Assignment3

Group Member:

17221294 杨洪源

17301066 张冰诚

项目要求：

Video streaming server providing video streaming service

Messenger enabling chat between/among class mates

In java and with Spring Boot

Using websocket

技术栈：

自搭的阿里云turn服务器，信令服务器使用了websocket

WebRtc框架

用docker在阿里云Centos7上部署Kurento服务器:

Video Streaming:

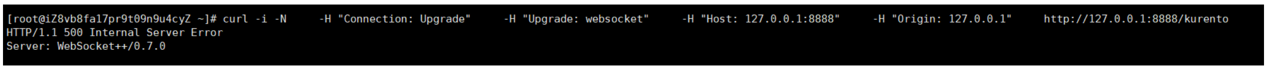
Server:

运用了Java Spring自己的webrtc api

```
const iceServer = {
        "iceServers": [{
            "urls": 'stun:39.99.208.132:3478'
        }, {
            "urls": "turn:39.99.208.132:3478",
            "username": "kurento",
            "credential": "kurento"
        }]
    };
//兼容浏览器的getUserMedia写法
const getUserMedia = (navigator.getUserMedia || navigator.mozGetUserMedia ||
navigator.webkitGetUserMedia || navigator.msGetUserMedia);
//兼容浏览器的PeerConnection写法
const PeerConnection = (window.PeerConnection ||
    window.webkitPeerConnection00 ||
    window.webkitRTCPeerConnection ||
    window.RTCPeerConnection ||
    window.mozRTCPeerConnection);
```

用docker在阿里云Centos7上部署Kurento服务器:

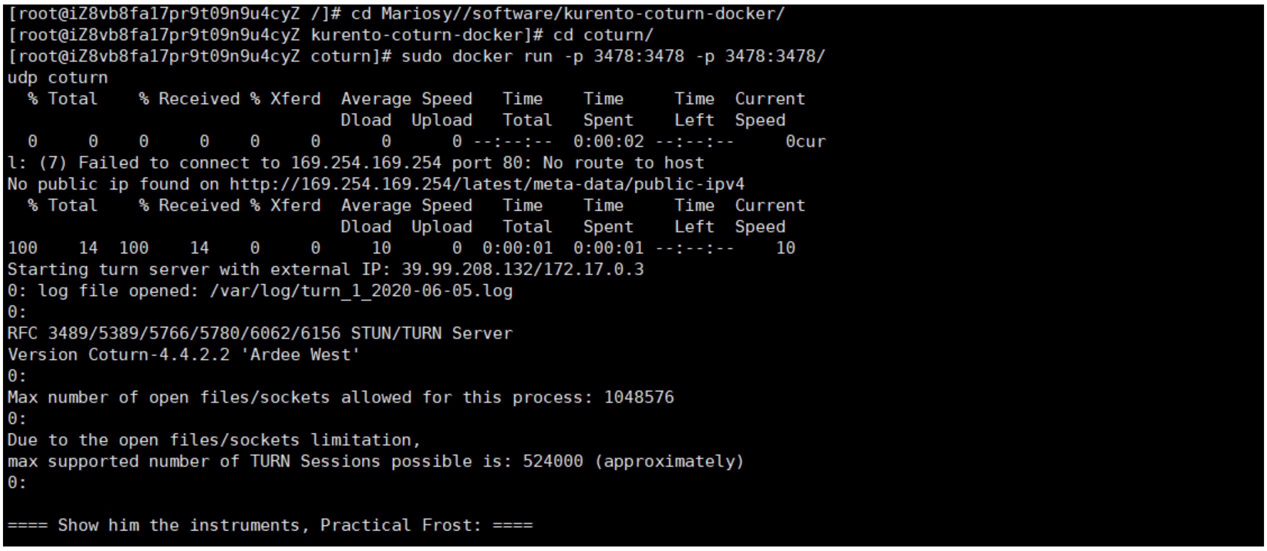
项目由于时间原因，最后未使用到



```
[root@iZ8vb8fa17pr9t09n9u4cyZ ~]# curl -i -N     -H "Connection: Upgrade"     -H "Upgrade: websocket"     -H "Host: 127.0.0.1:8888"     -H "Origin: 127.0.0.1"     http://127.0.0.1:8888/kurento
HTTP/1.1 500 Internal Server Error
Server: WebSocket++/0.7.0
```

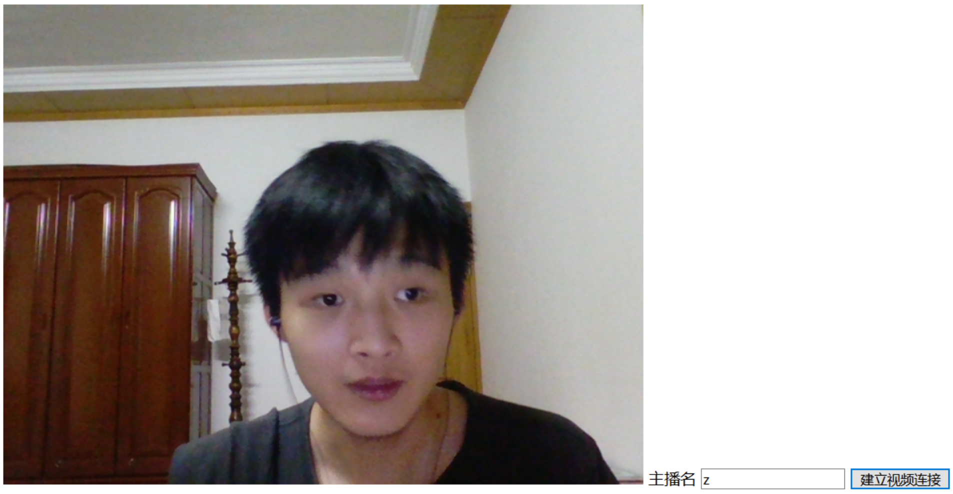
ScreenShot:

线上的turn服务器：

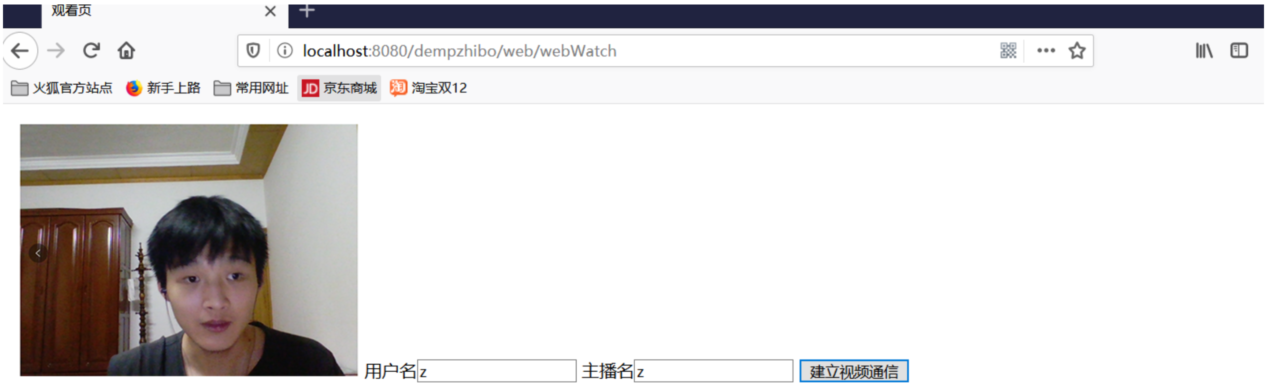


```
[root@iZ8vb8fa17pr9t09n9u4cyZ /]# cd Mariosy//software/kurento-coturn-docker/
[root@iZ8vb8fa17pr9t09n9u4cyZ kurento-coturn-docker]# cd coturn/
[root@iZ8vb8fa17pr9t09n9u4cyZ coturn]# sudo docker run -p 3478:3478 -p 3478:3478/
udp coturn
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:02 --:--:--     0cur
l: (7) Failed to connect to 169.254.169.254 port 80: No route to host
No public ip found on http://169.254.169.254/latest/meta-data/public-ipv4
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    14  100    14    0     0     10      0  0:00:01  0:00:01 --:--:--    10
Starting turn server with external IP: 39.99.208.132/172.17.0.3
0: log file opened: /var/log/turn_1_2020-06-05.log
0:
RFC 3489/5389/5766/5780/6062/6156 STUN/TURN Server
Version Coturn-4.4.2.2 'Ardee West'
0:
Max number of open files/sockets allowed for this process: 1048576
0:
Due to the open files/sockets limitation,
max supported number of TURN Sessions possible is: 524000 (approximately)
0:

==== Show him the instruments, Practical Frost: ====
```

实时主播效果：



实时观看效果：



Using Websocket:

添加pom：

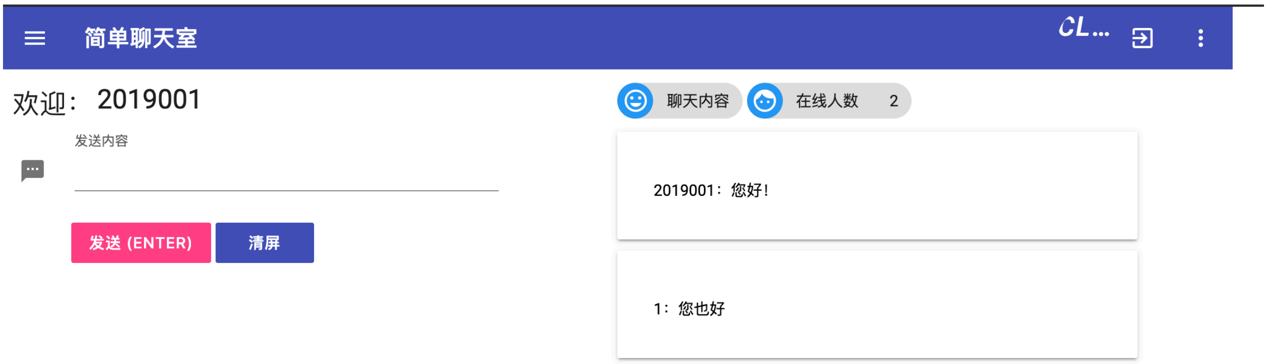
```
<dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-  websocket</artifactId>
</dependency>
```

配置websocket消息服务器：

```
package com.example.demo.WebSocket;

import com.alibaba.fastjson.JSON;
import com.example.demo.bean.Message;
import org.springframework.stereotype.Component;
import javax.websocket.*;
import javax.websocket.server.ServerEndpoint;
import java.io.IOException;
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;

@Component
@ServerEndpoint("/chat")
public class WebSocketChatServer {
    /**

     * 全部在线会话  PS：基于场景考虑 这里使用线程安全的Map存储会话对象。

     */
    private static Map<String, Session> onlineSessions = new
ConcurrentHashMap<>();
    /**

     * 当客户端打开连接：1.添加会话对象 2.更新在线人数

     */
    @OnOpen
    public void onOpen(Session session) {
        onlineSessions.put(session.getId(), session);
        sendMessageToAll(Message.jsonStr(Message.ENTER, "", "",
onlineSessions.size()));
    }

    /**

     * 当客户端发送消息：1.获取它的用户名和消息 2.发送消息给所有人

     * <p>

     * PS：这里约定传递的消息为JSON字符串 方便传递更多参数！

     */
    @OnMessage
    public void onMessage(Session session, String jsonStr) {
        Message message = JSON.parseObject(jsonStr, Message.class);
        sendMessageToAll(Message.jsonStr(Message.SPEAK, message.getUsername(),
message.getMsg(), onlineSessions.size()));
    }

    /**

     * 当关闭连接：1.移除会话对象 2.更新在线人数

     */
    @OnClose
    public void onClose(Session session) {
        onlineSessions.remove(session.getId());
        sendMessageToAll(Message.jsonStr(Message.QUIT, "", "",
onlineSessions.size()));
    }

    /**

     * 当通信发生异常：打印错误日志

     */
    @OnError
    public void onError(Session session, Throwable error) {
        error.printStackTrace();
    }

    /**

     * 公共方法：发送信息给所有人

     */
    private static void sendMessageToAll(String msg) {
        onlineSessions.forEach((id, session) -> {
            try {
                session.getBasicRemote().sendText(msg);
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
    }
}
```

chat Stream：

运行截图：



2019001：您好！

1：您也好

Java with SpringBoot

项目采取Springboot框架

```xml
<dependency>
           <groupId>org.springframework.boot</groupId>
           <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
           <groupId>org.mybatis.spring.boot</groupId>
           <artifactId>mybatis-spring-boot-          starter</artifactId>
           <version>2.1.1</version>
</dependency>
```