路转码

5. 获得本结点的转码流信息、计数；

5. 获得所有结点的转码流信息、计数(仅用于中心机)；

6. 获得本结点的日志信息；

### 4.2.3 节点机组成框图



### 4.2.4 集群组成框图



## 4.3 RTMP直播分发系统

### 4.3.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3

重要技术: Tornado-4.0, FFmpeg version 0.8.15, Nginx version 1.4.2

### 4.3.2 REST API功能设计

1. 获得本结点的配置参数列表；

2. 修改本结点的配置参数；

3. 启动/停止/重启指定的某路分发

4. 获得本结点的分发流信息、计数；

5. 获得所有结点的分发流信息、计数(仅用于中心机)；

6. 获得本结点的日志信息；

### 4.3.3节点机组成框图



### 4.3.4 集群组成框图



## 4.4 分布式文件系统

### 4.4.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

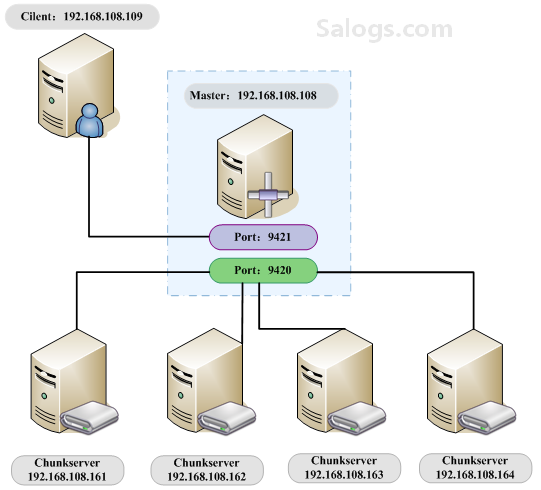
重要技术: Tornado-4.0, Nginx version 1.4.2, MooseFS 1.6

### 4.4.2 REST API功能设计

1. 查看文件系统容量，可用空间，已用空间；

2. 查看系统状态；

### 4.4.3 系统组成框图



## 4.5 RTMP实时录制系统

### 4.5.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3

重要技术: Tornado-4.0, FFmpeg version 0.8.15, Nginx version 1.4.2

### 4.5.2 REST API功能设计

1. 获得本结点的配置参数列表；

2. 修改本结点的配置参数；

3. 启动/停止/重启指定的某路录制

4. 获得本结点的录制流信息、计数；

5. 获得所有结点的录制流信息、计数(仅用于中心机)；

6. 获得本结点的日志信息；

### 4.5.3 节点机组成框图



### 4.5.4 集群组成框图



## 4.6 文件转RTMP系统

### 4.6.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3

重要技术: Tornado-4.0, FFmpeg version 0.8.15, Nginx version 1.4.2

### 4.6.2 REST API功能设计

1. 获得本结点的配置参数列表；

2. 修改本结点的配置参数；

3. 启动/停止/重启指定的某路录制

4. 获得本结点的录制流信息、计数；

5. 获得所有结点的录制流信息、计数(仅用于中心机)；6. 6.3 节点机组成框图



### 4.6.4 集群组成框图



## 4.7 文件转码系统

### 4.7.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3

重要技术: Tornado-4.0, FFmpeg version 0.8.15, Nginx version 1.4.2

### 4.7.2 REST API 功能设计

对分布式文件系统中的文件进行转码再存储；

提供统一的REST API，包括配置转码参数；转码任务的启动、停止；转码任务队列的状态，任务优先级设置；转码任务的状态、路数计数，以及系统的状态；

系统内部可配置，具有自动负载均衡，状态自检等功能；

集群结点可热删除和扩展。

1. 获得本结点的配置参数列表；

2. 修改本结点的配置参数；

3. 启动/停止/重启指定的某路转码；

4. 查看任务队列；

5. 调整在队列中任务的优先级；

6. 获得本结点的转码流信息、计数；

7. 获得所有结点的转码流信息、计数(仅用于中心机)；

8. 获得本结点的日志信息；

### 4.7.3 节点机组成框图



### 4.7.4 集群组成框图



## 4.8 点播与直播服务系统

### 4.8.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3

重要技术: Tornado-4.0, Nginx version 1.4.2

### 4.8.2 集群组成框图



## 4.9 后台管理系统

### 4.9.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : MySQL

重要技术: Tornado-4.0, Nginx version 1.4.2

### 4.9.2 REST API功能设计

1. 查看子系统的设置；

2. 设置子系统的配置；

3. 添加或删除子系统的节点；

4. 操作子系统的功能；

5. 子系统的数据查看；

6. 子系统的日志查看；

7. 子系统的软件版本查看；

8. 子系统的软件版本升级与回滚；

9. 后台管理页面的Web界面展示；

10. 后台管理系统的用户权限设置；

11. 查看后台管理系统的用户操作日志；

### 4.9.3 组成框图



## 4.10 播放数据收集系统

### 4.10.1 系统实现平台

实现环境: Red Hat Enterprise 4/5/6 , Linux version 2.6.18, gcc version 4.1.2

开发语言: C/C++, Python 2.7.3

数据库 : Sqlite3/MySQL

重要技术: Tornado-4.0, Nginx version 1.4.2

### 4.10.2 REST API功能设计

1. 播放器心跳数据访问接口；

2. 播放数据查看；

3. 系统运行日志查看；

### 4.10.3 组成框图

