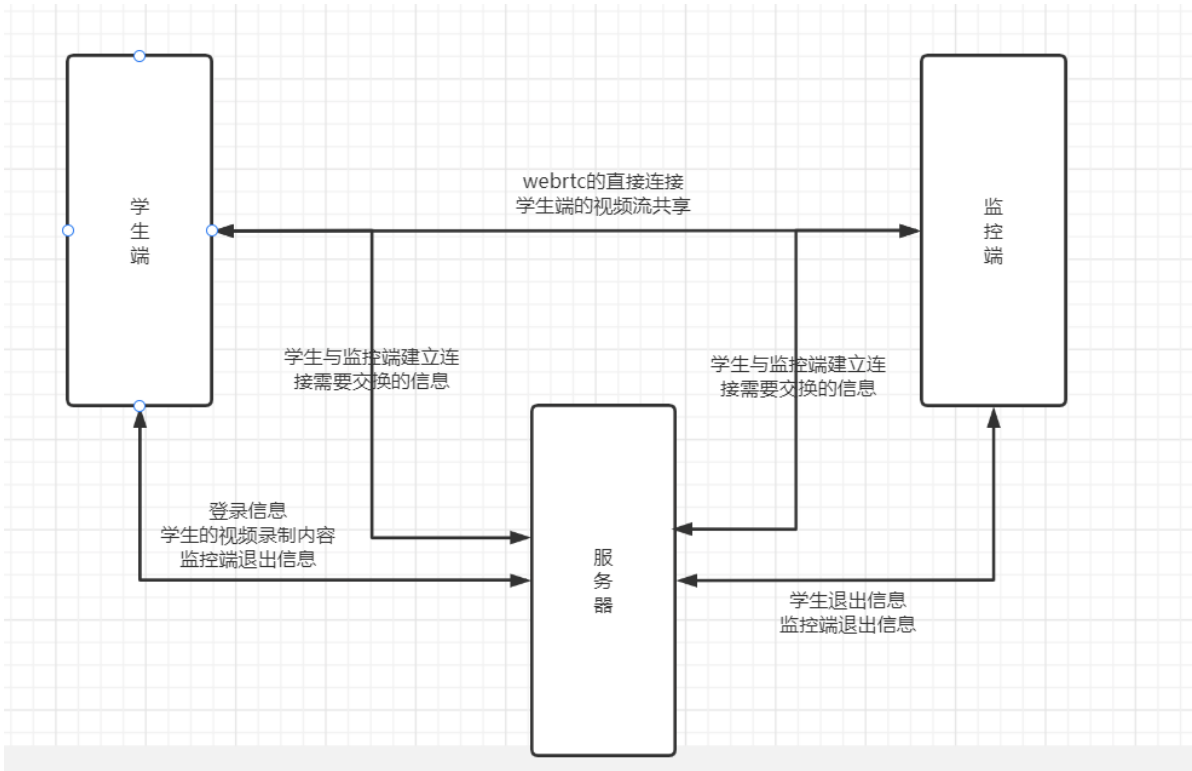


技术手册

技术手册

- 学生端, 监控端, 服务器端的数据流与控制信令关系
- 服务器部署
- 视频的编码和解码标准
- 需要通过修改源码来完成修改的内容
 - webrtc网页的监听端口
 - 数据库的种类和字符集
 - python错误日志的记录位置
- 不同帧率, 码率, 采样率对系统负载的影响
- 遗憾, 不足, 待改进的地方

学生端, 监控端, 服务器端的数据流与控制信令关系



学生端与服务器端建立websocket连接, 主要用于传送视频录制的内容

学生端与监控端直接建立webrtc连接, 用于监控端获取学生端的视频信息

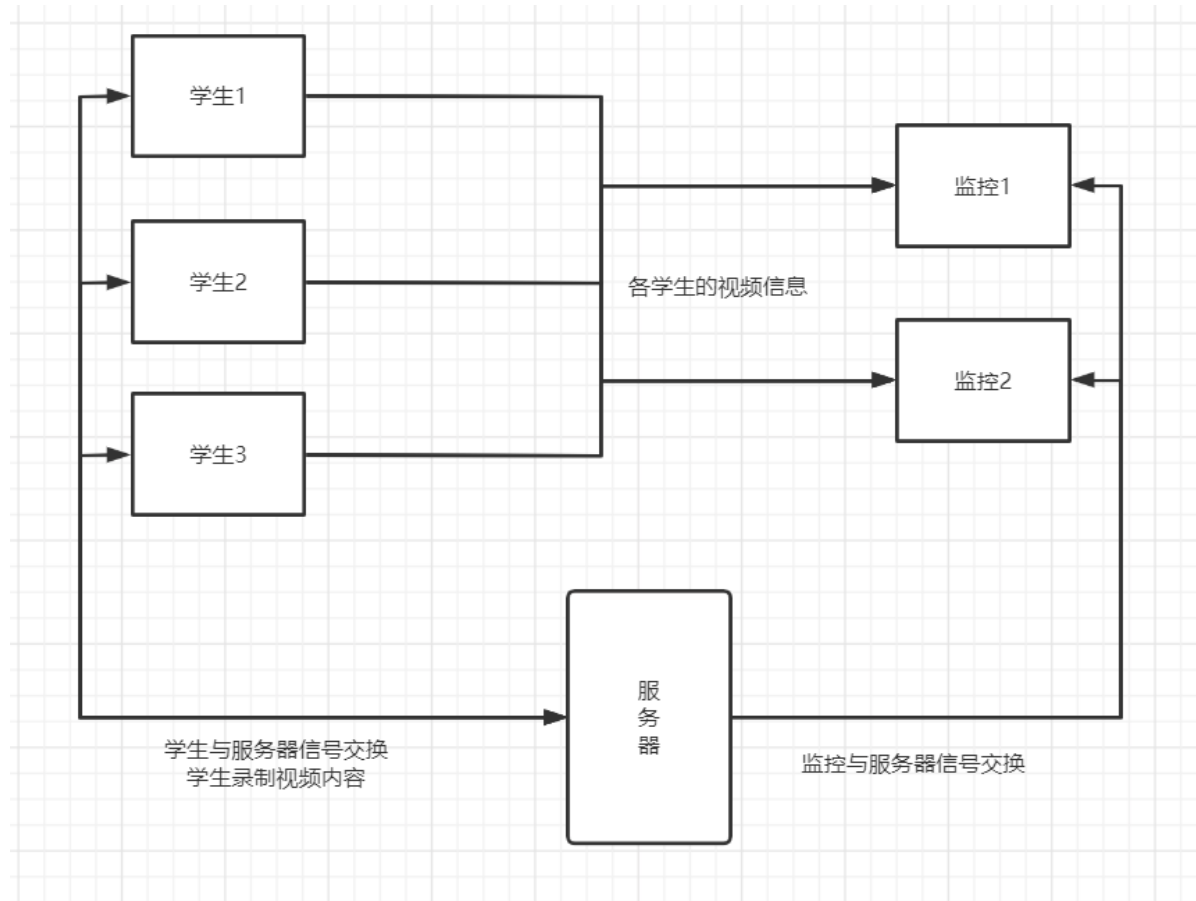
通过这种方式, 可以减少服务器的带宽压力, 使较小的服务器带宽也可以完成webrtc的任务

代价是:

随着监控端数目的增加, 学生端的上行压力会增加

随着学生端数目的增加, 监控端的下行压力会增加

如下图



服务器部署

使用python中的flask和flask相关拓展包进行服务端的部署, 服务端使用python编写, 避免了对apache/php等相关的配置

视频的编码和解码标准

录制时: webm格式

通过ffmpeg进行视频转码, 转码后得到mp4格式

需要通过修改源码来完成修改的内容

webrtc网页的监听端口

在 `/usr/bin/webrtc_Tony/init.py` 中最后一行(734行)

修改port与host的值, 可以修改webrtc的监听端口和监听ip

```
733     ssl_certificate_key = config_info[ 'ssl' ][ 'key' ]
734     socketio.run(app, host='0.0.0.0', port=8080, debug=DEBUG, keyfile=ssl_certificate_key, certfile=ssl_certificate)
735     # ssl_context=(ssl_certificate, ssl_certificate_key)
```

数据库的种类和字符集

在 `/usr/bin/webrtc_Tony/init.py` 中(19行)

修改开头的mariadb为其他值即可, 同理, 通过更改charset的键值即可更改数据库的字符集

```
DB_URL = "mariadb://{username}:{password}@{host}:{port}/{db}?charset=gbk".format(  
    username=config_info['database']['user'],  
    password=config_info['database']['password'],  
    host=config_info['database']['host'],  
    port=config_info['database']['port'],  
    db=DB_DATABASE)
```

python错误日志的记录位置

`/usr/bin/webrtc_Tony/flask_run.sh` 中

```
6  nohup python3 $FLASK_APP > /var/log/webrtc_Tony/error.log 2>&1 &  
7
```

修改 `/var/log/webrtc_Tony/error.log` 即可

不同帧率, 码率, 采样率对系统负载的影响

由于webrtc项目的工作时间较短, 没有测试

目前已知的是ffmpeg在视频转码大文件时会占用较大的内存, webrtc在调用ffmpeg转码时使用的是多线程的方式, 在同时转码大量/过大视频文件时, 可能会触发linux的oomkiller, 导致转码生成的.mp4错误. 此时可以手动调用ffmpeg完成转码

遗憾, 不足, 待改进的地方

这个webrtc项目只用了4天时间, 一个人完成, 时间确实非常的紧张, 有许多缺陷都没有考虑

- webrtc判断学生是否登录使用了cookie, 导致如果学生端/监控端误触了页面刷新, 就会导致页面判断为学生重复登录
- webrtc判断学生掉线只在监控端设置了监控, 如果监控端不登录, 而学生断网, 服务器端会等待学生端重连, 不会认为学生端下线. 只有在学生端关闭页面的时候会检测到学生的下线
- webrtc不支持查看单个学生的状态, 只能全部查看, 这导致这个webrtc不能完成对较多学生的监视, 可能在带宽上会出现问题
- webrtc的监控端开始录制和停止录制机制没有完善, 不能让在停止录制之后加入会议的学生默认停止录制
- webrtc为了不因为视频转码影响系统对网页的访问, 设置了在所有用户都退出后进行视频转码, 在转码期间可能会导致网页访问困难, 这个问题没有时间解决
- webrtc的签名文件没有去对机构的认证, 所以导致浏览器对服务器的证书不信任, 导致在flask在运行的时候会提示关于ssl证书不被信任的错误
- webrtc网站没有对UI进行优化和改进, 导致UI界面不理想