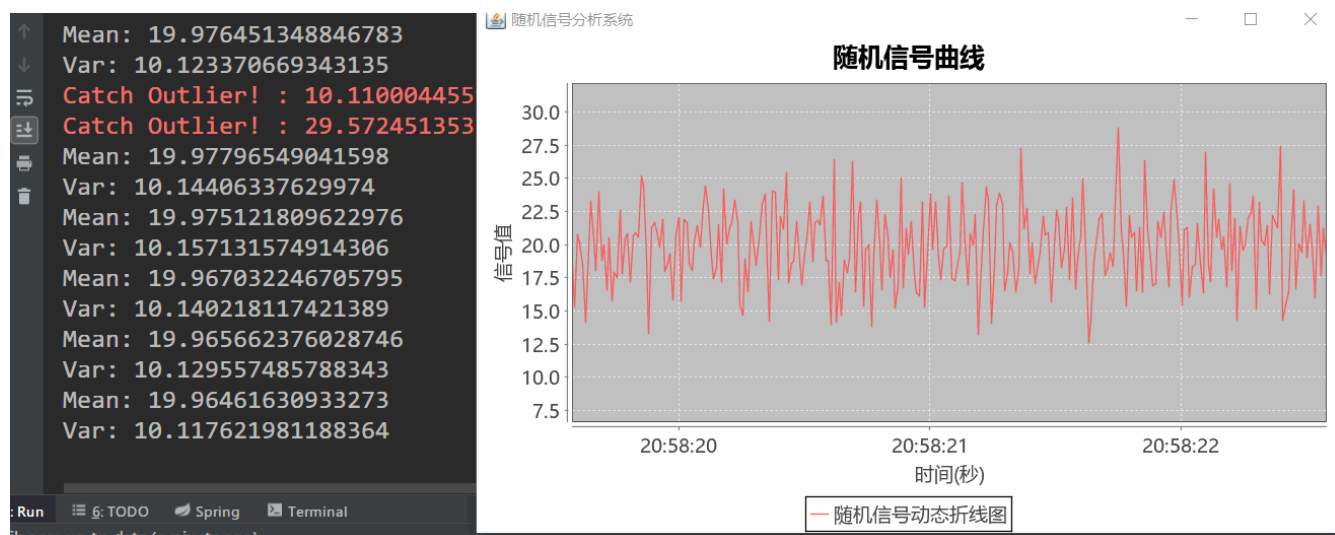


# MOM 消息队列技术实现的随机信号分析系统

## 效果



系统每秒钟输出信号流的方差和均值，检测到异常点值进行警告，并以动态曲线的方式将信号流实时可视化输出。

动态演示效果点击这里查看：[http://media.sumblog.cn/img/20190501210005.gif-min\\_pic](http://media.sumblog.cn/img/20190501210005.gif-min_pic)

## 设计思路

该信号分析系统采用 ActiveMQ 作为 MOM 中间件，使用消息队列的方式，进行随机信号的生成和分析

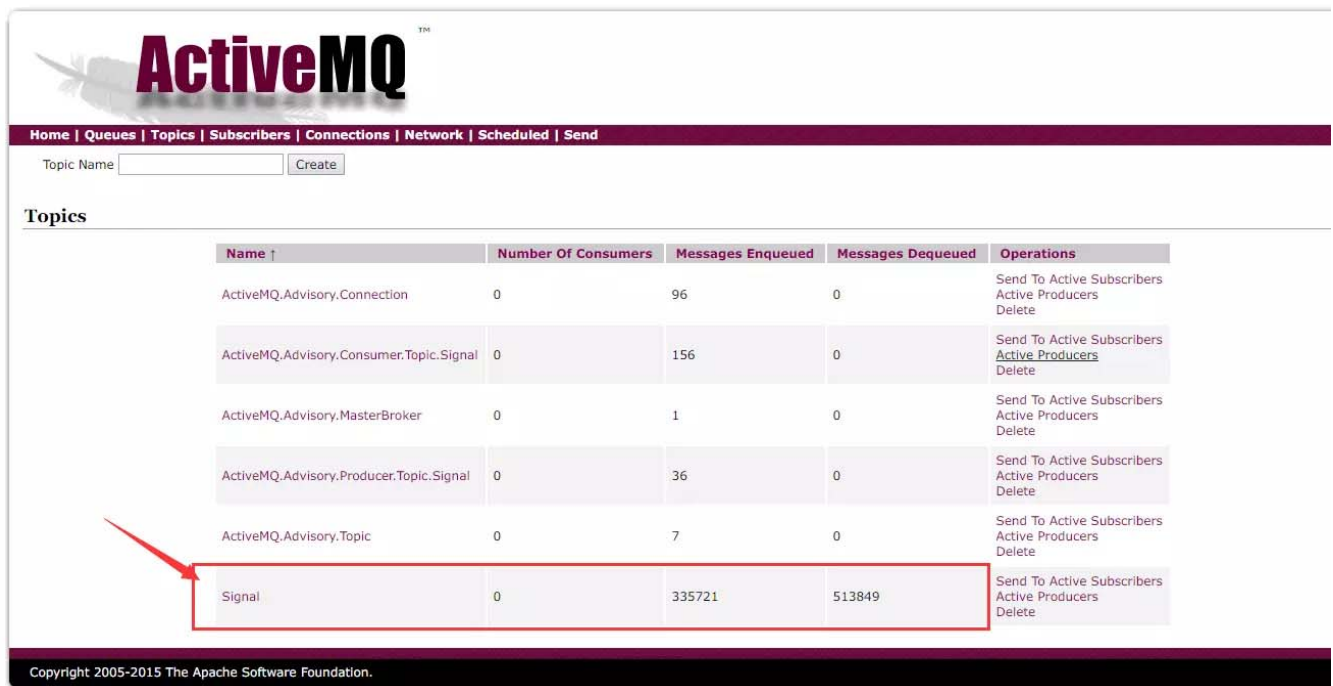
### 1. 随机信号产生器 RandomSignalGenerator

产生器与ActiveMQ 进行连接，并建立“Signal” Topic，每隔 10 ms，将产生的正态分布的随机数字，加入消息队列。

```
1 public static void main(String[] args) throws JMSEException, InterruptedException {
2     RandomSignalGenerator generator = new RandomSignalGenerator("Signal");
3     generator.sendMessage(20,10);
4     generator.close();
5 }
6
7 public void sendMessage(double a, double b) throws JMSEException,
8 InterruptedException {
9     Random random = new Random();
10    while (true){
11        Double sig = Math.sqrt(b)*random.nextGaussian()+a;
12        Message sigMessage = session.createObjectMessage(sig);
13        producer.send(sigMessage);
14        System.out.println(sig);
15        Thread.sleep(10);
16    }
17 }
```

```
16     }
17 }
```

随机数字的生成，采用 java random 模块的 nextGaussian 方法，先生成均值为 0，方差为 1 的随机数，再改变均值和方差



Name ↑	Number Of Consumers	Messages Enqueued	Messages Dequeued	Operations
ActiveMQ.Advisory.Connection	0	96	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Consumer.Topic.Signal	0	156	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.MasterBroker	0	1	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Producer.Topic.Signal	0	36	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Topic	0	7	0	Send To Active Subscribers Active Producers Delete
Signal	0	335721	513849	Send To Active Subscribers Active Producers Delete

## 2. 随机信号分析模块

为了完成题目要求的目标，建立了 `AverageCalculator`、`VarianceCalculator`、`OutliersDetector` 分别完成均值计算，方差计算，异常点检测。这些类实现 `MessageListener` 接口，实现了 `onMessage` 方法，并在 `ListenerController` 类中，通过对应 `MessageConsumer` 的 `setMessageListener` 方法完成注册，使得消息队列中有了新消息后，主动触发这些方法完成对应功能。

例如 `OutlierDetector` 异常点检测类：

```
1  import javax.jms.Message;
2  import javax.jms.MessageListener;
3  import javax.jms.ObjectMessage;
4
5  public class OutliersDetector implements MessageListener {
6
7
8      OutliersDetector(){
9          super();
10     }
11
12     public void onMessage(Message message) {
13         try {
14             Double mean = AverageCalculator.getMean();
15             Double sigma = Math.sqrt(VarianceCalculator.getVariance());
16             Double num = (Double) ((ObjectMessage)message).getObject();
17             if(num > mean + 3*sigma || num < mean - 3 * sigma){
```

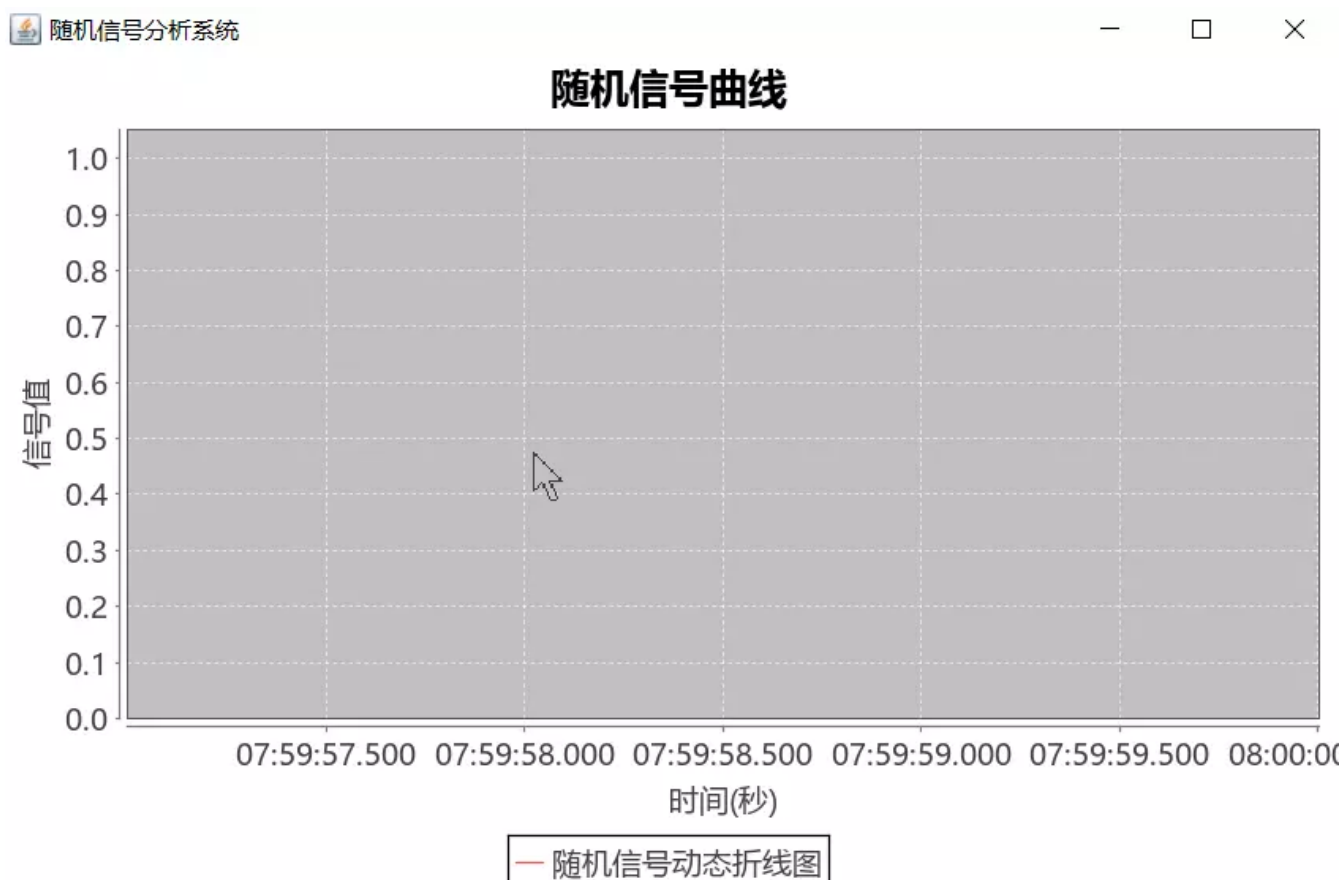
```

18         System.err.println("catch outlier! : " + num);
19     }
20     } catch (Exception e) {
21         e.printStackTrace();
22     }
23 }
24
25 }

```

### 3. 实时信号折线图绘制

项目采用 `JFreeChart` 完成信号折线图的绘制，信号时序数据存放在 `timeSeries` 对象中。编写 `RealTimeChart` 类完成数据的可视化。



为了实现动态更新的功能，编写 `RealTimeChartListener` 类，当队列中有消息时，`onMessage` 方法将信号值和当前时间加入到 `timeSeries` 对象当中，完整折线图的更新。

```
1 class RealTimeChartListener implements MessageListener {
2     public void onMessage(Message message) {
3         try {
4             Double num = (Double) ((ObjectMessage)message).getObject();
5             RealTimeChart.timeSeries.add(new Millisecond(), num);
6         } catch (Exception e) {
7             e.printStackTrace();
8         }
9     }
10 }
```

随机信号曲线

