徐晟恺 10165101265

**华东师范大学 计算机科学与软件工程学院**

摘要

Timeline系统

性能测试报告

——桌面端

目录

[1 测试结果评价与建议 4](#_Toc501373243)

[1.1 测试结果 4](#_Toc501373244)

[1.2 系统性能评价 5](#_Toc501373245)

[1.2.1 交易响应时间 5](#_Toc501373246)

[1.2.2 业务处理能力 5](#_Toc501373247)

[1.2.3 稳定性/健壮性 5](#_Toc501373248)

[1.3 系统性能建议 5](#_Toc501373249)

[2 性能问题解决分析 5](#_Toc501373250)

[2.1 待解决问题 5](#_Toc501373251)

[2.1.1 严重性能问题 5](#_Toc501373252)

[2.1.2 响应时间问题 5](#_Toc501373253)

[2.2 已解决问题 5](#_Toc501373254)

[2.2.1 严重性能问题 5](#_Toc501373255)

[2.2.2 响应时间问题 5](#_Toc501373256)

[3 独立/混合场景结果 5](#_Toc501373257)

[3.1 测试结果描述 5](#_Toc501373258)

[3.2 详细测试结果 5](#_Toc501373259)

[4 峰值测试场景报告 5](#_Toc501373260)

[4.1 测试结果摘要 5](#_Toc501373261)

[4.2 本次测试结论 5](#_Toc501373262)

[4.3 系统性能监控 5](#_Toc501373263)

[4.4 事务出错分析 5](#_Toc501373264)

[5 容量场景测试报告 6](#_Toc501373265)

[5.1 测试结果摘要 6](#_Toc501373266)

[5.2 联机交易结论 6](#_Toc501373267)

[5.2.1 最佳并发数/最佳处理能力 6](#_Toc501373268)

[5.2.2 最大并发数/最大处理能力 6](#_Toc501373269)

[5.2.3 系统容量上限 6](#_Toc501373270)

[5.2.4 系统上限 6](#_Toc501373271)

[5.2.5 业务处理能力 6](#_Toc501373272)

[5.2.6 并发对性能影响 6](#_Toc501373273)

[5.3 批量作业结论 6](#_Toc501373274)

[5.4 系统性能监控 6](#_Toc501373275)

[5.5 事务出错分析 6](#_Toc501373276)

[6 疲劳场景测试报告 6](#_Toc501373277)

[6.1 测试结果摘要 6](#_Toc501373278)

[6.2 测试结论 7](#_Toc501373279)

[6.3 响应时间结果 7](#_Toc501373280)

[6.3.1 第一疲劳响应时间 7](#_Toc501373281)

[6.3.2 末一次疲劳响应时间 7](#_Toc501373282)

[6.4 处理能力结果 7](#_Toc501373283)

[6.4.1 第一疲劳处理能力 7](#_Toc501373284)

[6.4.2 末一次疲劳处理能力 7](#_Toc501373285)

[6.5 系统性能监控 7](#_Toc501373286)

[6.6 事务出错分析 7](#_Toc501373287)

[6.6.1 第一次疲劳分析 7](#_Toc501373288)

[6.6.2 末一次疲劳分析 7](#_Toc501373289)

[7 性能缺陷统计分析 7](#_Toc501373290)

[7.1 缺陷严重程度 7](#_Toc501373291)

[7.2 缺陷类型分析 7](#_Toc501373292)

[7.3 缺陷原因分析 7](#_Toc501373293)

[8 测试环境分析 7](#_Toc501373294)

[8.1 系统架构设计 7](#_Toc501373295)

[8.2 测试环境配置 7](#_Toc501373296)

[8.3 环境差异分析 7](#_Toc501373297)

表目录

**未找到图形项目表。**

# 测试结果评价与建议

## 测试结果

|  |  |
| --- | --- |
| 测试开始日期 | 2018.12.26 |
| 测试结束日期 | 2019.1.7 |
| 提交缺陷数 | 3 |
| 延期/未解决缺陷数 | 1 |
| 拒绝缺陷数 | 0 |
| 总测试场景 | 独立场景、混合场景、峰值场景、容量场景、疲劳场景 |
| 执行场景数 | 独立场景、混合场景、峰值场景、容量场景、疲劳场景 |
| 场景执行率 | 100% |
| 缺陷率（%） | 0.0037% |
| 投入人数 | 1人 |
| 测试生产率 | 13.91分钟/个用例 |
| 系统上线建议 | 可以上线 |

## 系统性能评价

### 交易响应时间

大部分情况下能保证各个事务的平均响应时间小于1秒。

### 业务处理能力

能一直保证事务执行的成功率大于99%。

### 稳定性/健壮性

未做异常测试。

## 系统性能建议

建议最大并发用户数为20。

# 性能问题解决分析

## 待解决问题

### 严重性能问题

无。

### 响应时间问题

本项目中，后端刷新信息功能的实现是，遍历数据库中整个news表然后保存到list中。但是这样的话，刷新信息事务的响应时间会随着news表的增长而增长，而这是不合理的，因为用户刷新信息时只关心最新的那几条数据。因此，后端刷新信息功能的实现存在响应时间的性能问题，可以考虑做出如下修改。

每次刷新改为执行语句：

SELECT \* FROM news ORDER BY id DESC LIMIT 3;

获取news表中id最大（即最新）的三条数据。

加载更多改为：

保存每次浏览到的最小的id。加载更多即为加载id-3,id-2,id-1三条数据，因此执行语句：

SELECT \* FROM news LIMIT id-4, 3;

再把id-3。

由于修改这部分代码的代价太大（涉及单元测试、集成测试等），剩余时间也不充裕，本项目目前未做上述修改。

## 已解决问题

### 严重性能问题

服务器端的cpu性能可能存在瓶颈问题。

.Net内存溢出，服务器端的物理内存不够，可考虑增加内存。

### 响应时间问题

无。

# 独立/混合场景结果

## 测试结果描述

初始化全部用户后，加压5分钟。每个场景测试多组并发、并发数从20逐步增加到50。

两个独立场景（分别为send\_message事务和refresh\_message事务）的事务平均响应时间都能在1秒之内。

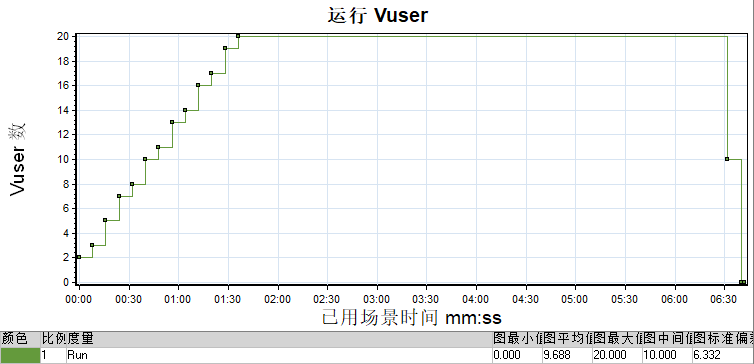
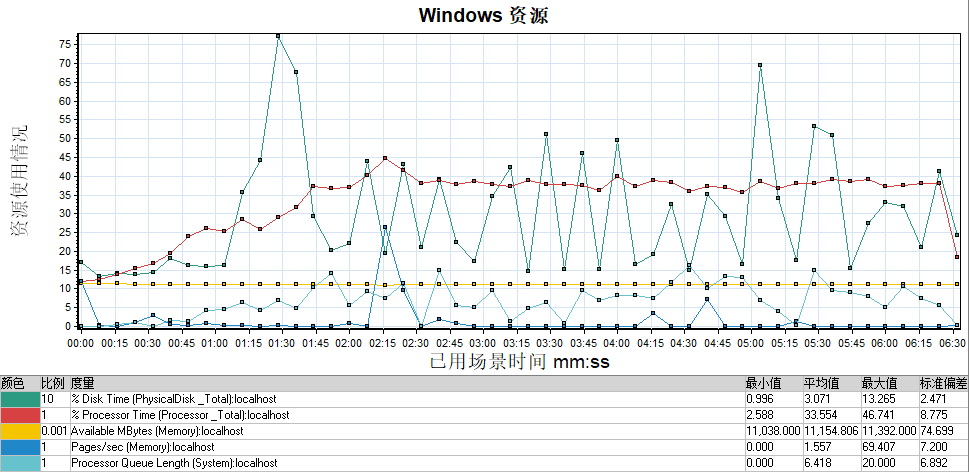
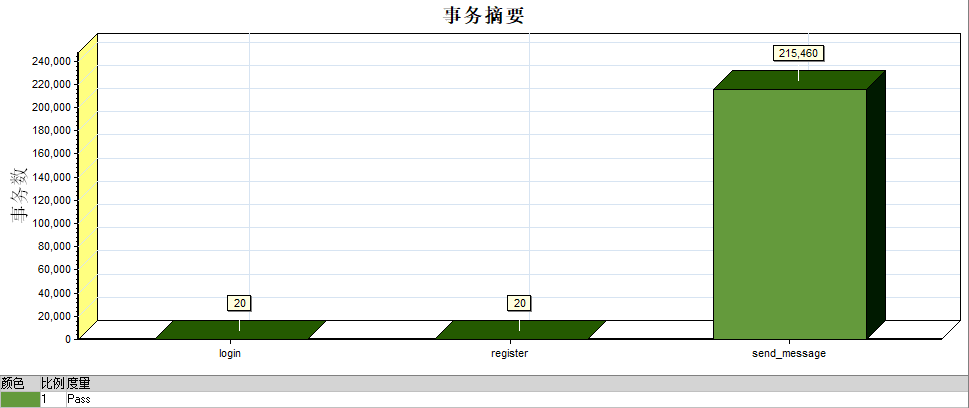
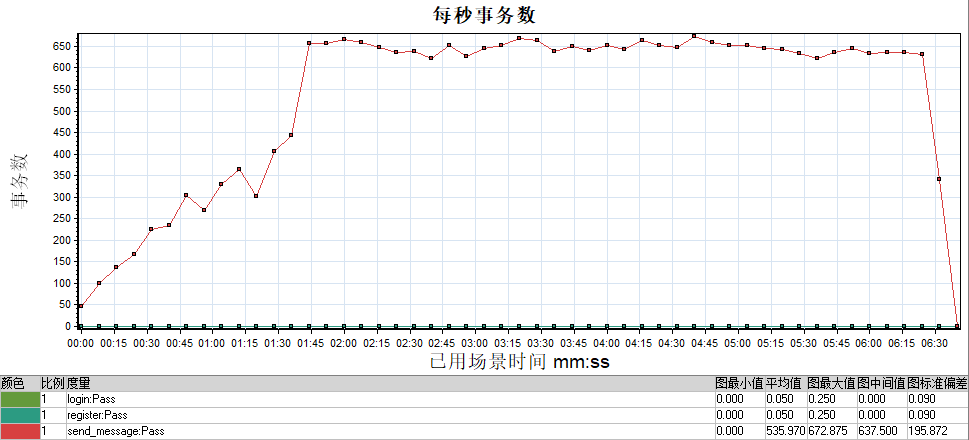
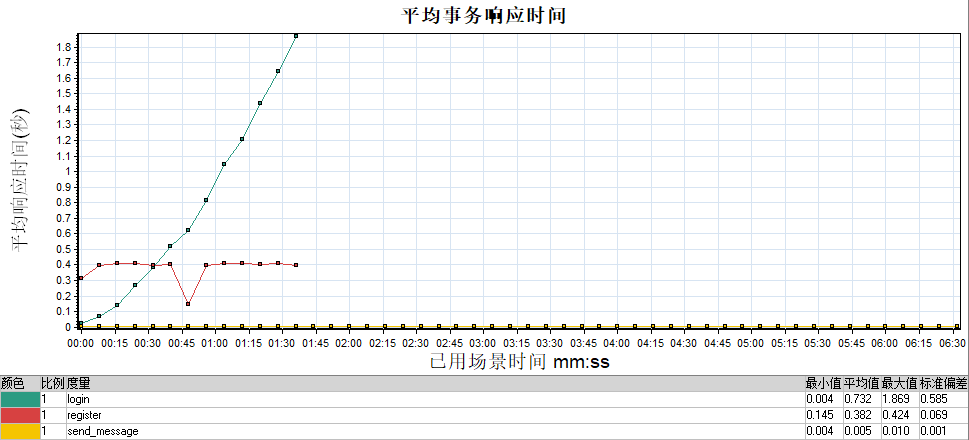
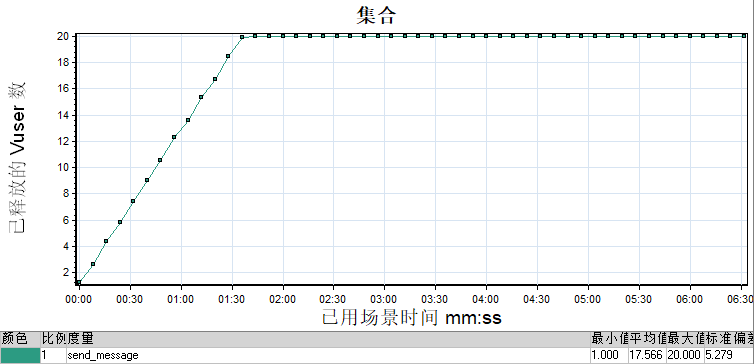
混合场景（执行send\_message事务，refresh\_message事务，more\_message事务）的总体平均响应时间会达到1~2秒。

所有测试场景的性能基本满足要求。

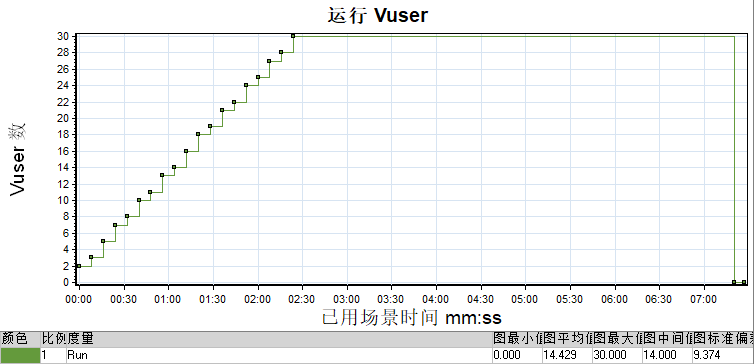
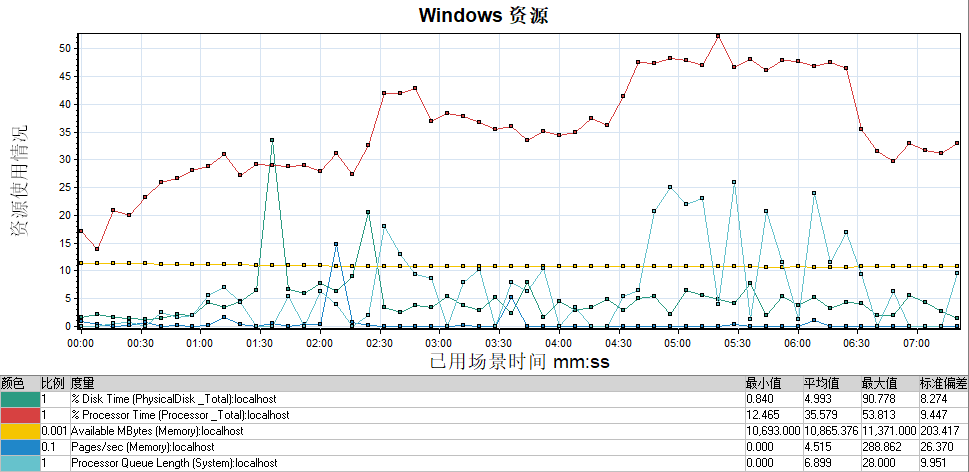
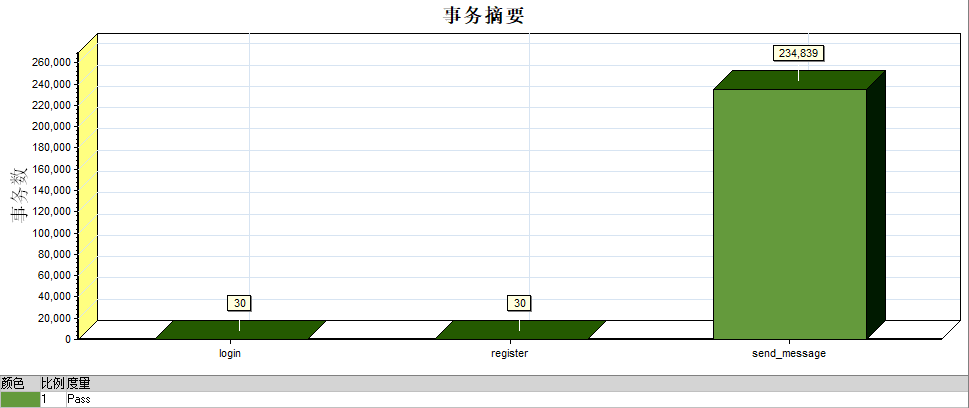
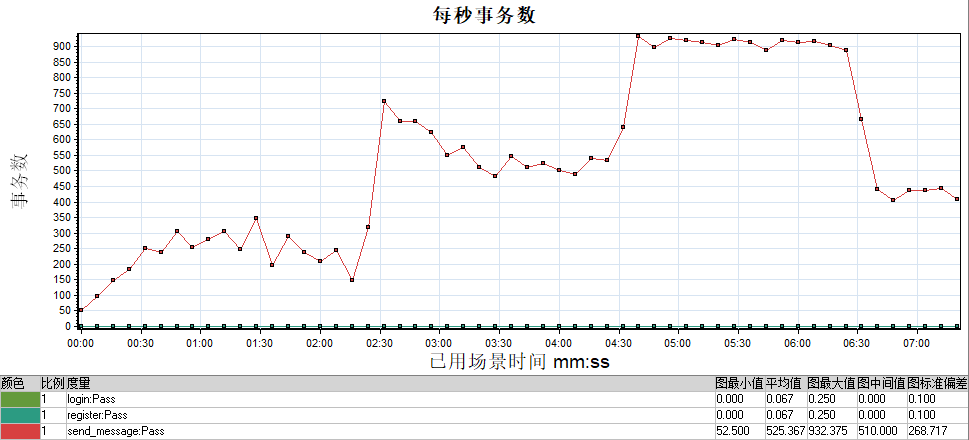
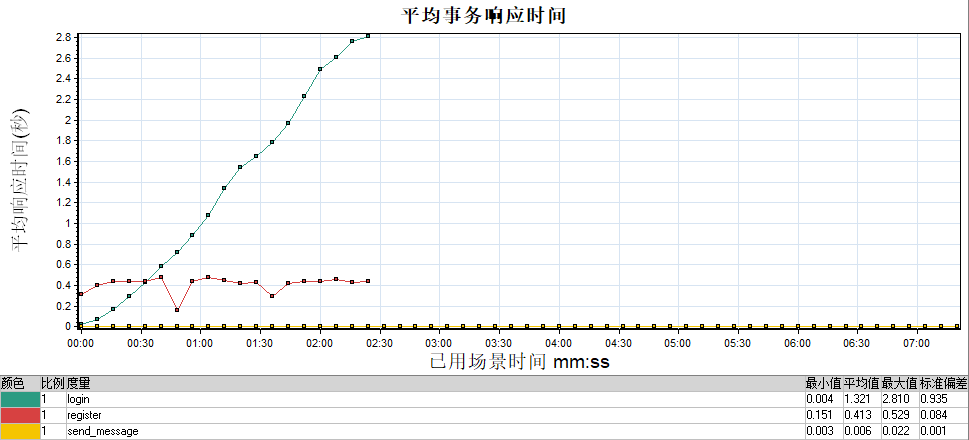
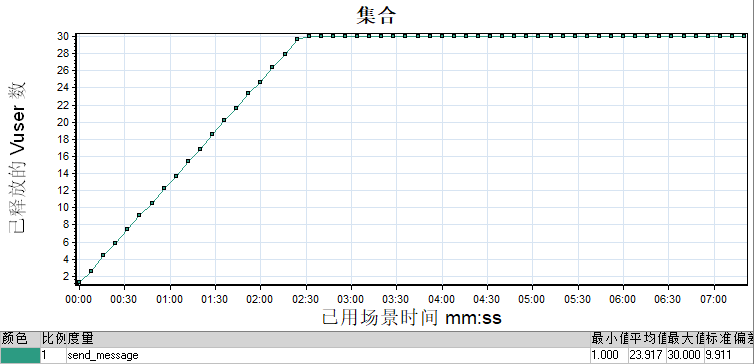
## 详细测试结果

独立场景1：多个并发用户同时点击”发布消息”按钮并发布消息

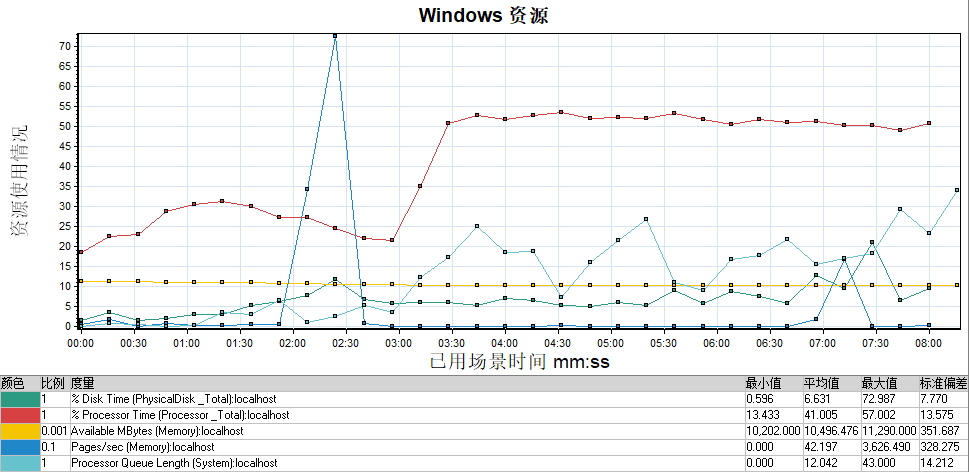
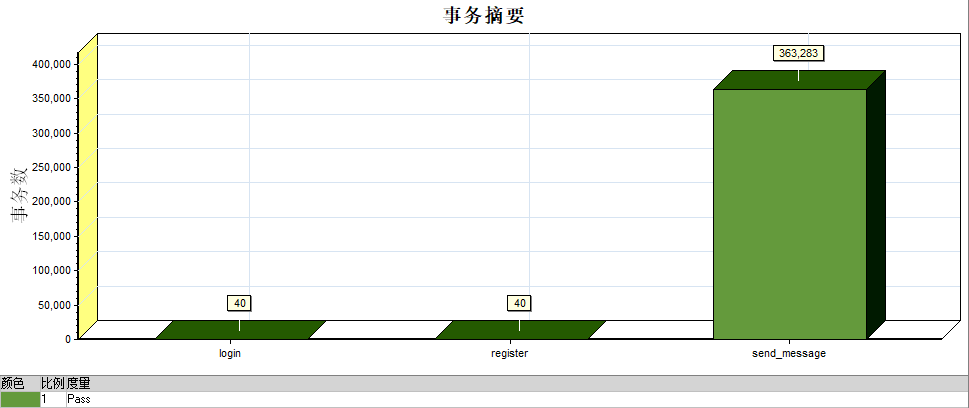
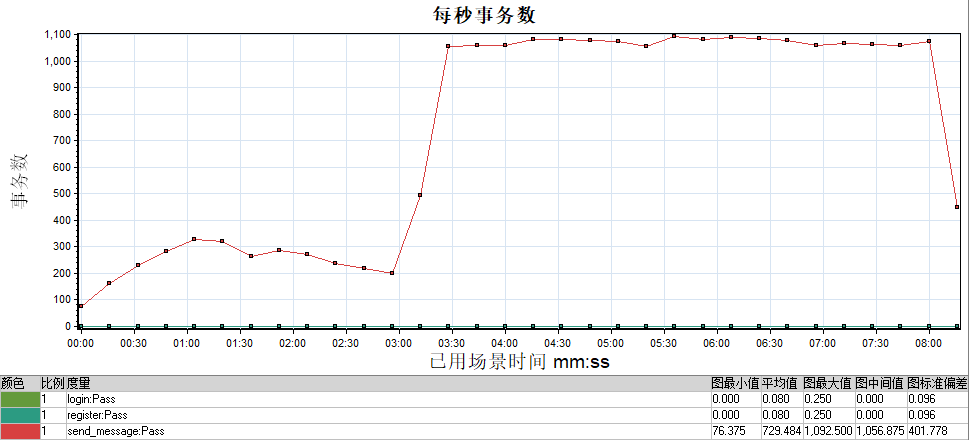
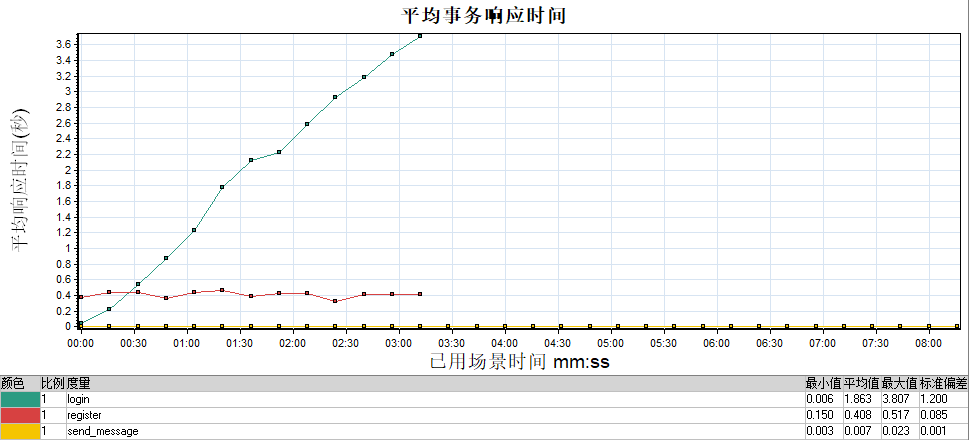
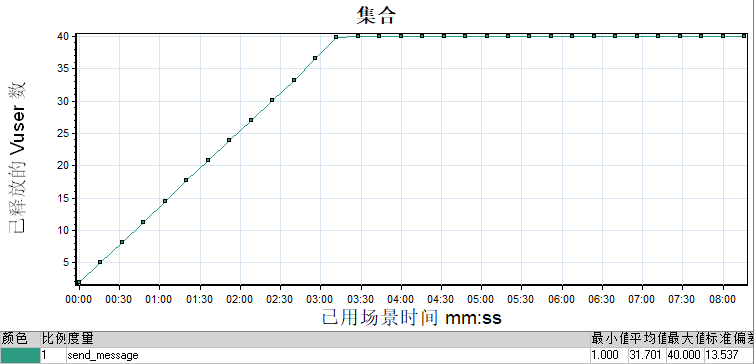
20并发用户数

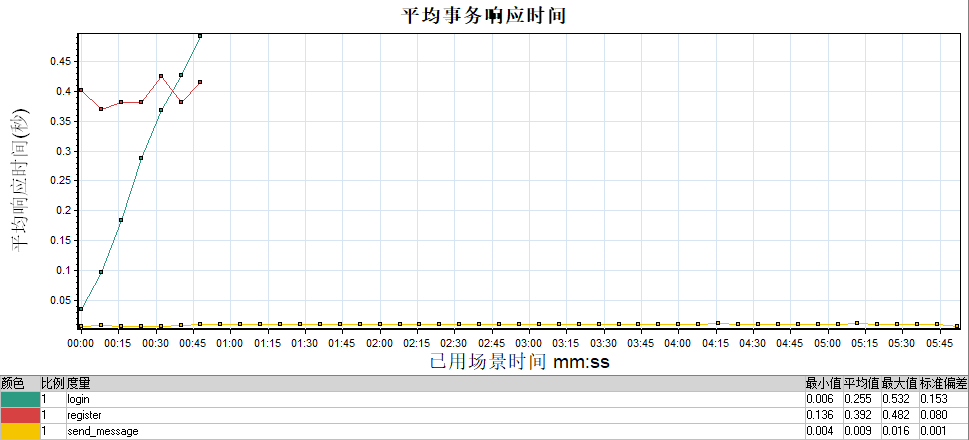
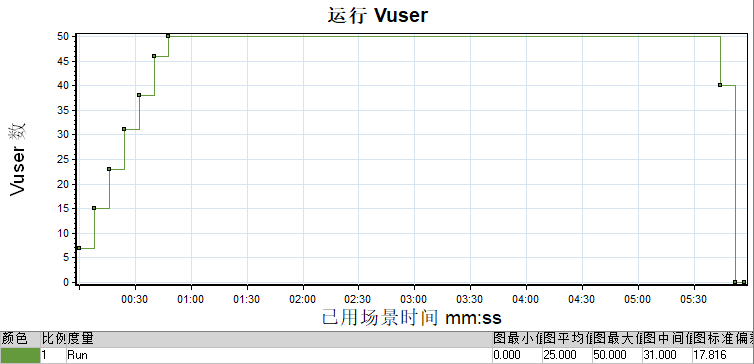
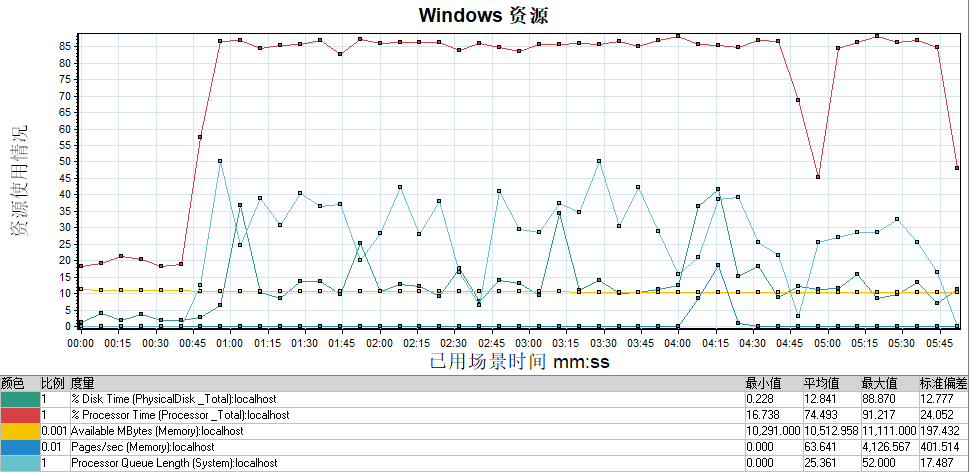
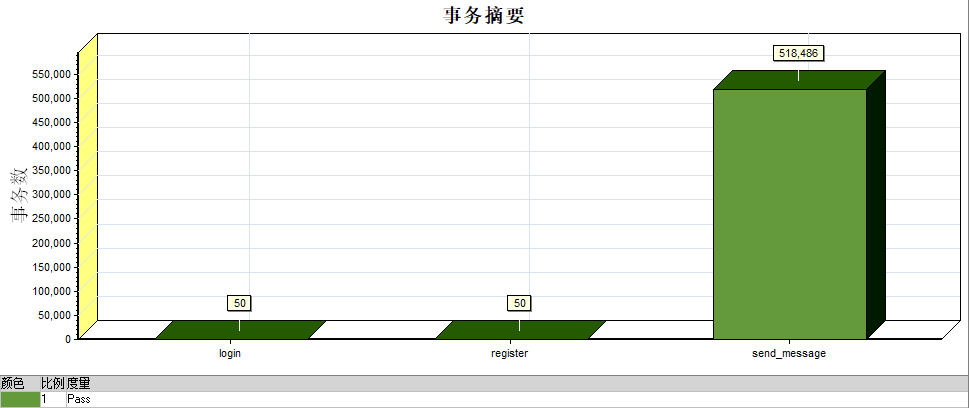
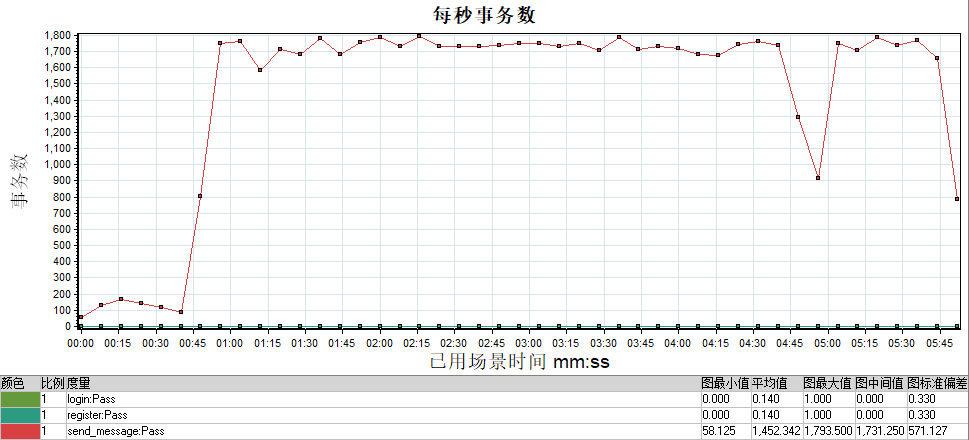
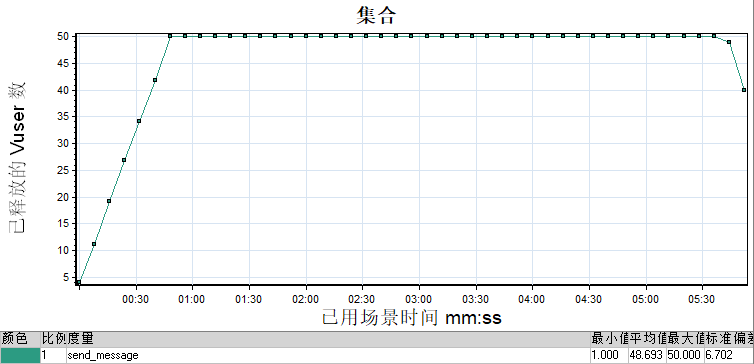
30并发用户数

40并发用户数

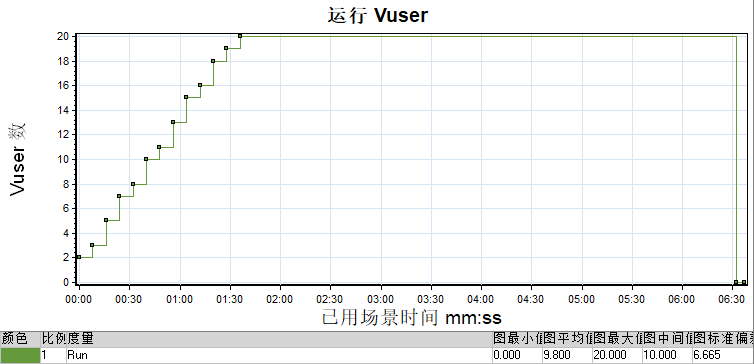
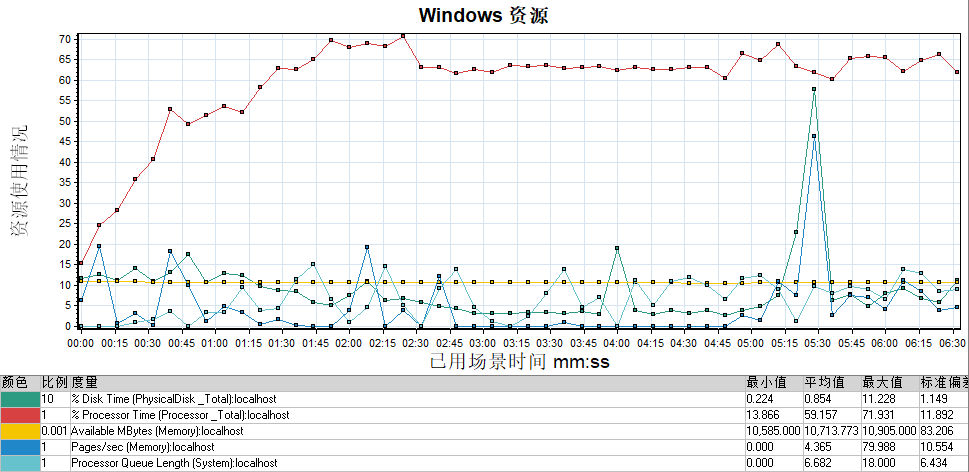
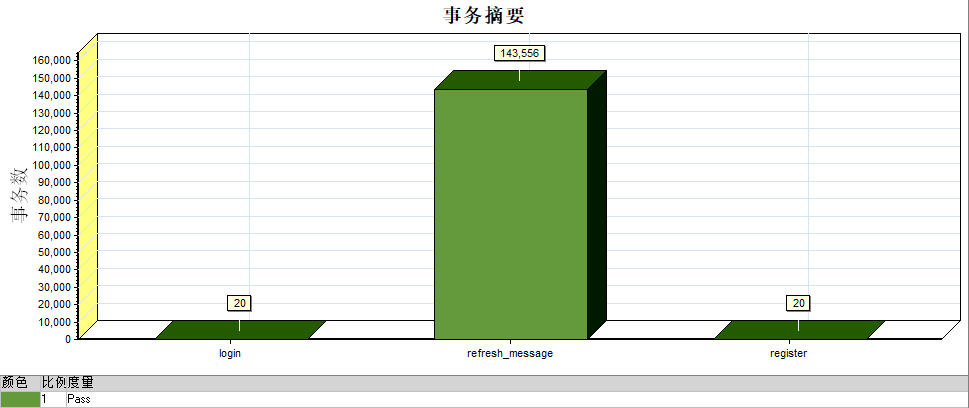
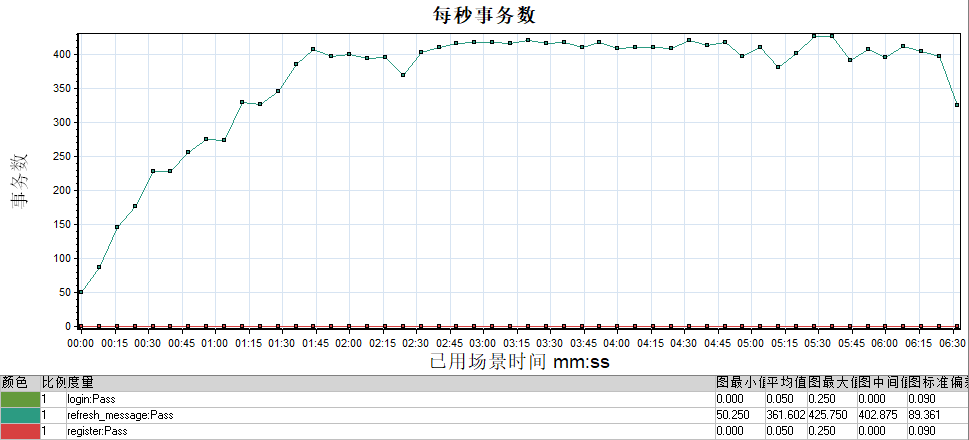
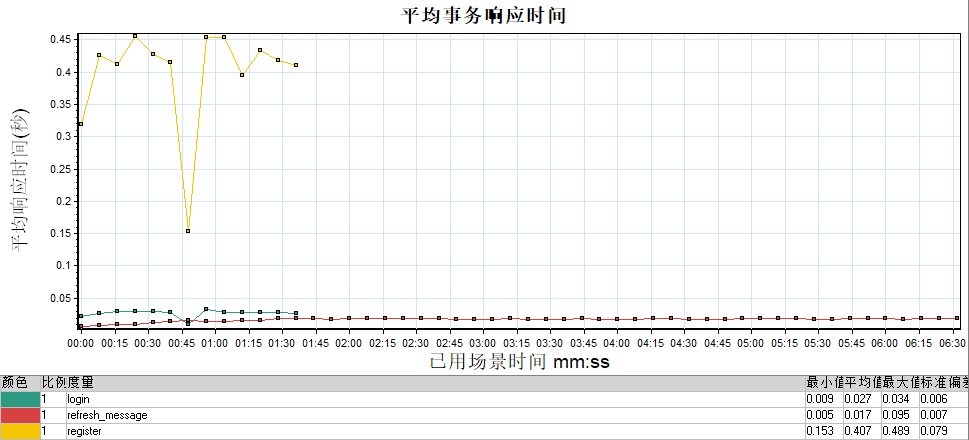
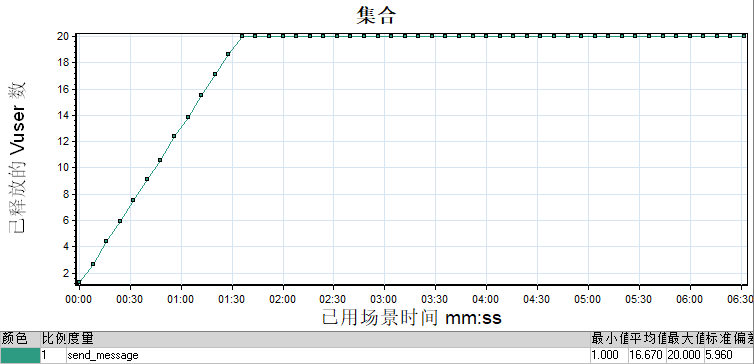
 

50并发用户数

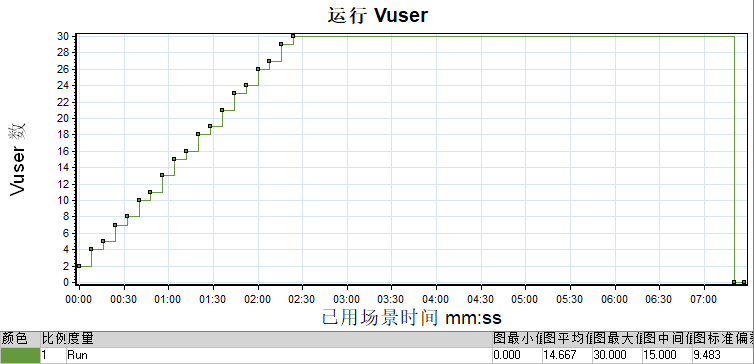
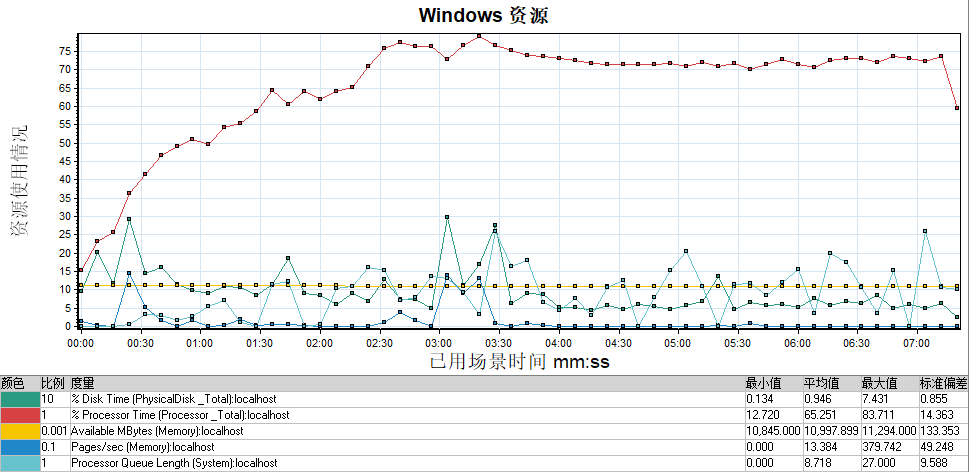
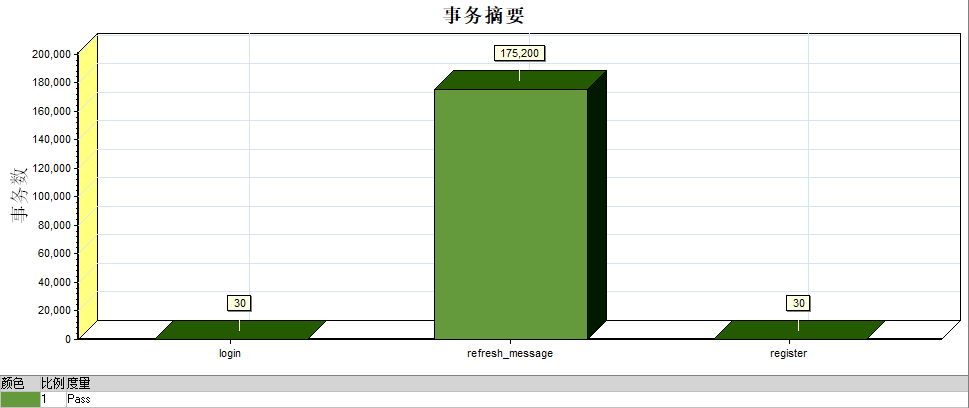
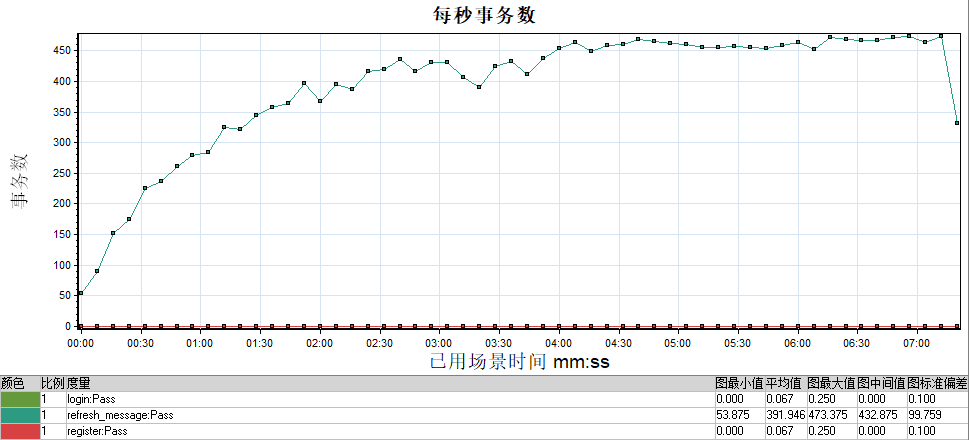
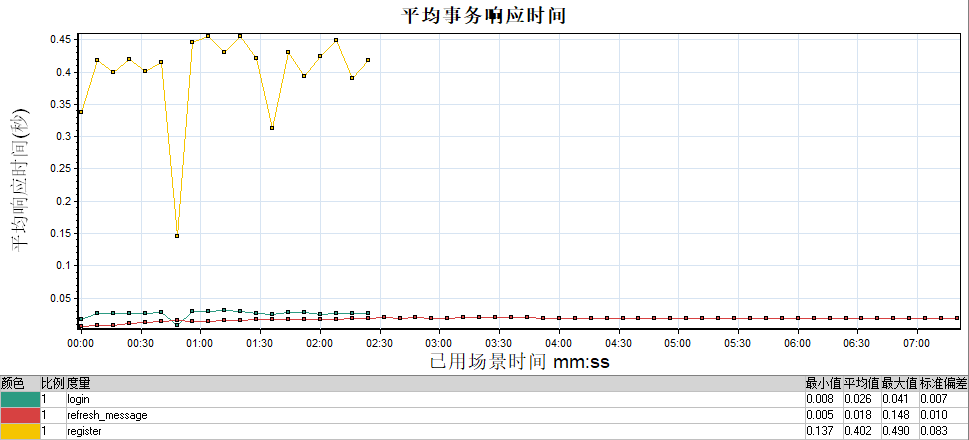
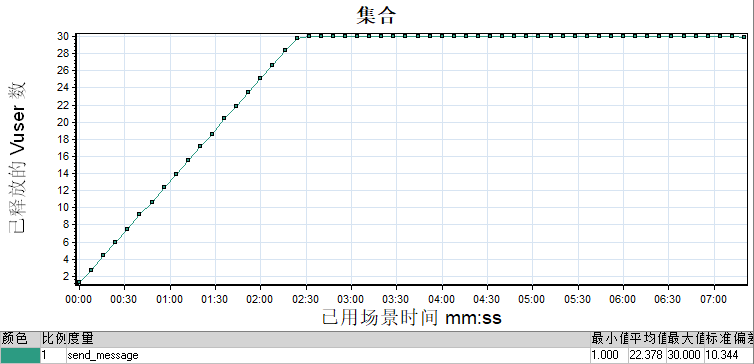
 

独立场景2：多个并发用户同时点击“更新“按钮更新消息

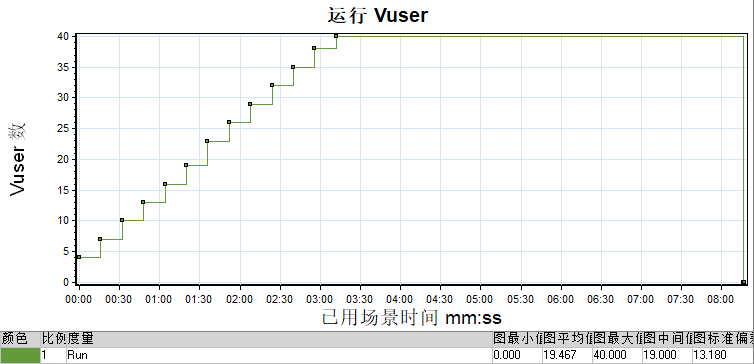
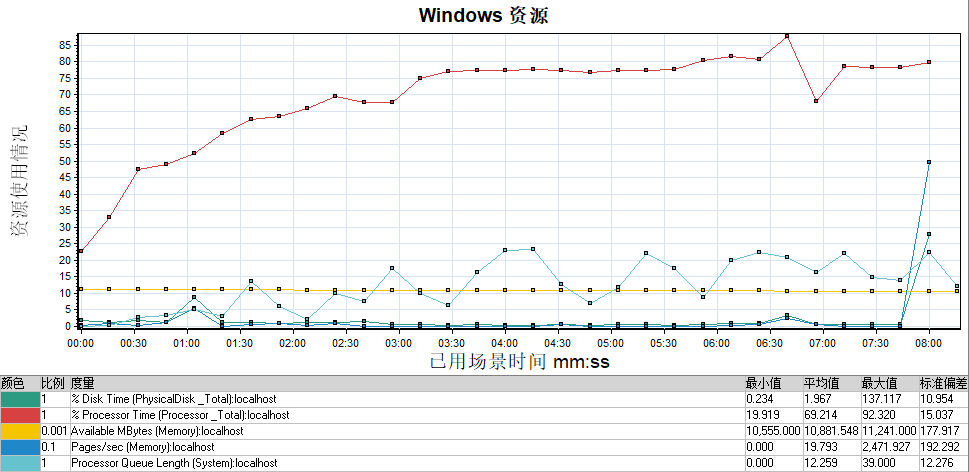
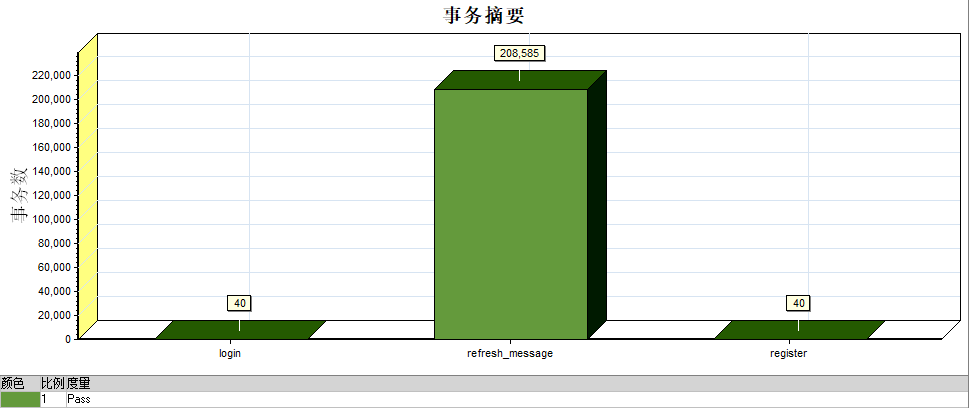
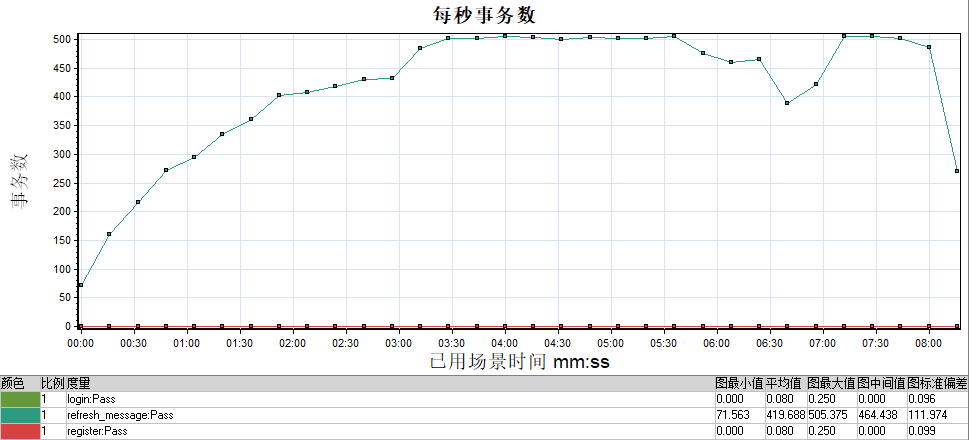
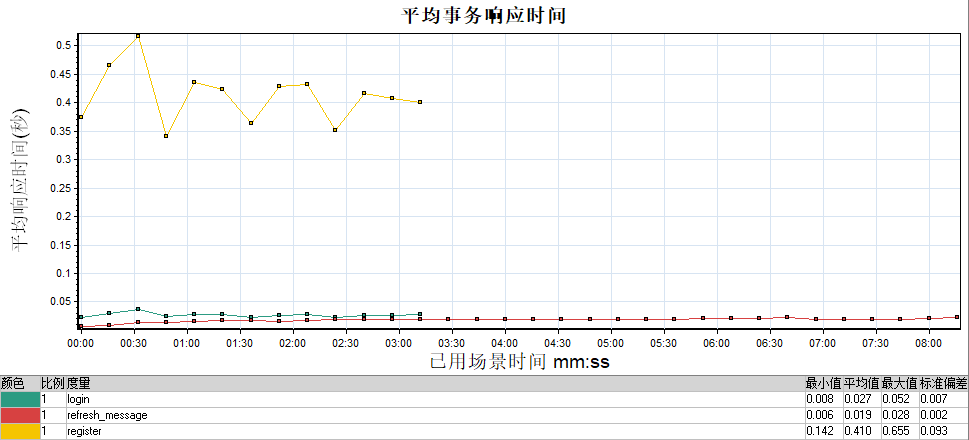
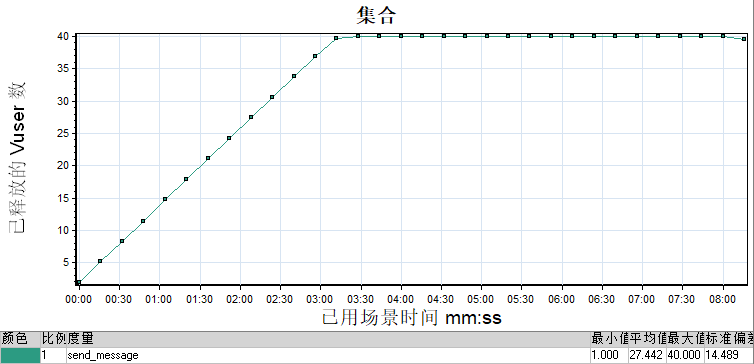
20并发用户数

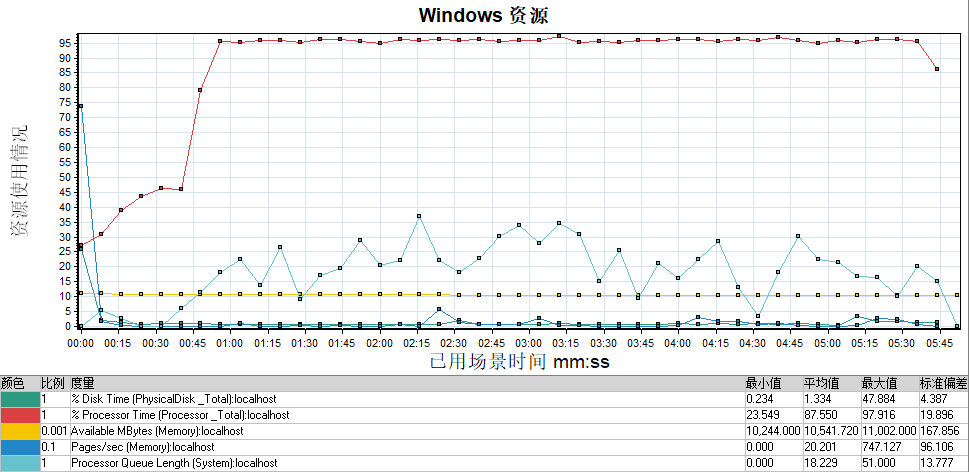
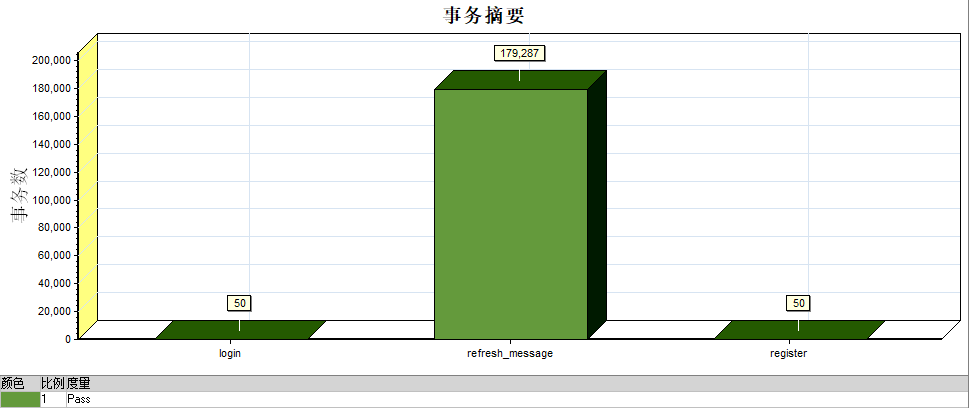
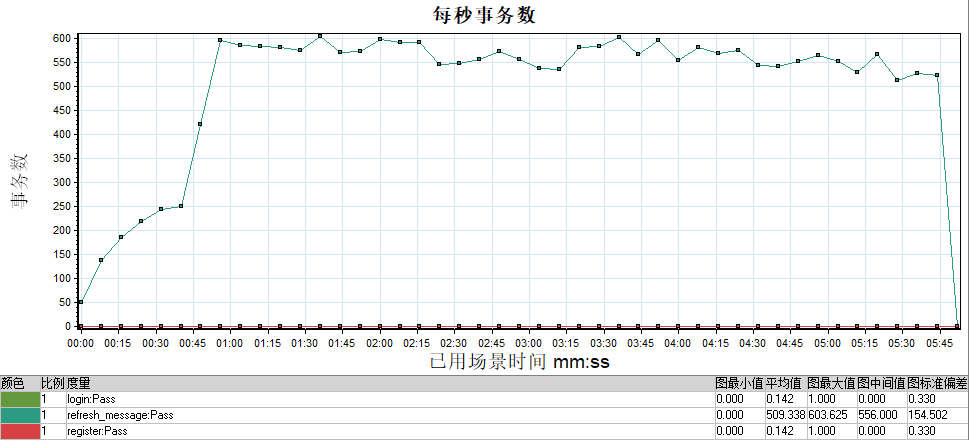
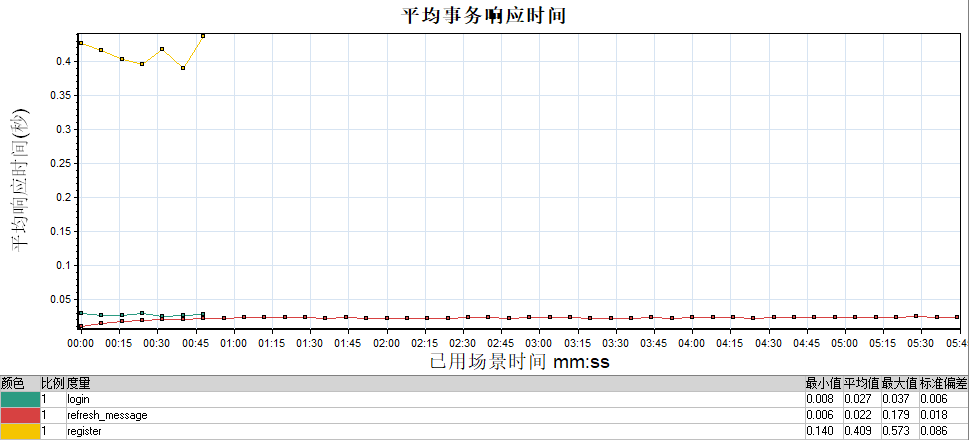
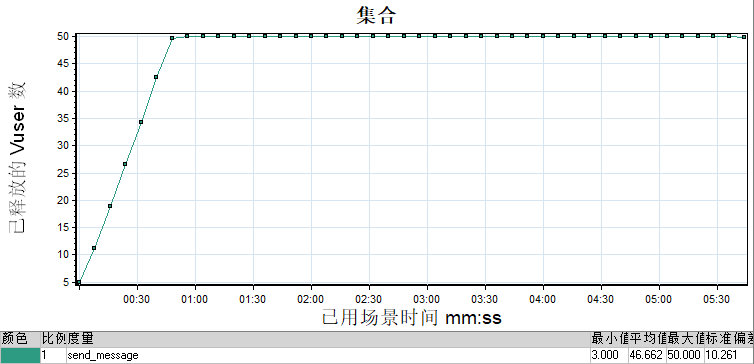
30并发用户数

40并发用户数

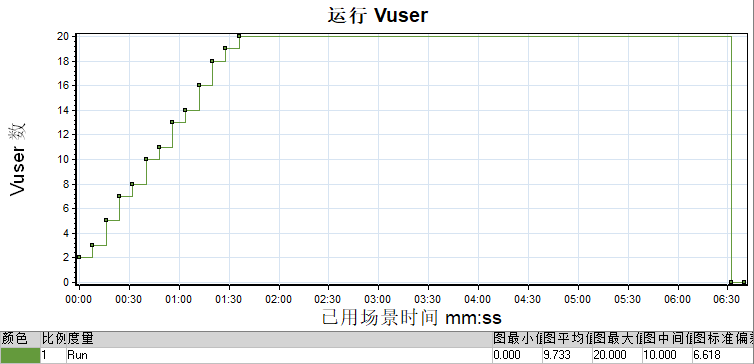
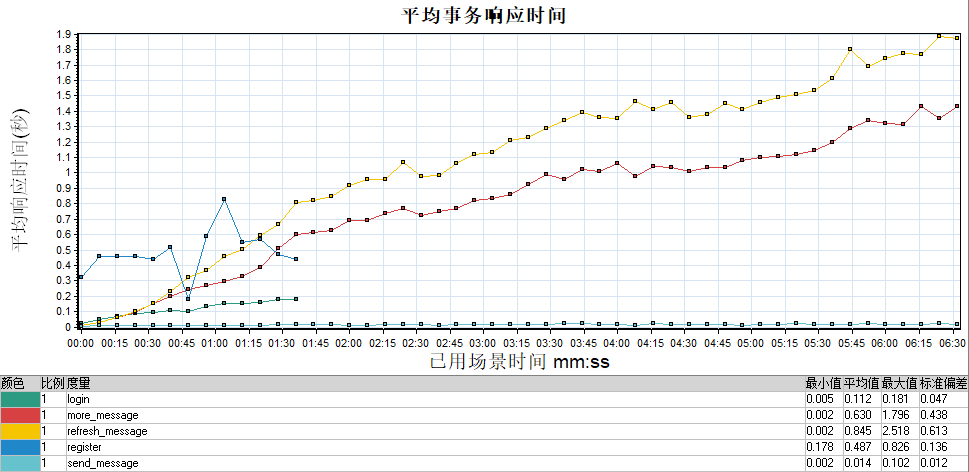
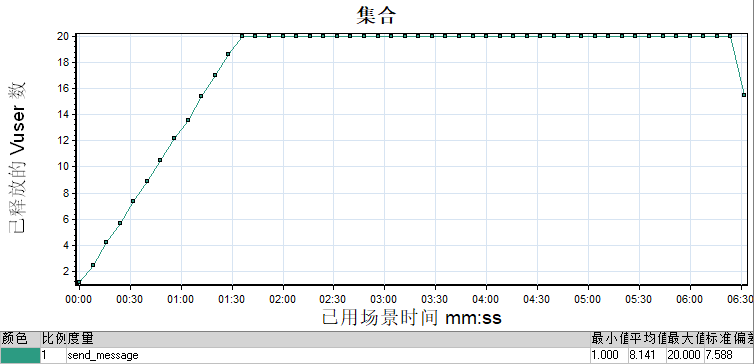
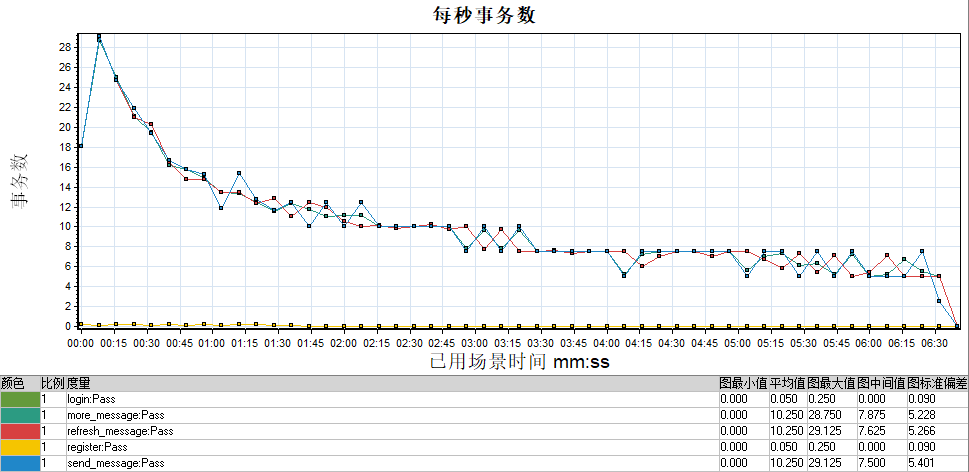
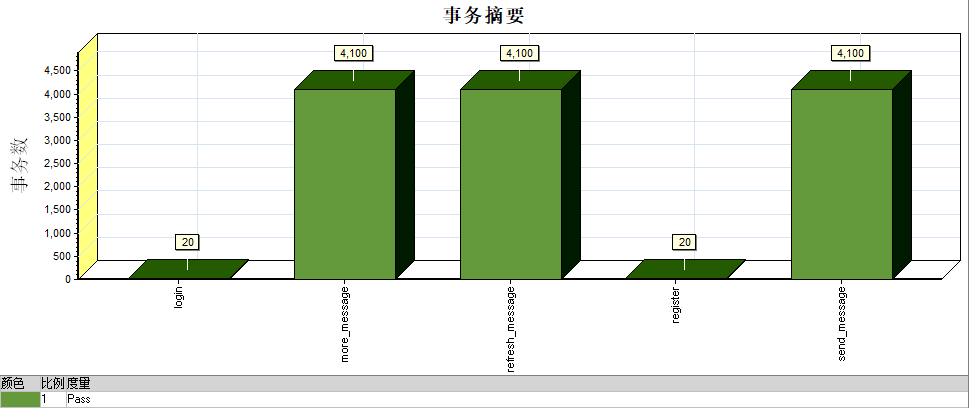
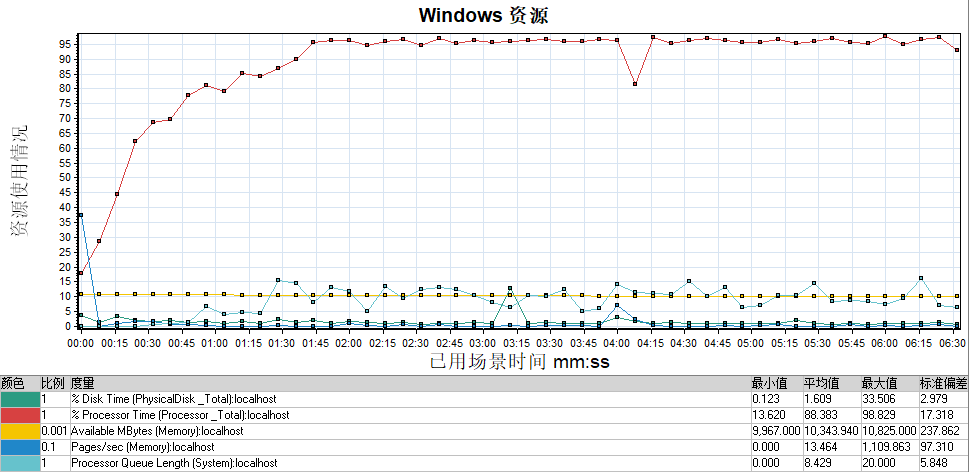
 

50并发用户数

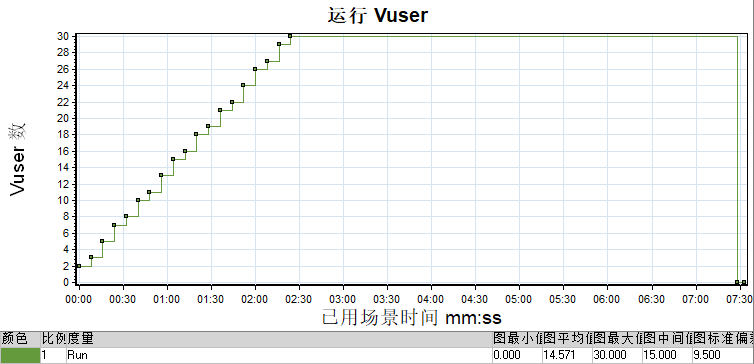
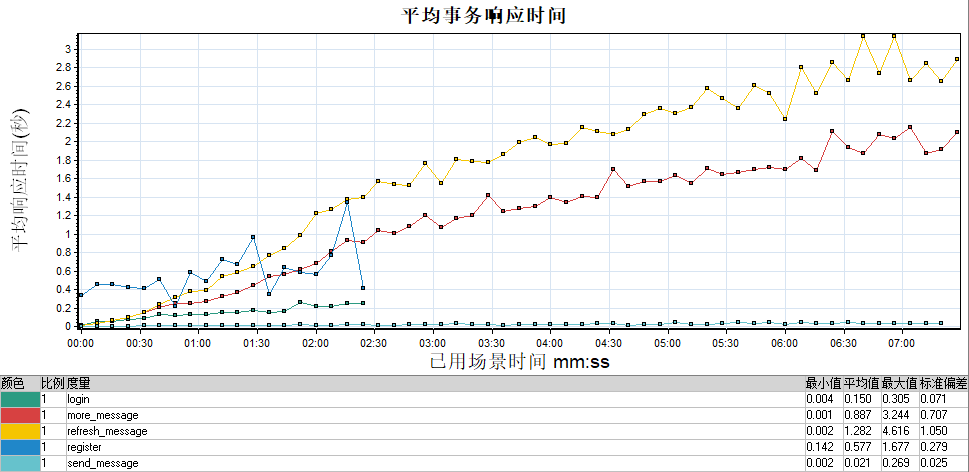
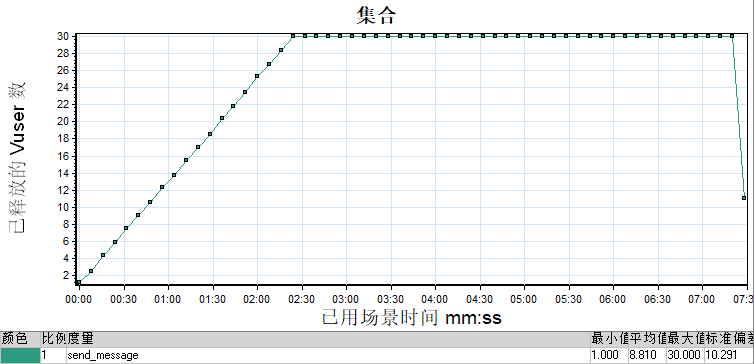
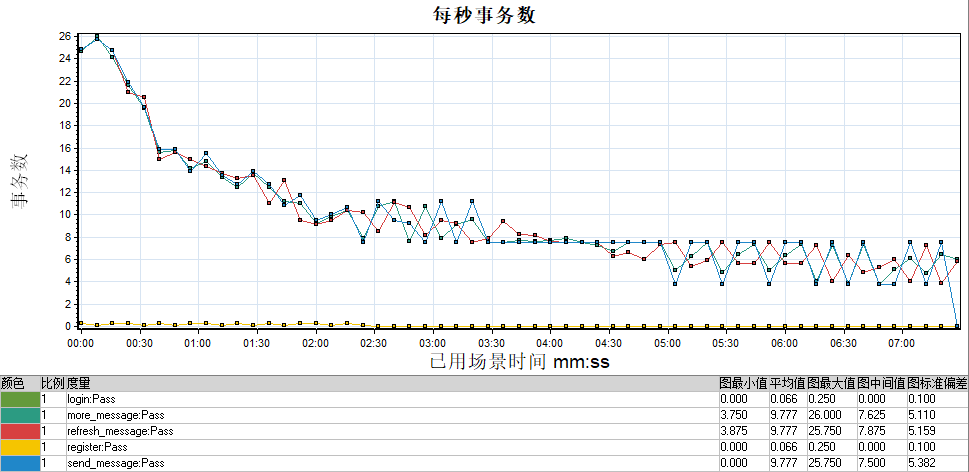
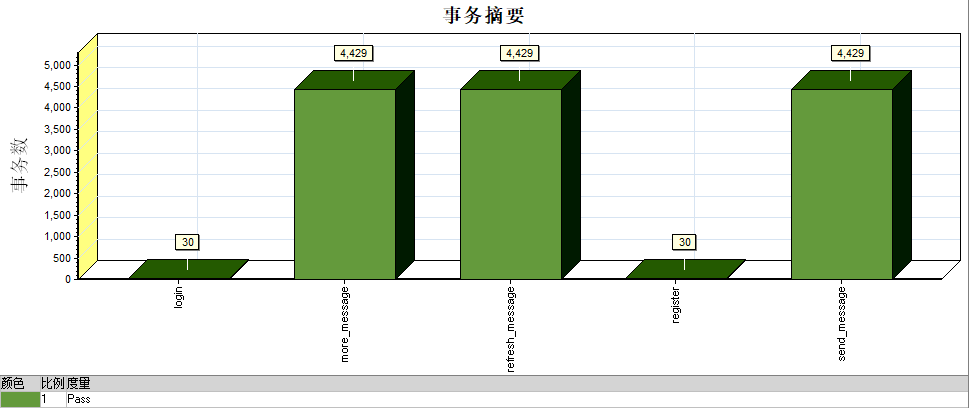
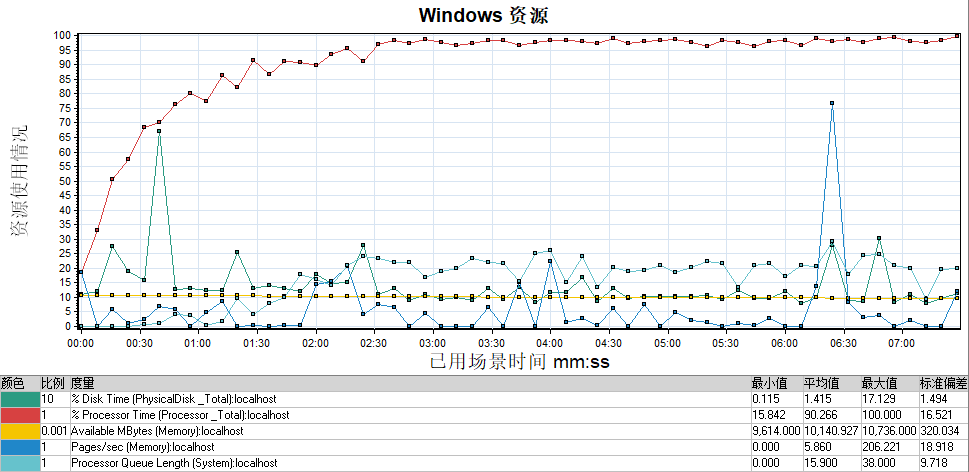
 

混合场景1：多个用户并发，每个用户登陆后，依次执行发布消息事务，更新消息事务，更多消息事务

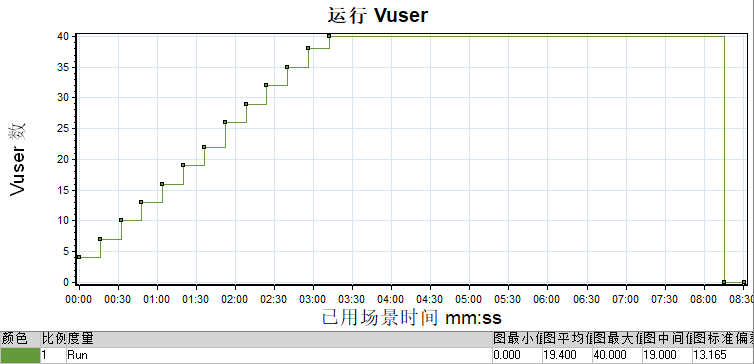
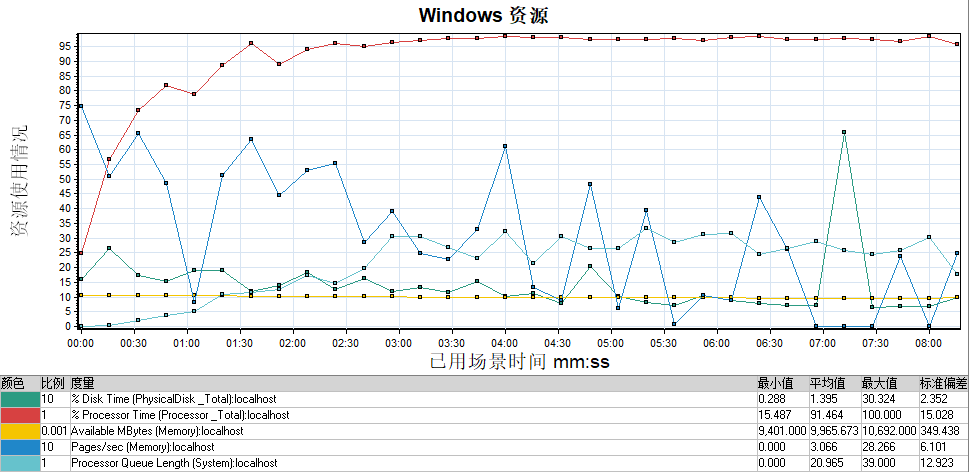
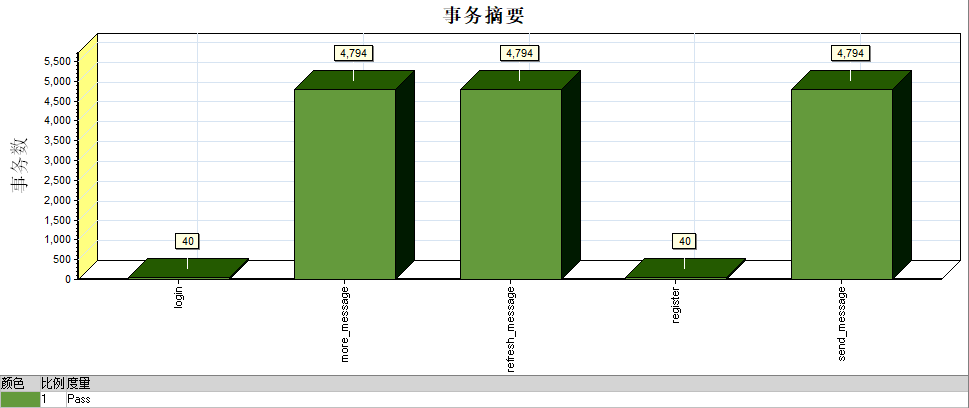
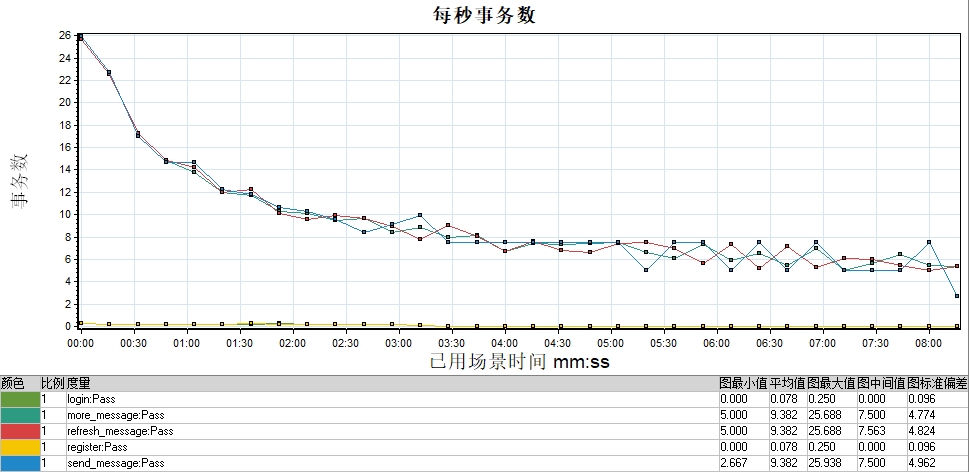
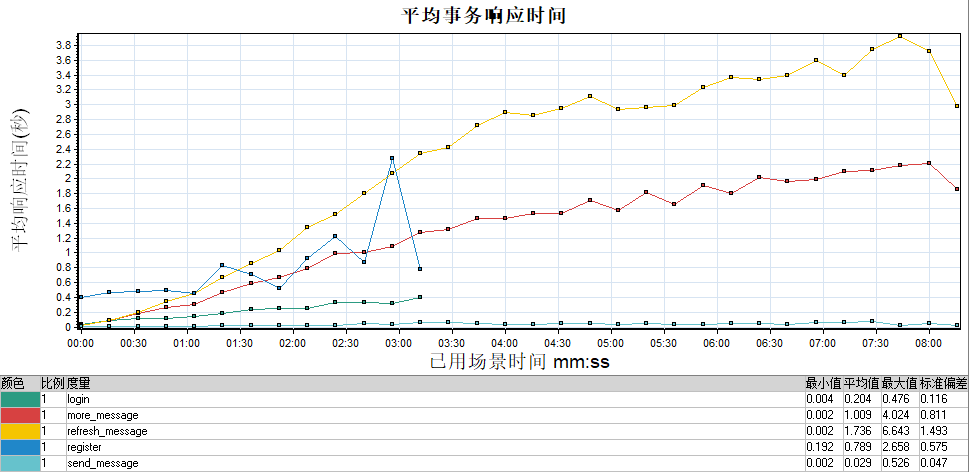
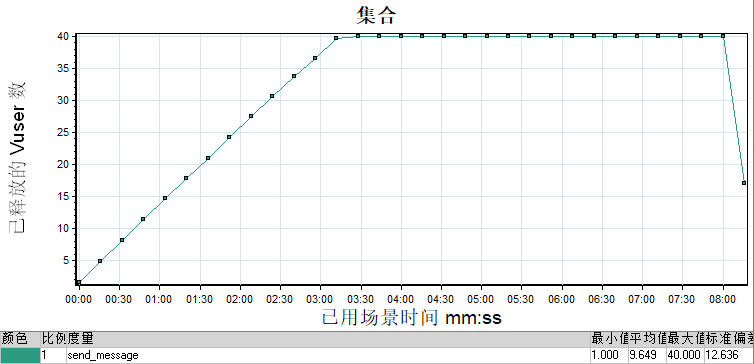
20并发用户数

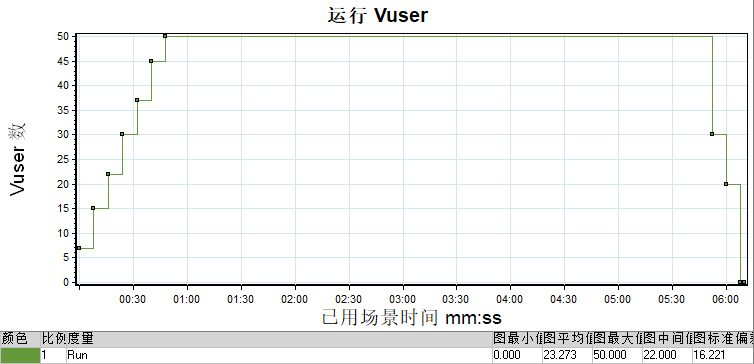
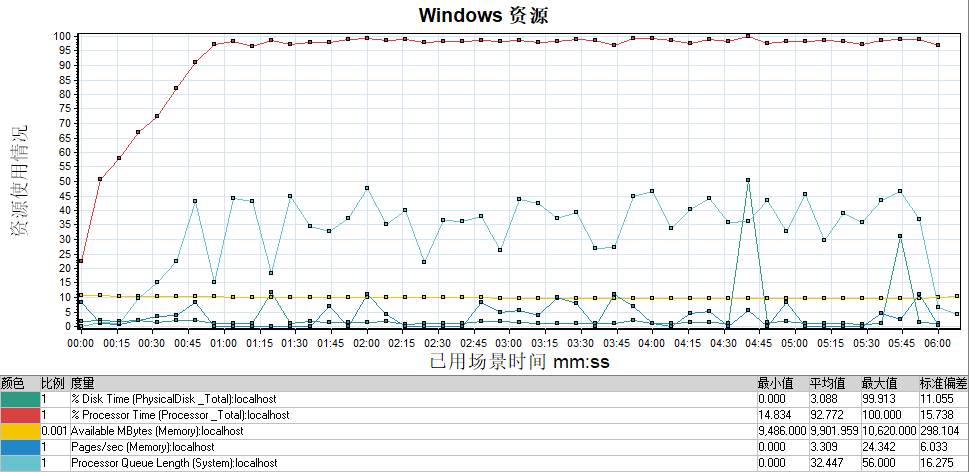
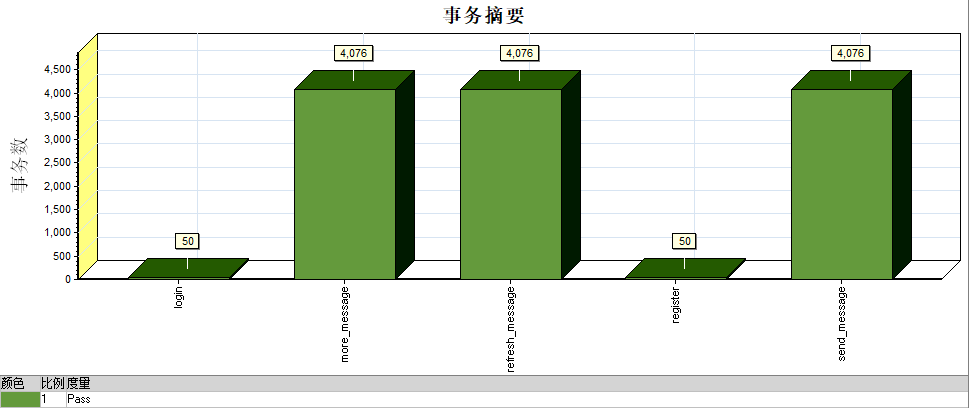
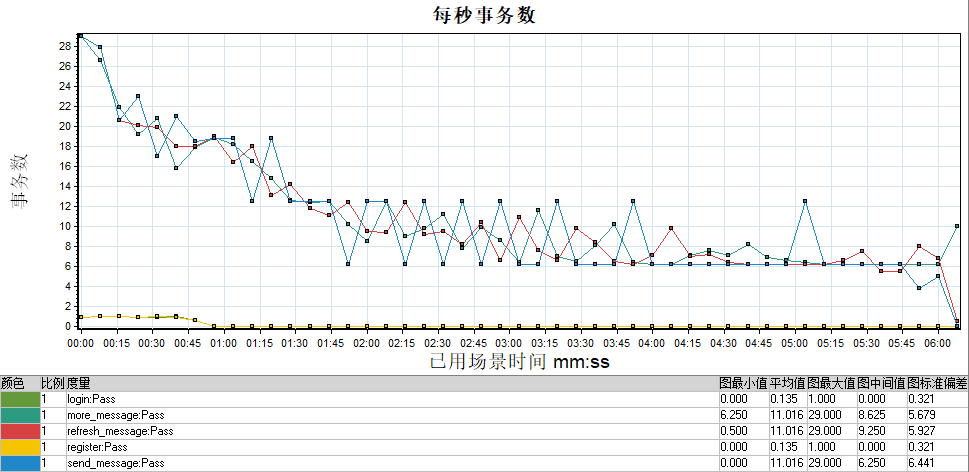
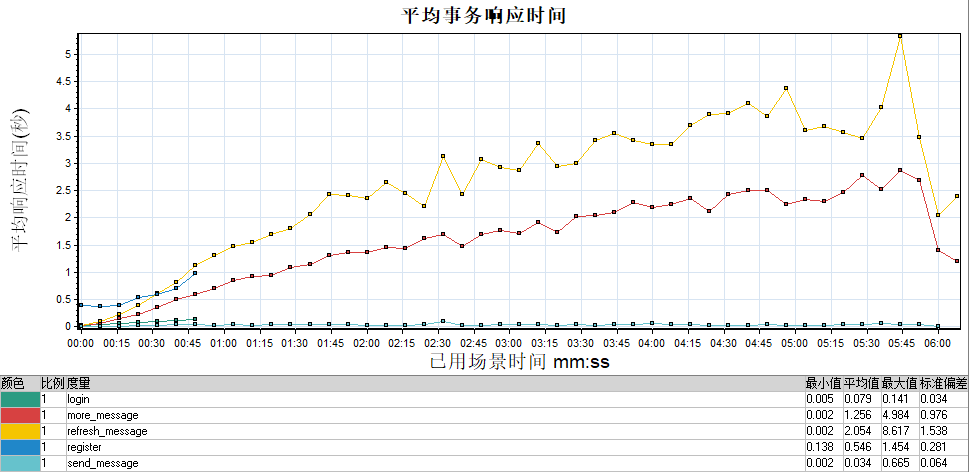
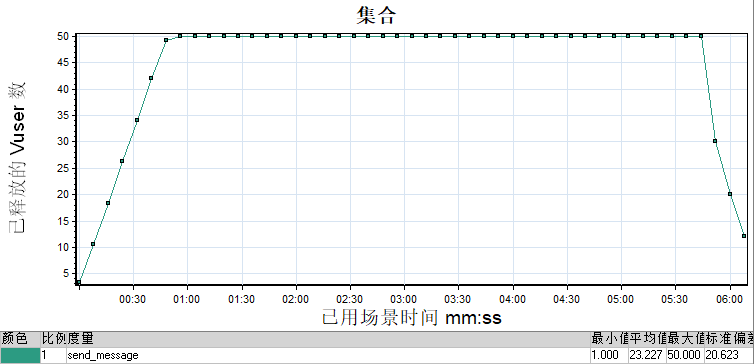
30并发用户数

40并发用户数

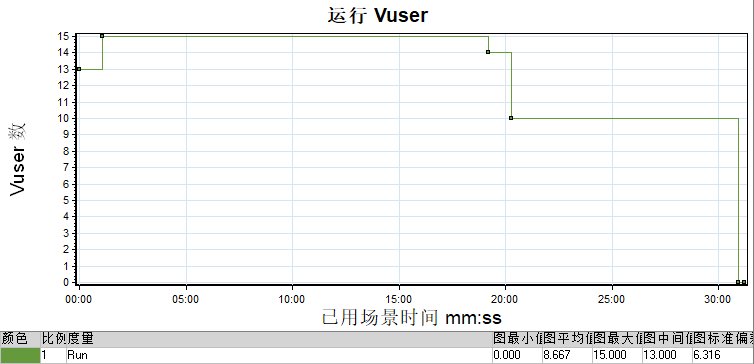
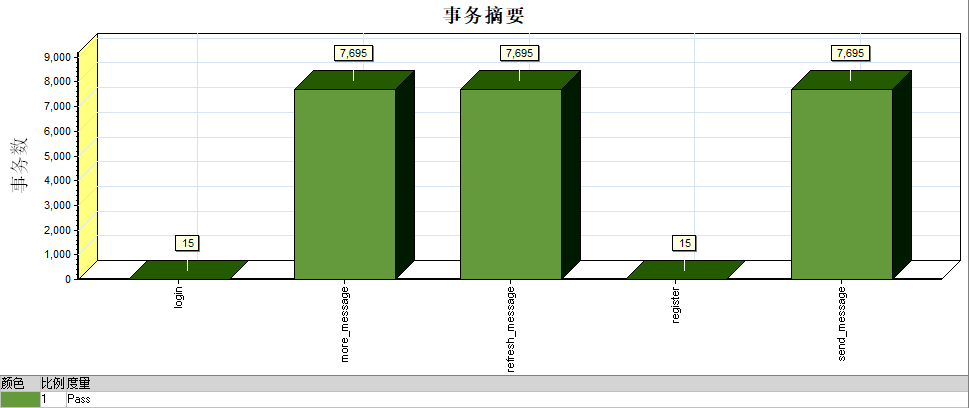
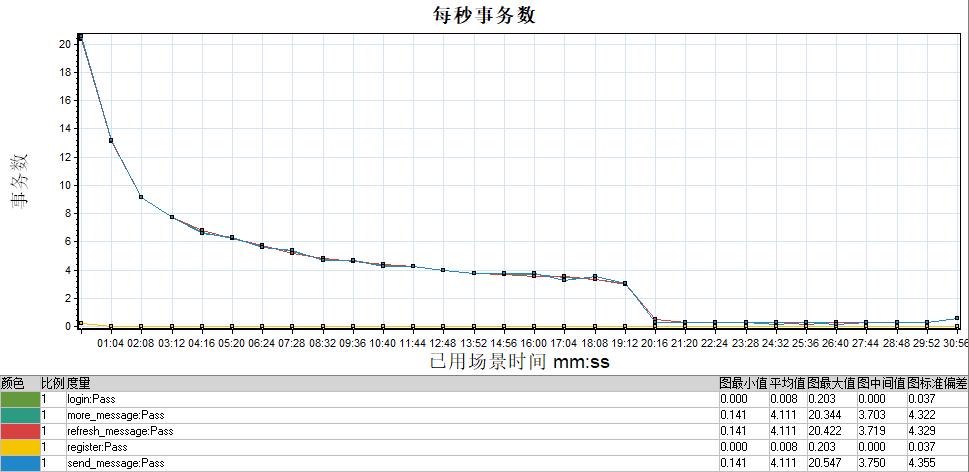
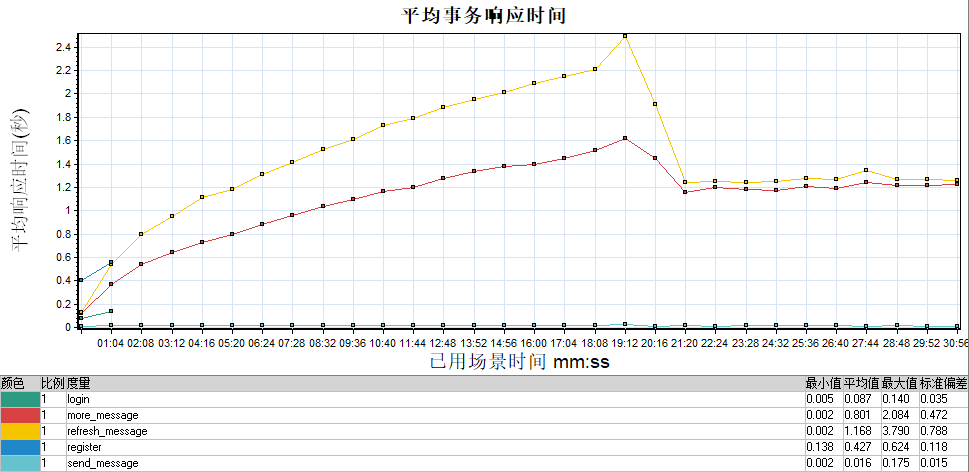
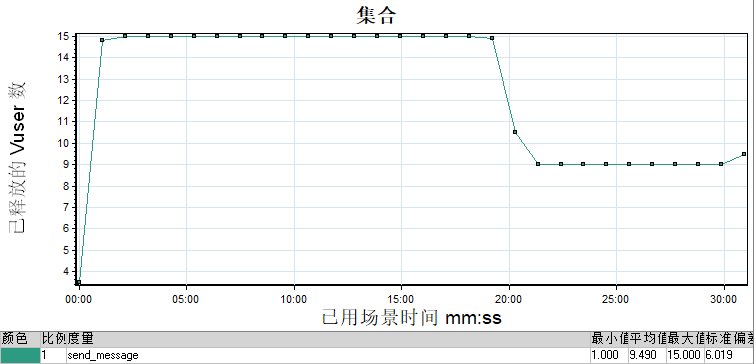
50并发用户数

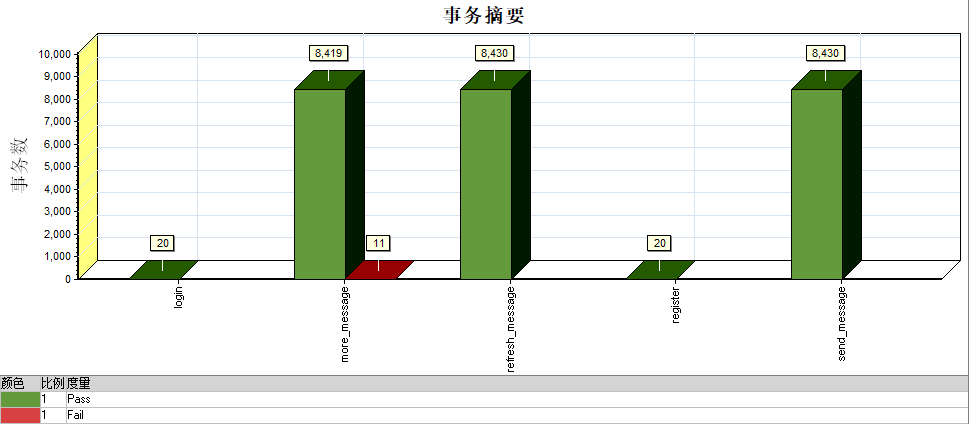
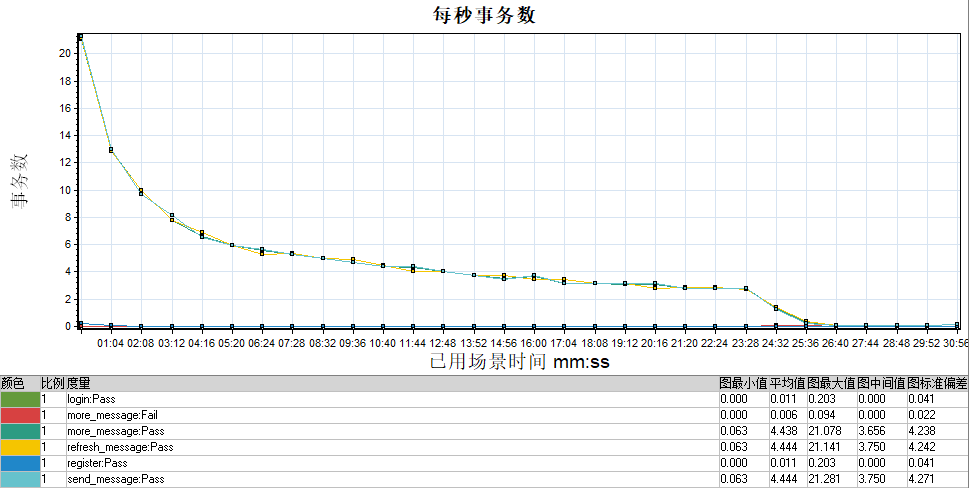
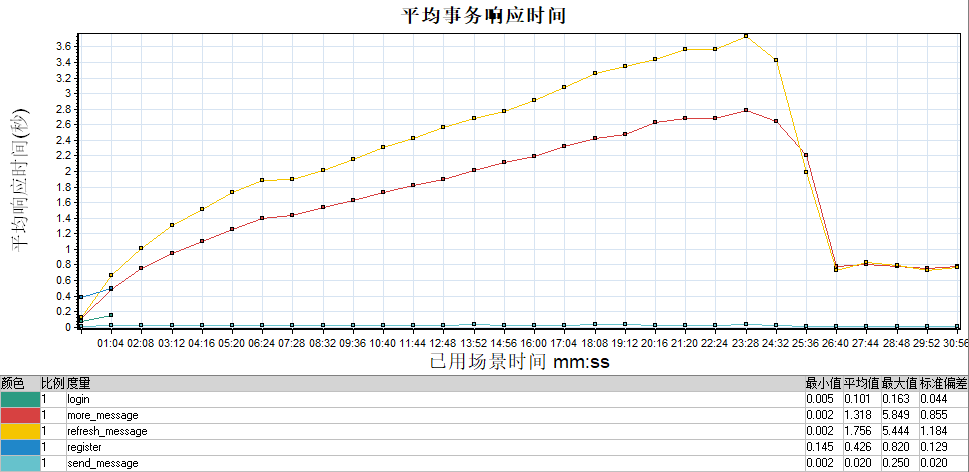
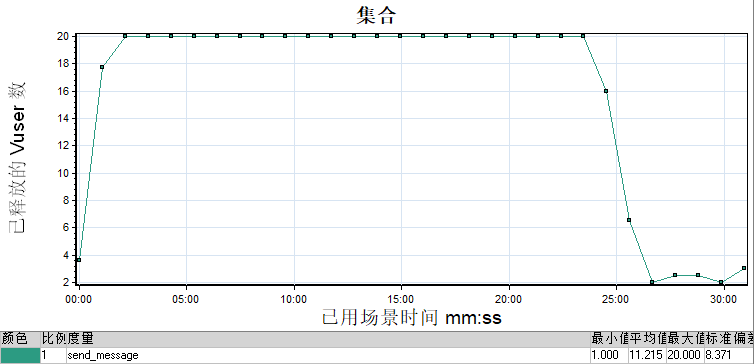
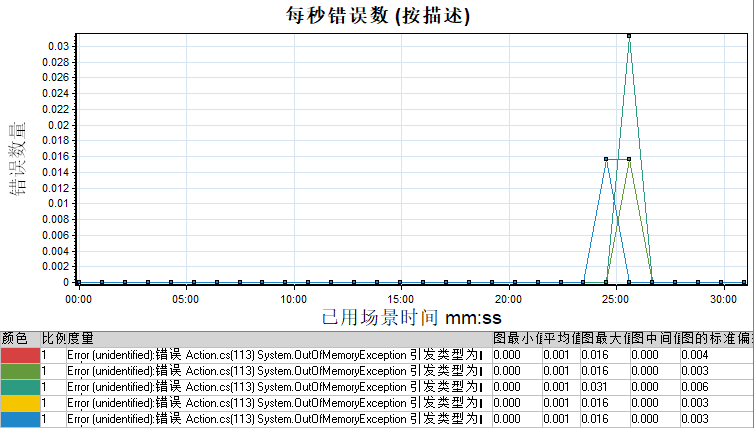
# 峰值测试场景报告

## 测试结果摘要

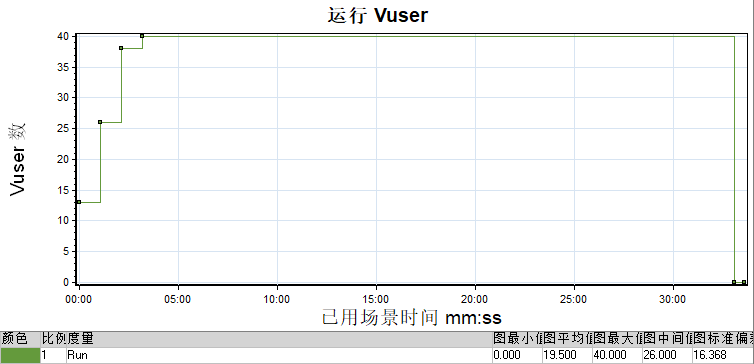
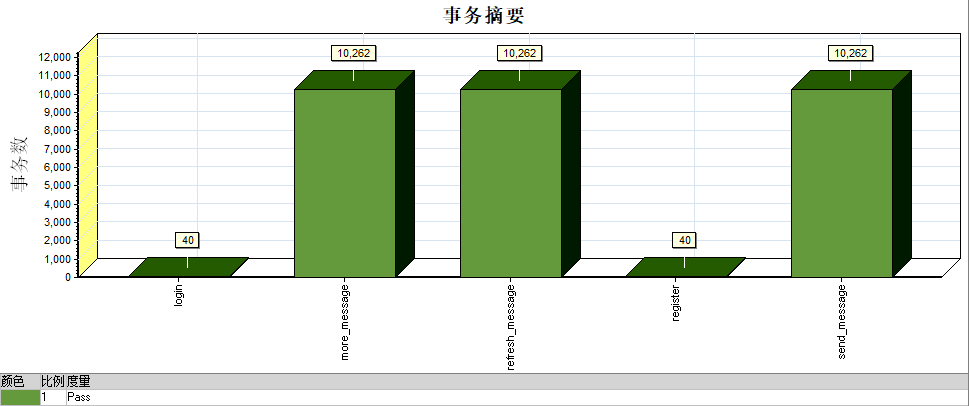
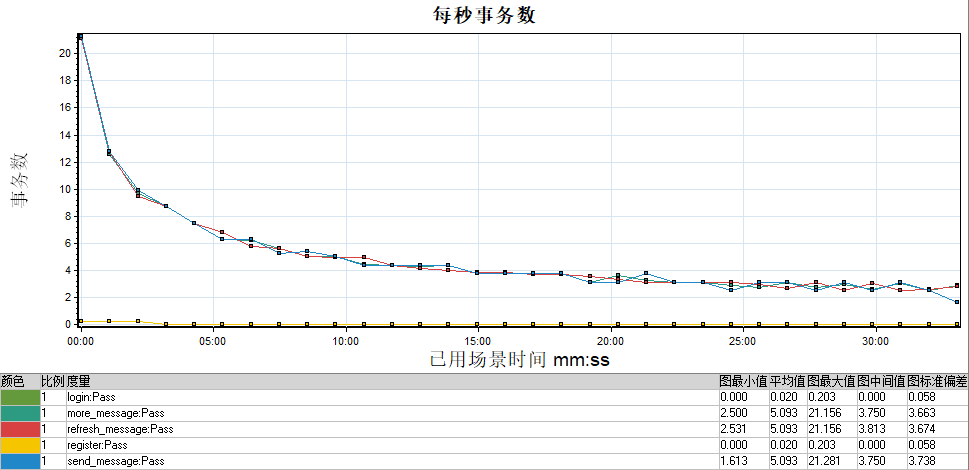
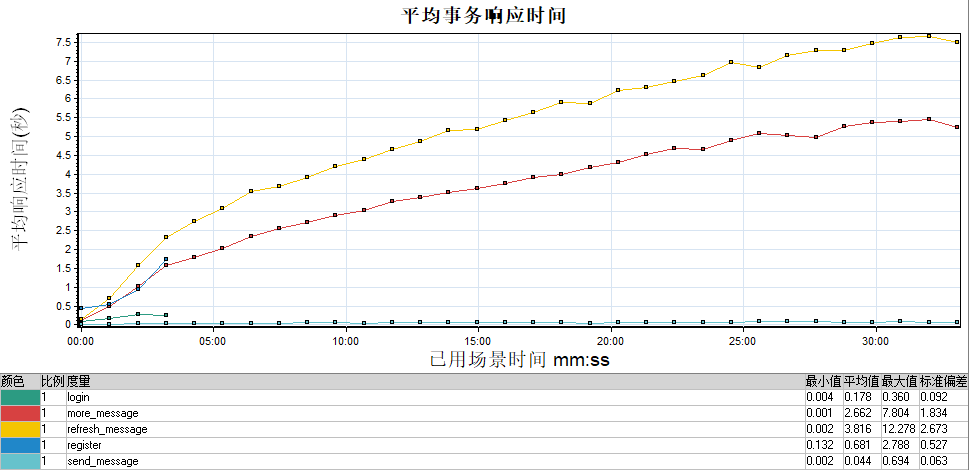
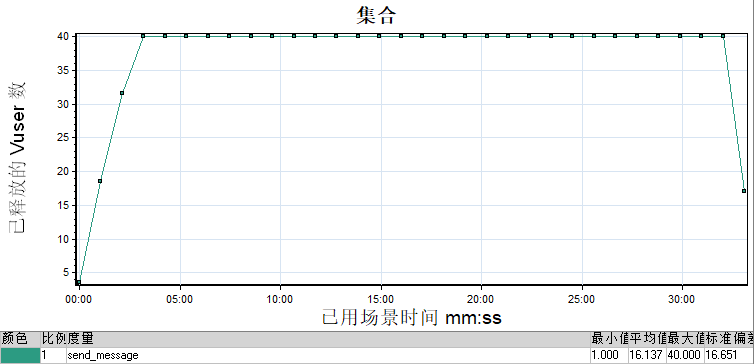
15并发用户数(生产压力的70%)

20并发用户数(生产压力)

40并发用户数(生产压力的2倍)

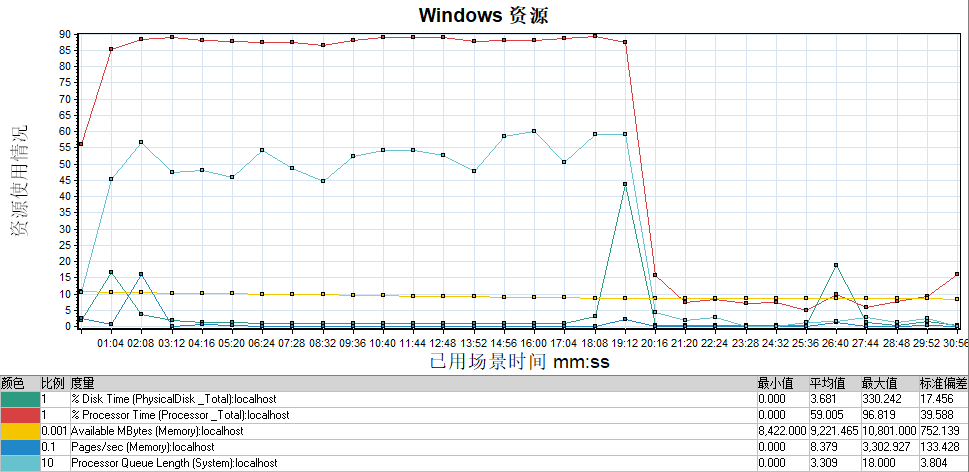
## 本次测试结论

峰值场景下模拟的生产压力为20并发用户，加压时间30分钟。此时，事务平均响应时间为3.904秒，最大响应时间11.543秒，基本满足响应时间2/5/10秒的电商规则。业务成功率为99.95%，cpu平均利用率75.855%，满足要求。

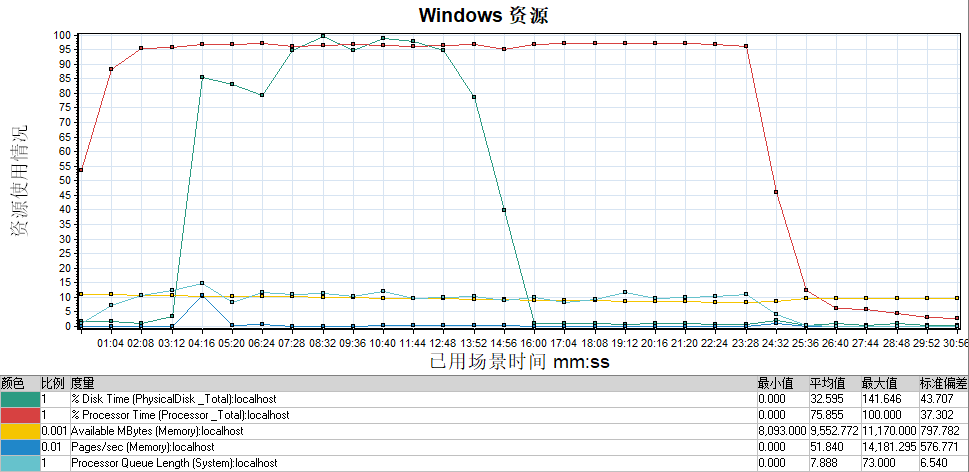
在2倍的生产压力（40并发用户）下，加压时间30分钟，事务平均响应时间为6.522秒，最大响应时间20.776秒，业务成功率为100%，cpu平均利用率96.390%。业务成功率满足要求，但事务平均响应时间和cpu利用率不理想。需要继续优化性能，以满足未来的业务发展需求。

## 系统性能监控

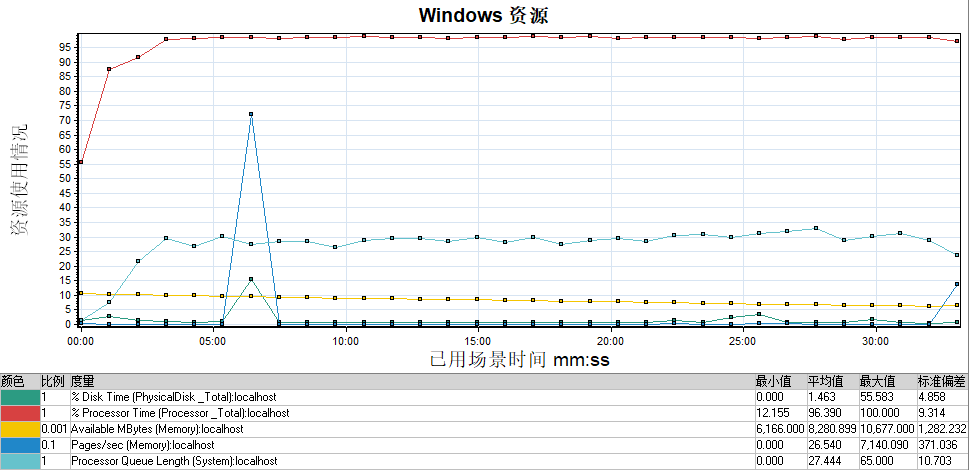
15并发用户数(生产压力的70%)



20并发用户数(生产压力)



40并发用户数(生产压力的2倍)



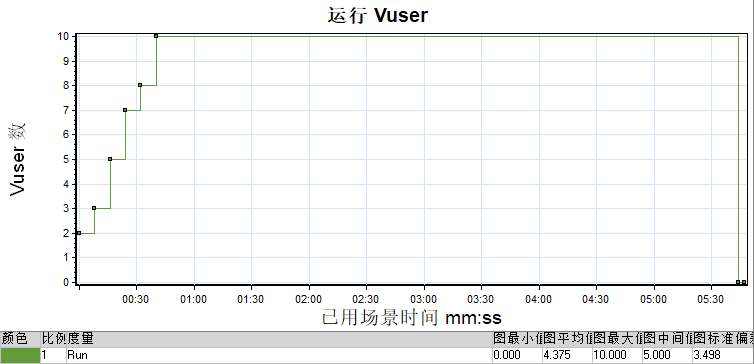
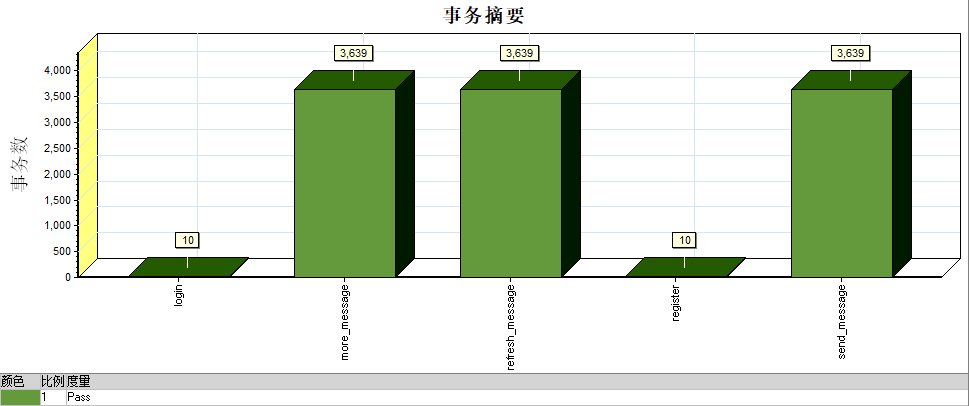
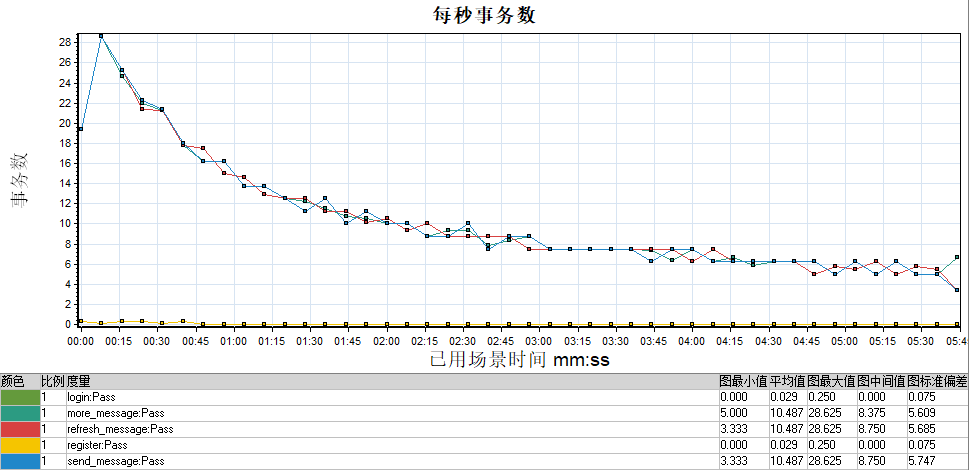
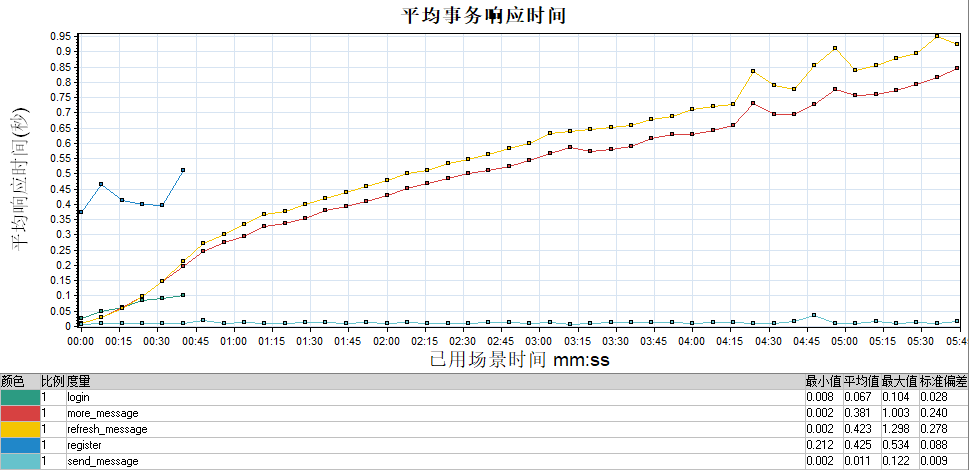
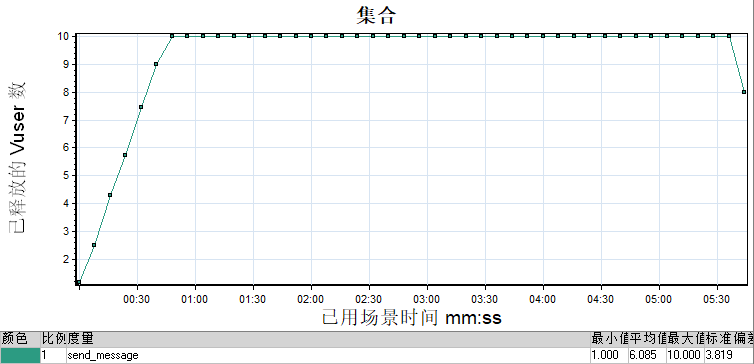
## 事务出错分析

错误类型为System.OutOfMemoryException，为.Net内存溢出。可能需要优化桌面端.Net项目的内存使用。

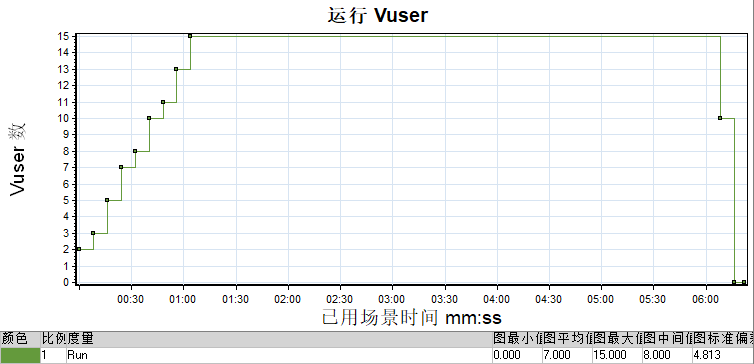
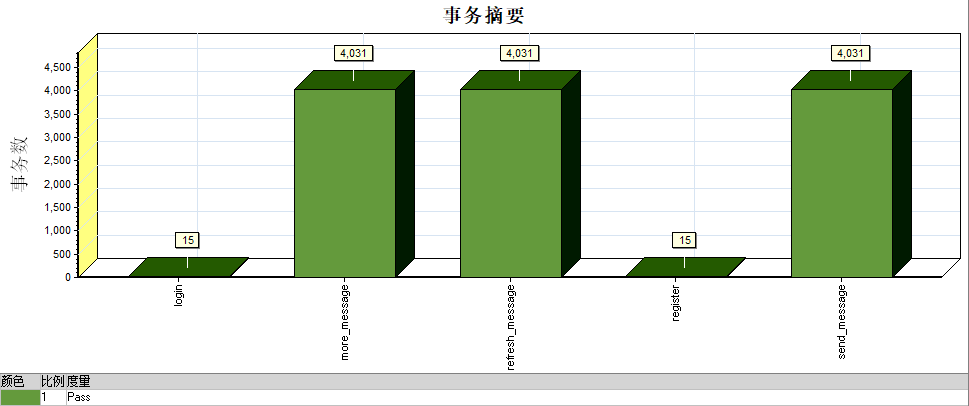
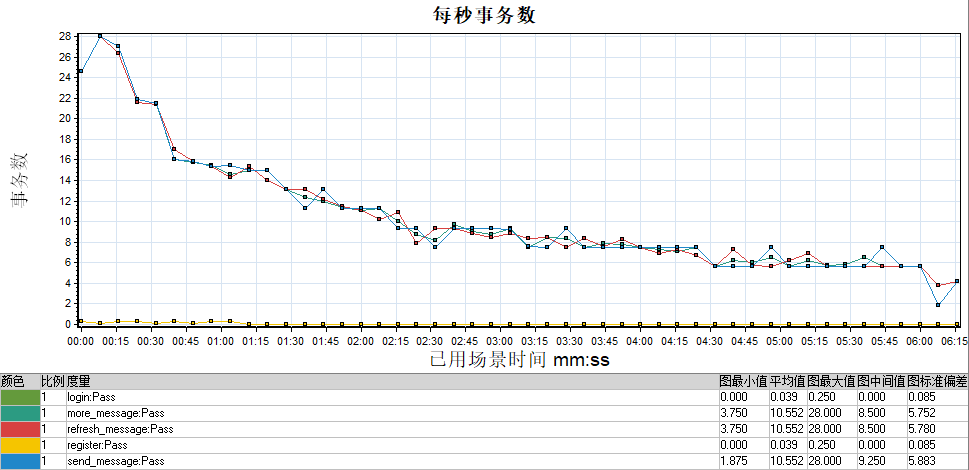
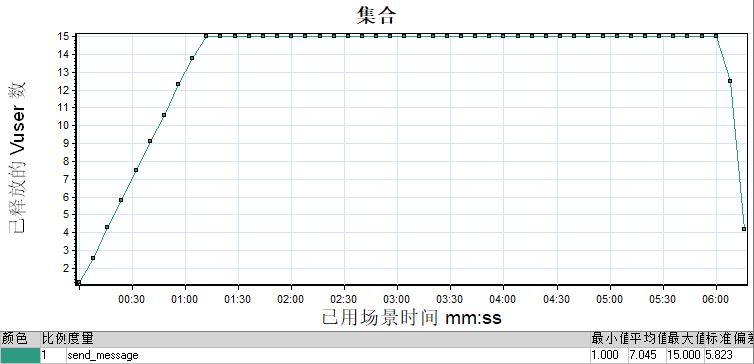
# 容量场景测试报告

## 测试结果摘要

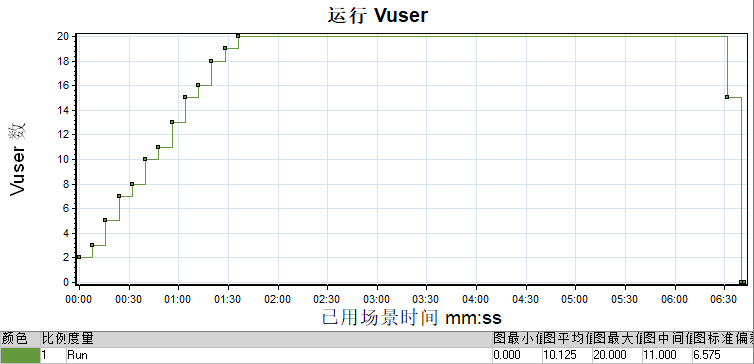
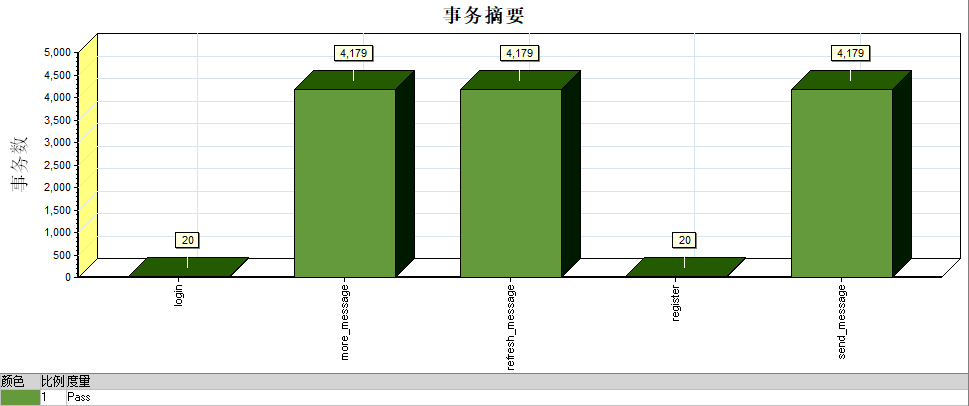
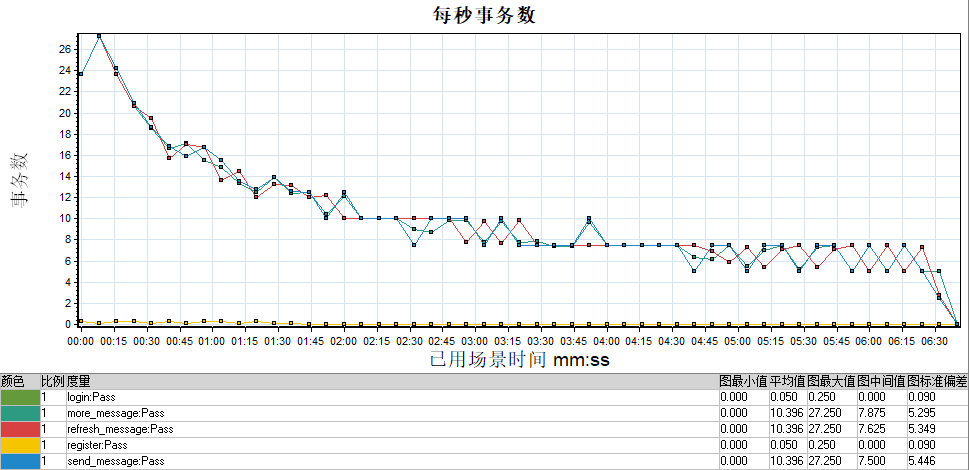
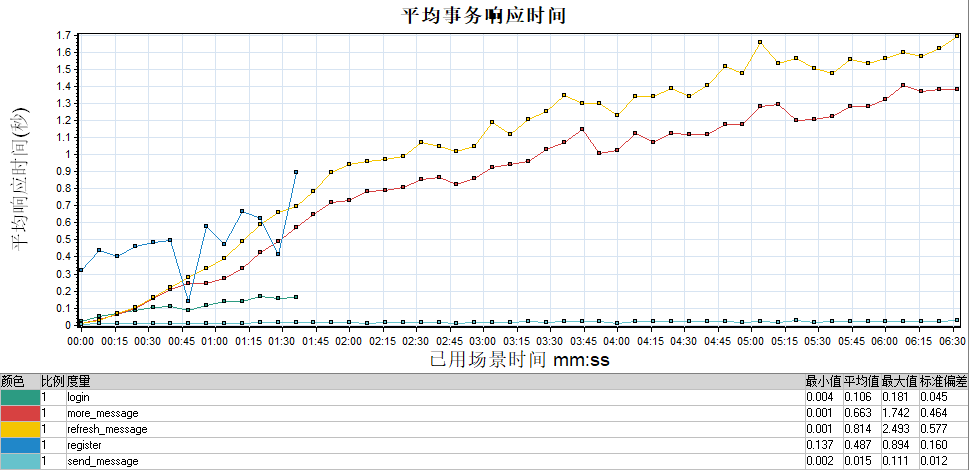
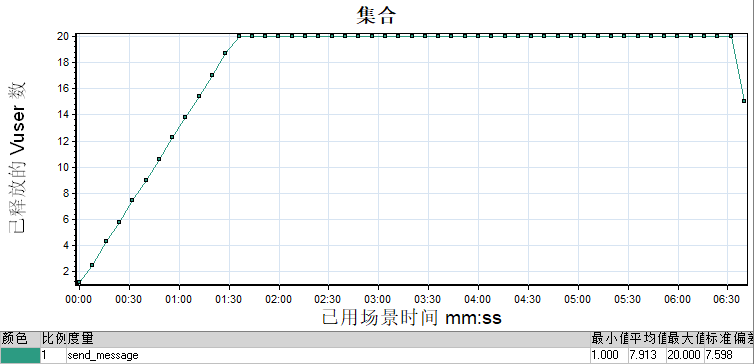
10并发用户数

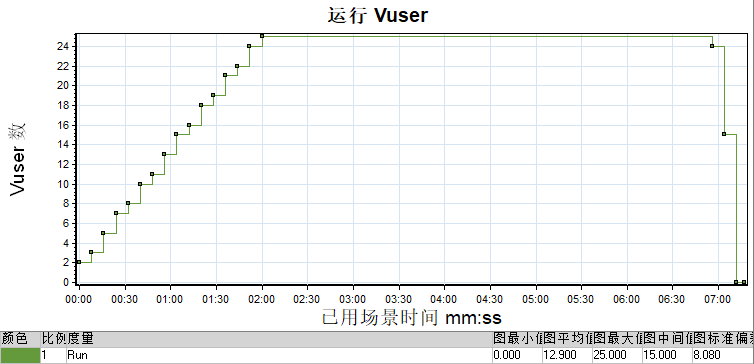
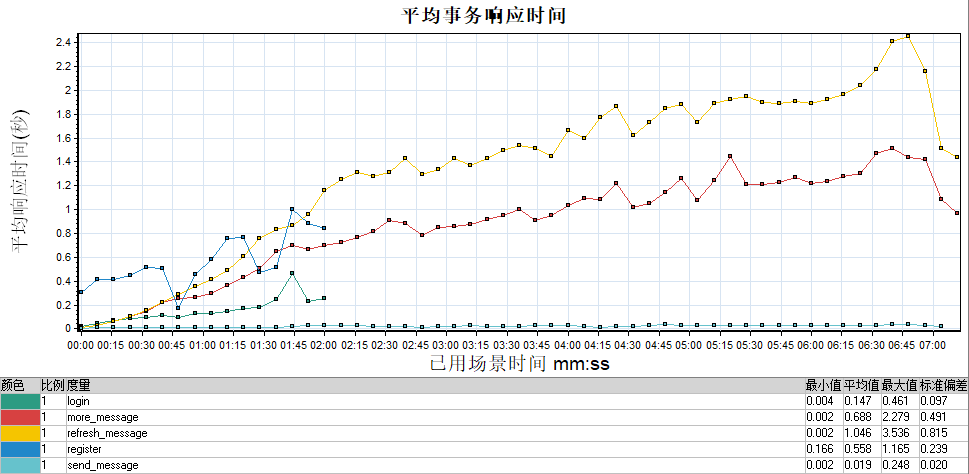
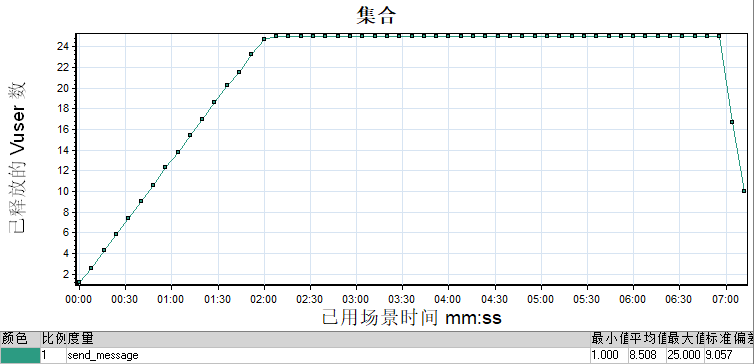
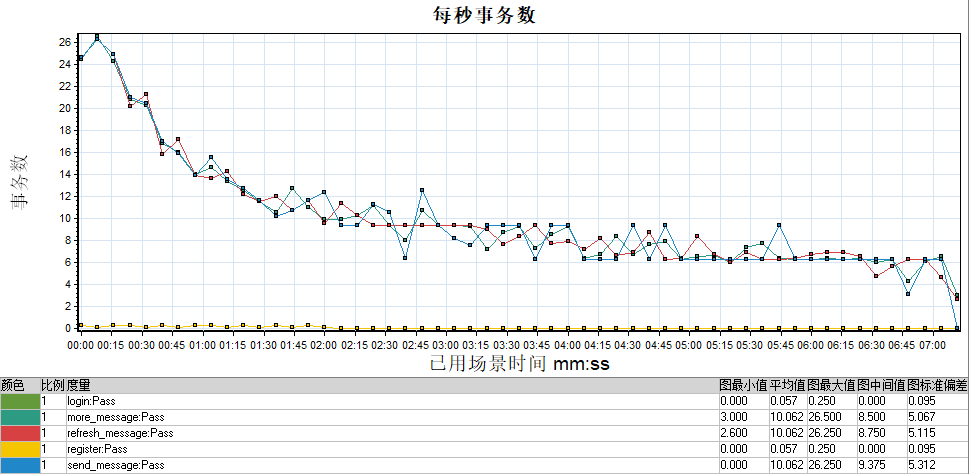
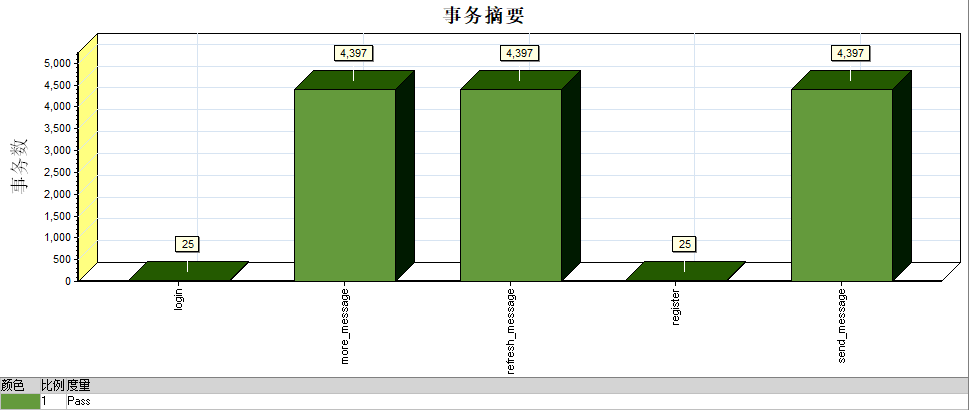
15并发用户数

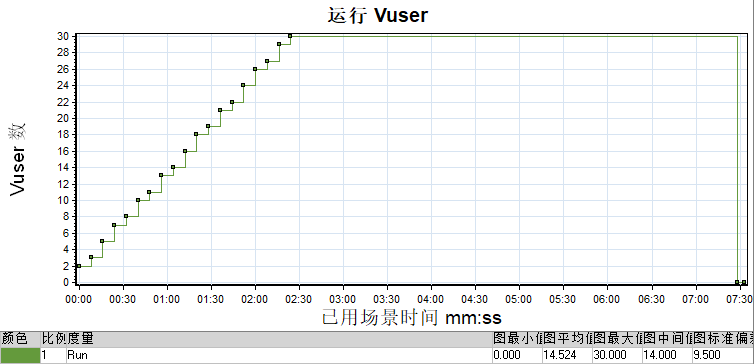
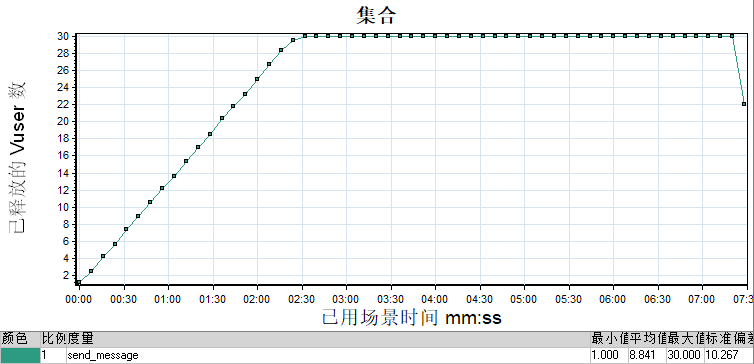
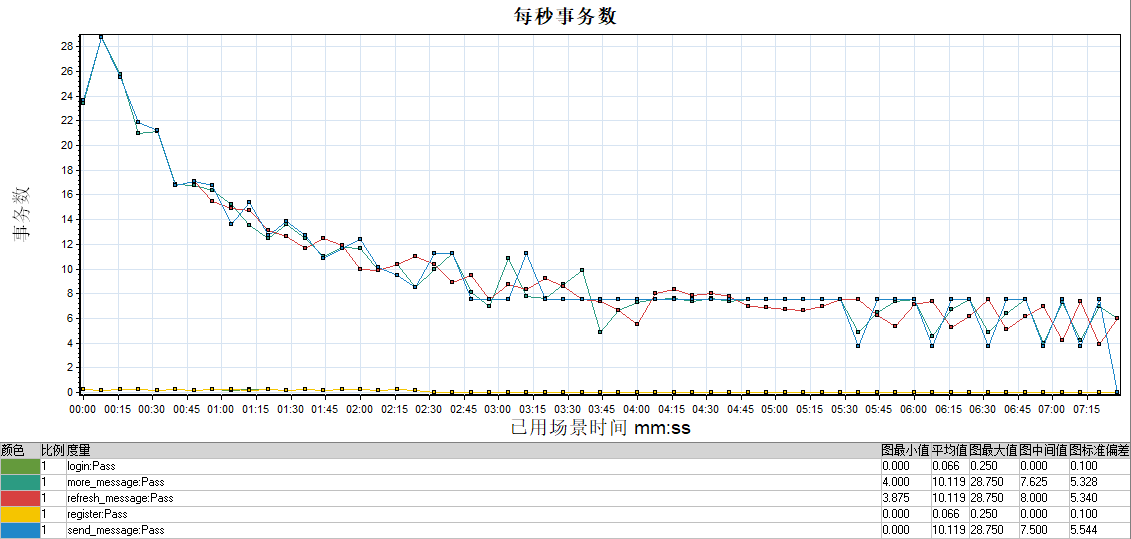
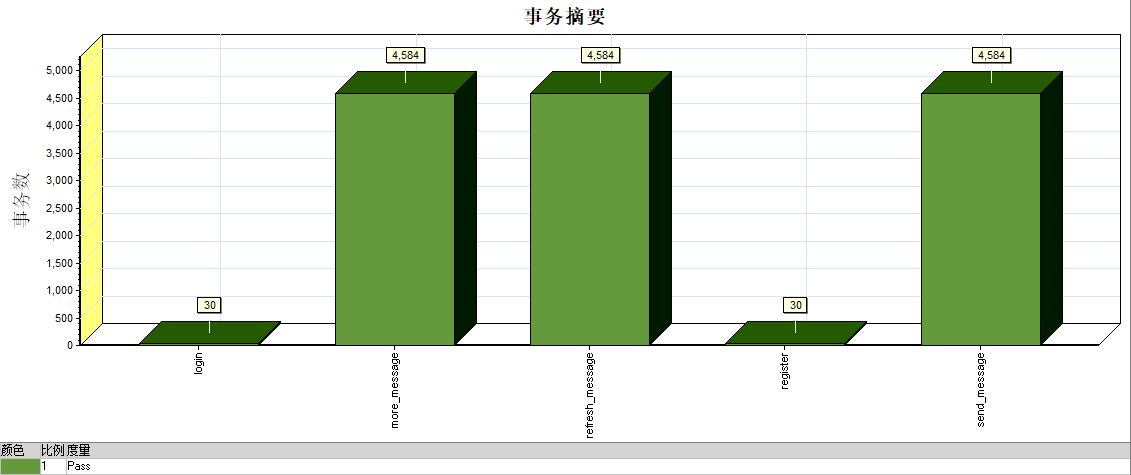
20并发用户数

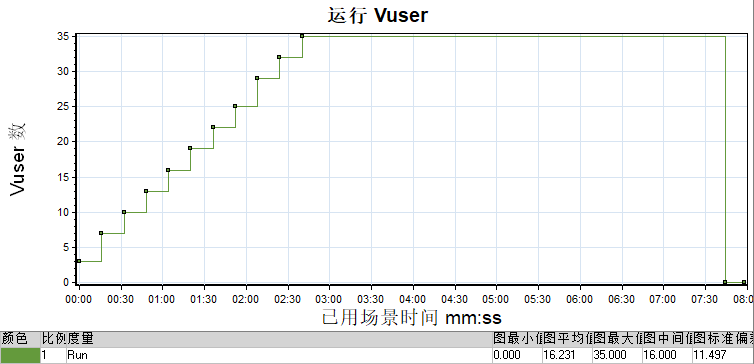
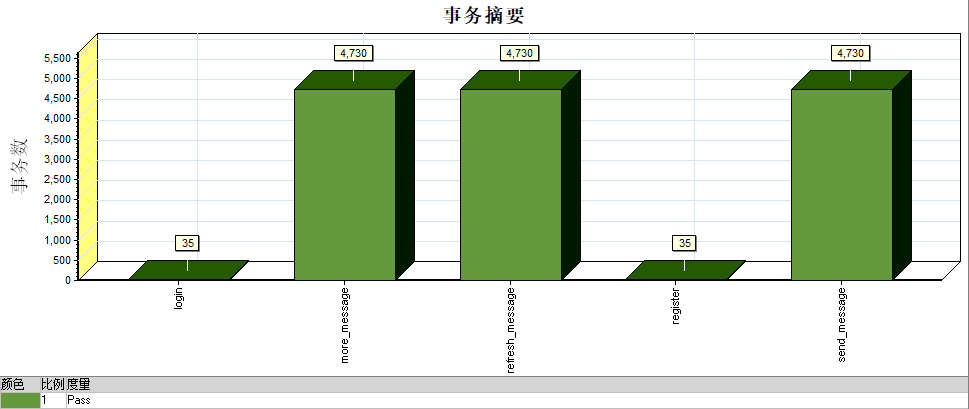
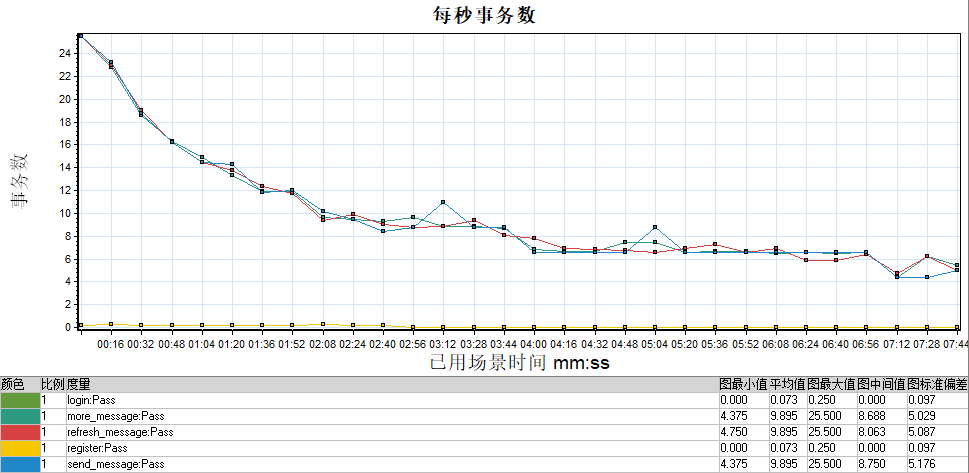
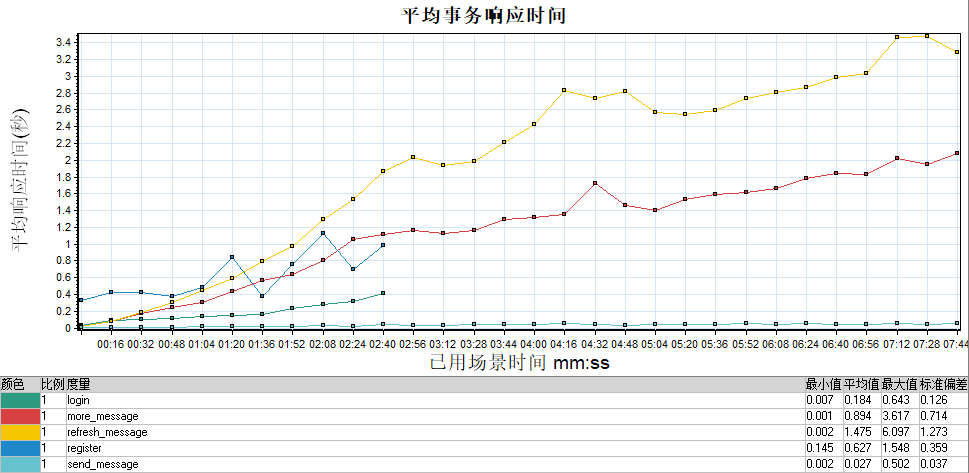
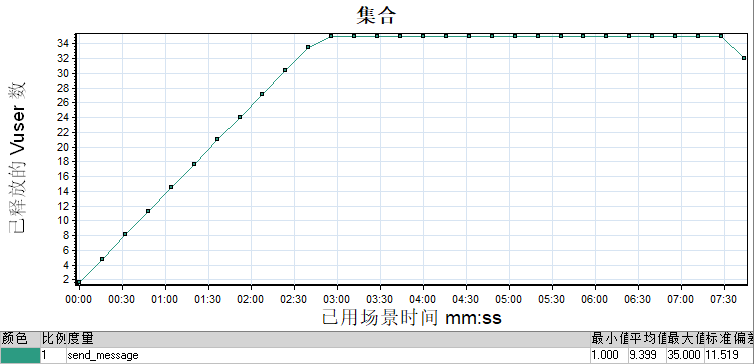
25并发用户数

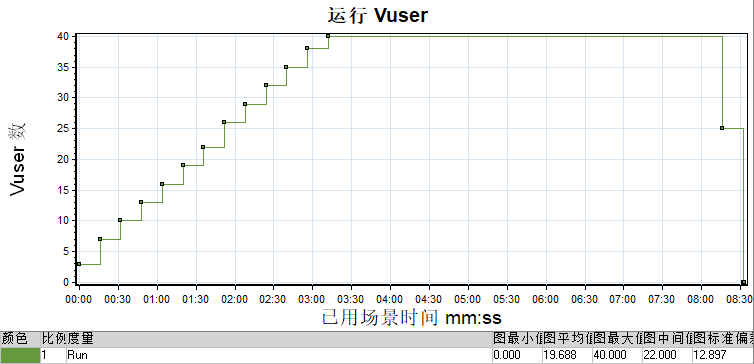
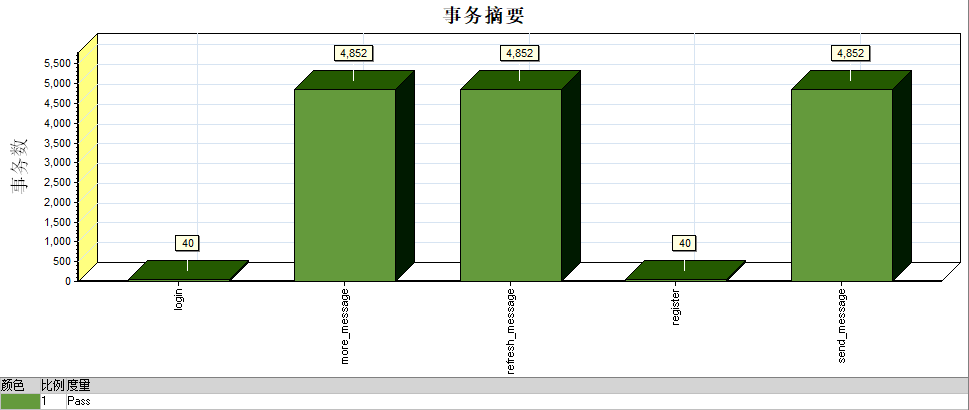
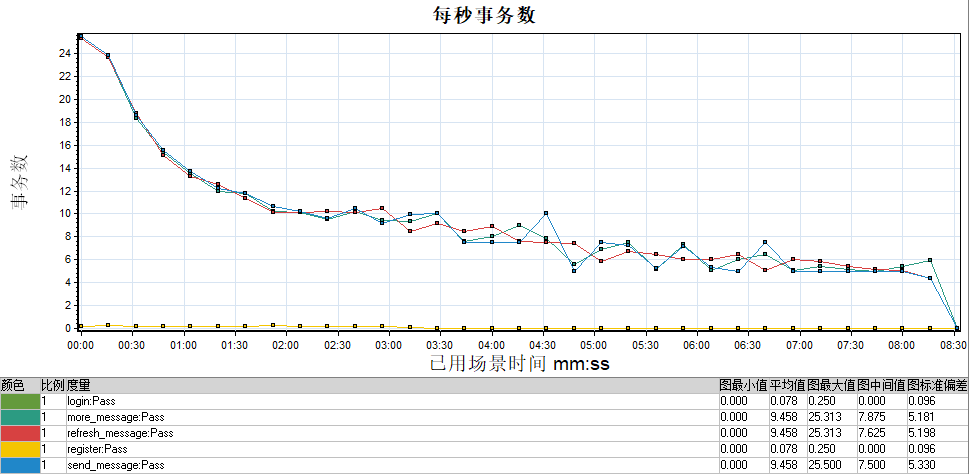
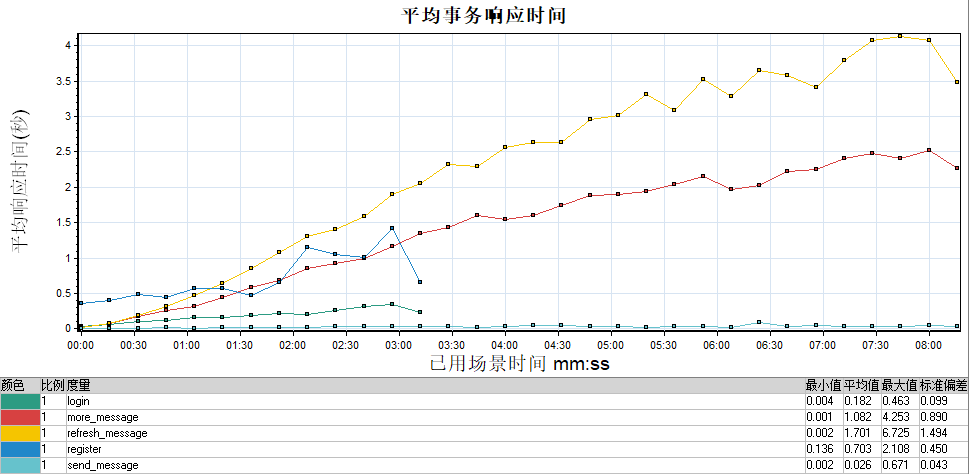
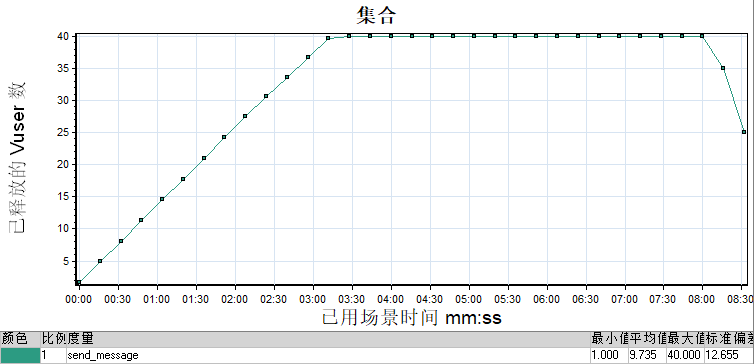
30并发用户数

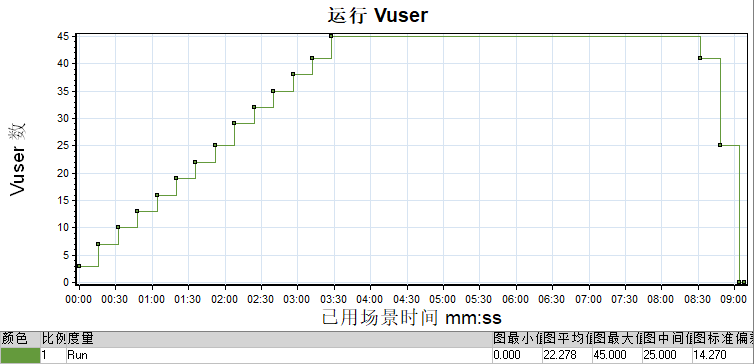
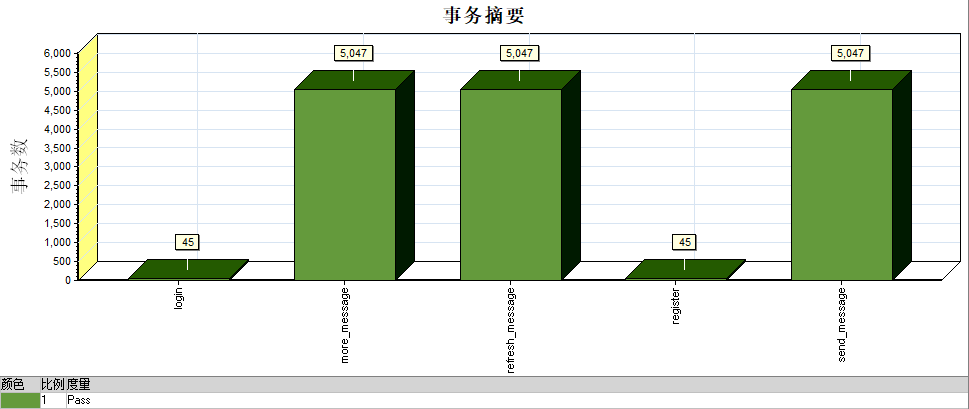
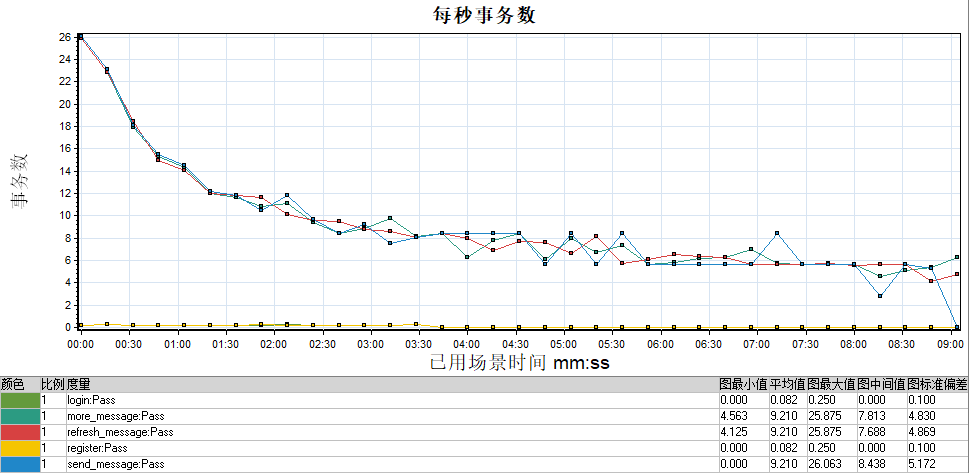
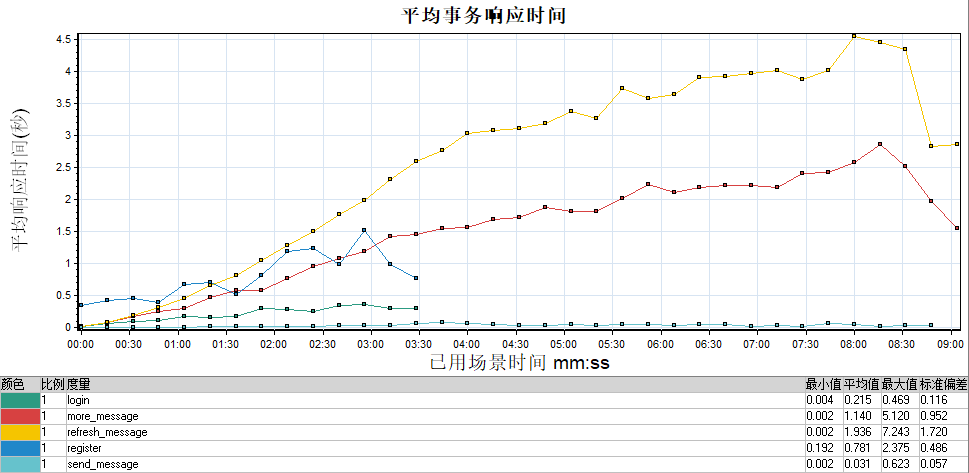
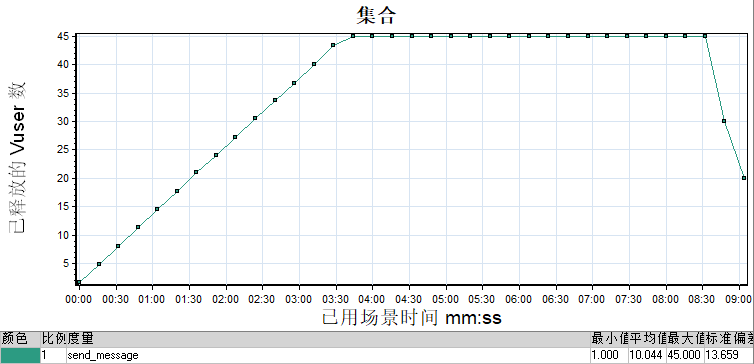
35并发用户数

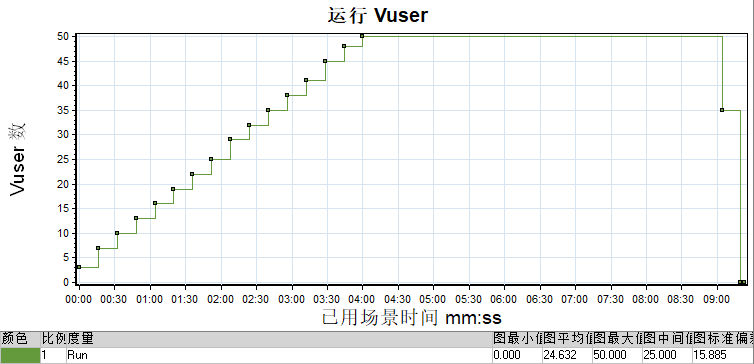
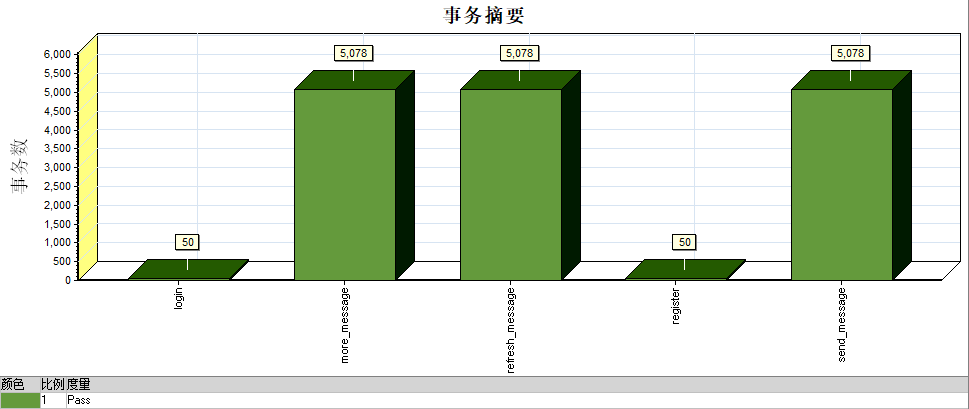
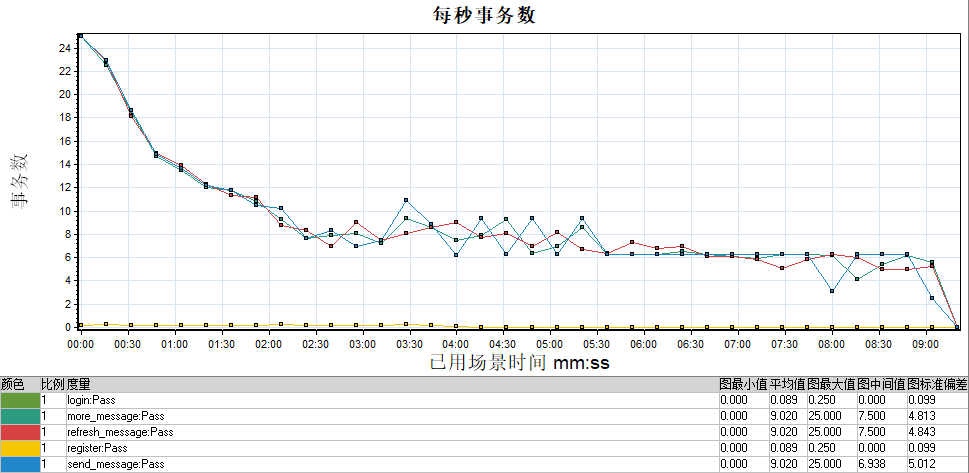
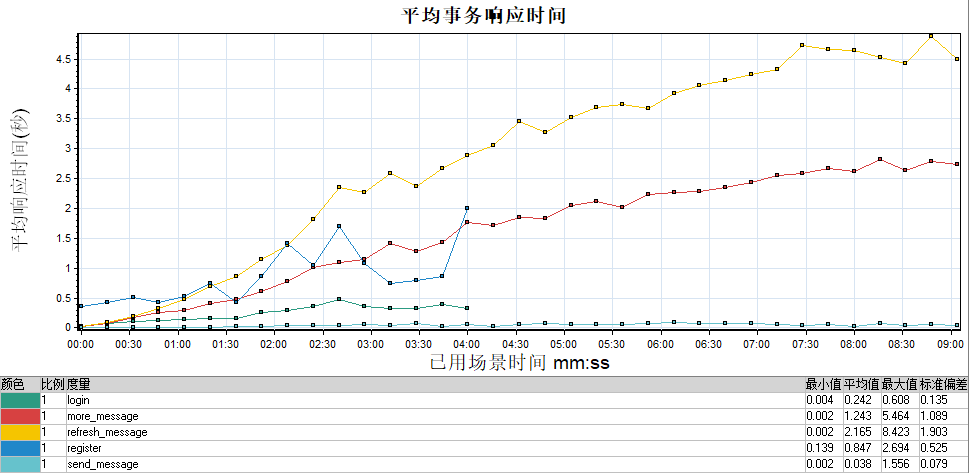
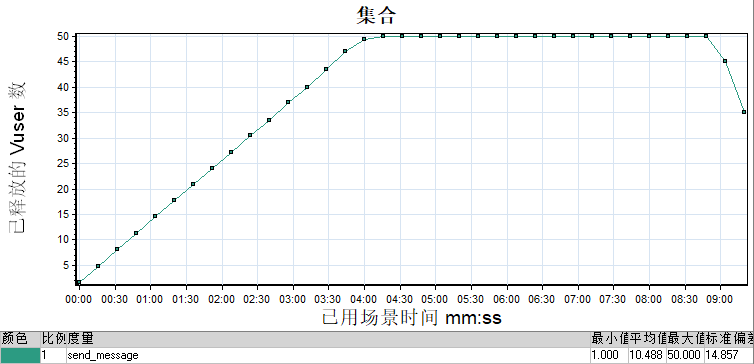
40并发用户数

45并发用户数

50并发用户数

## 联机交易结论

### 最佳并发数/最佳处理能力

最佳并发数为15。此时事务平均响应时间为1.012秒，最大响应时间3.626秒，事务成功率100%，cpu平均利用率78.003%。此时，事务平均响应时间、事务最大响应时间和事务成功率都满足要求，且cpu平均利用率位于理想情况（75%）左右。

### 最大并发数/最大处理能力

最大并发数为35。此时事务平均响应时间为2.396秒，最大响应时间10.216秒，事务成功率100%，cpu平均利用率90.481%。此时，基本满足响应时间2/5/10秒的电商规则，且能够保证事务成功率。cpu平均利用率尚未达到95%的瓶颈。

### 系统容量上限

35并发用户数。

### 系统上限

35并发用户数。

### 业务处理能力

平均每秒事务处理数25~35。

### 并发对性能影响

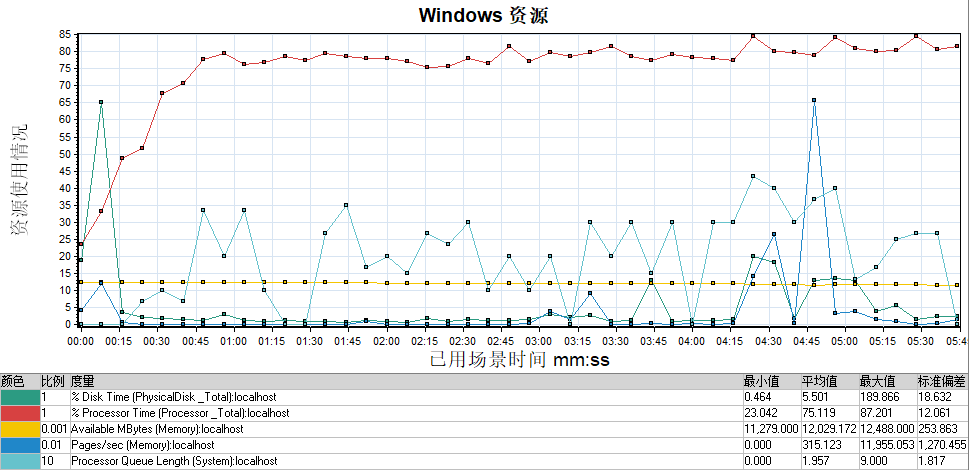
并发用户数越高，事务平均响应时间，事务最大响应时间和cpu平均利用率越大。并发用户数量达到35时，事务平均响应时间和事务最大响应时间过高，且cpu利用率接近瓶颈，不建议继续增加并发数。

## 批量作业结论

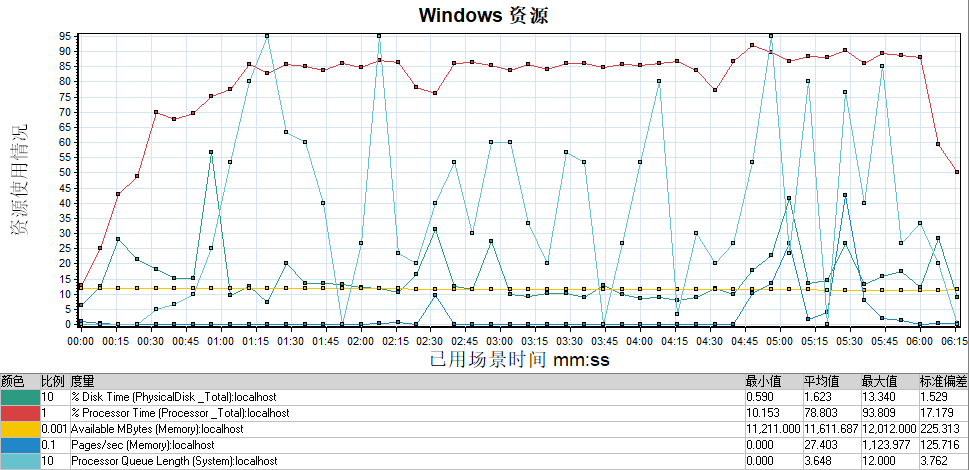
事务成功率能保证达到99%以上。

## 系统性能监控

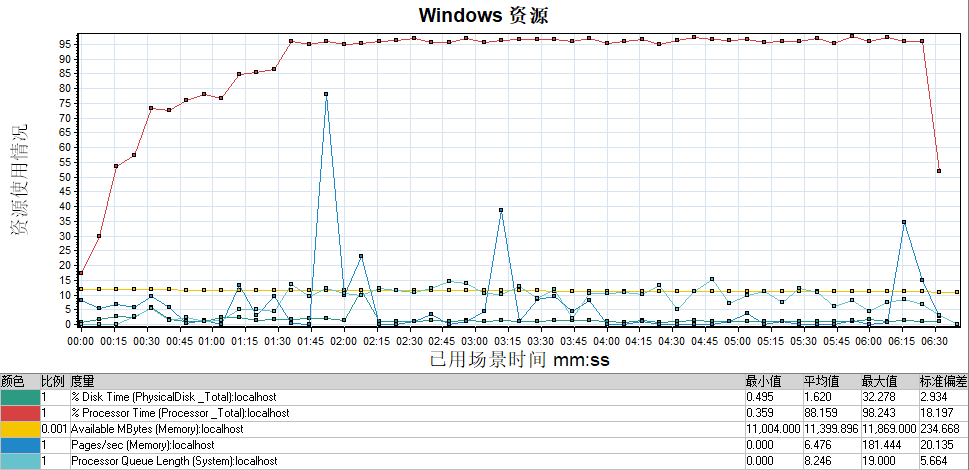
10并发用户数



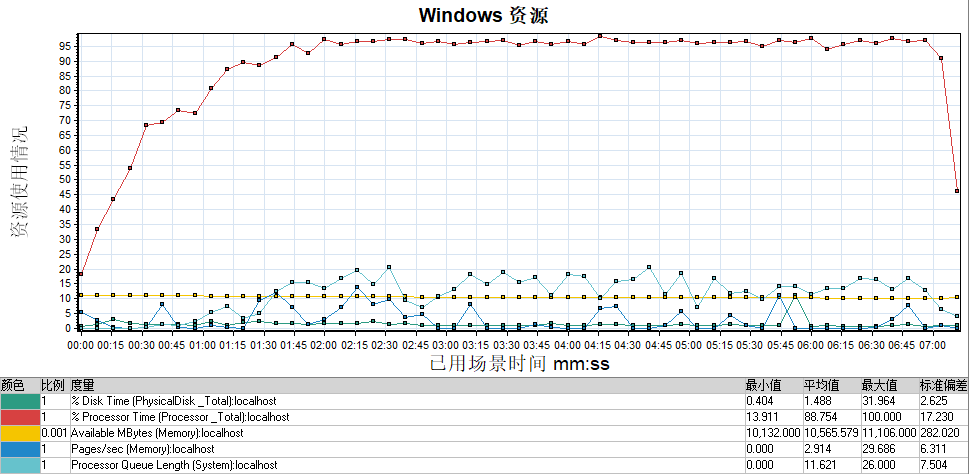
15并发用户数



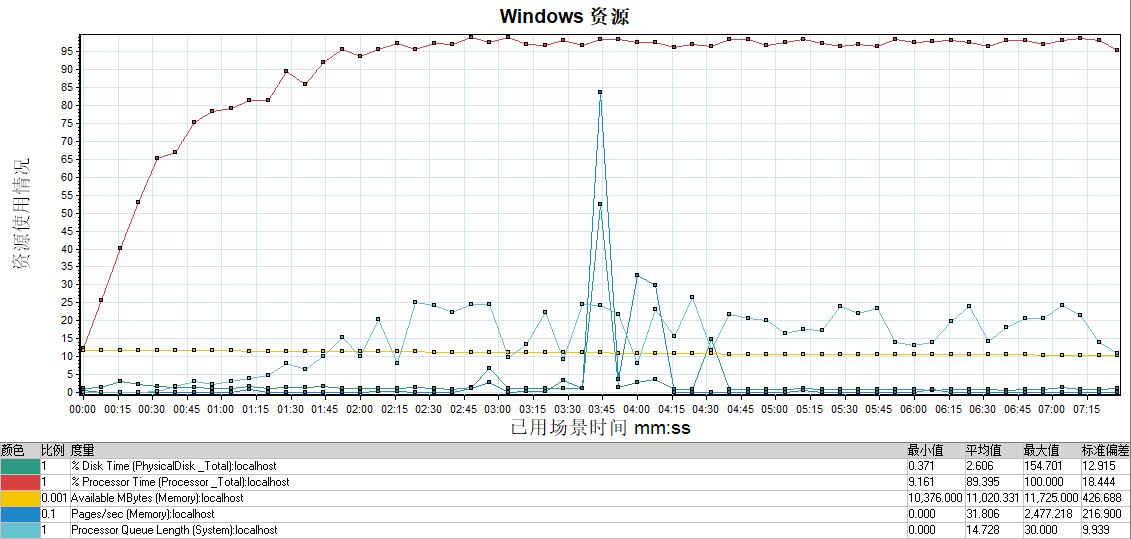
20并发用户数



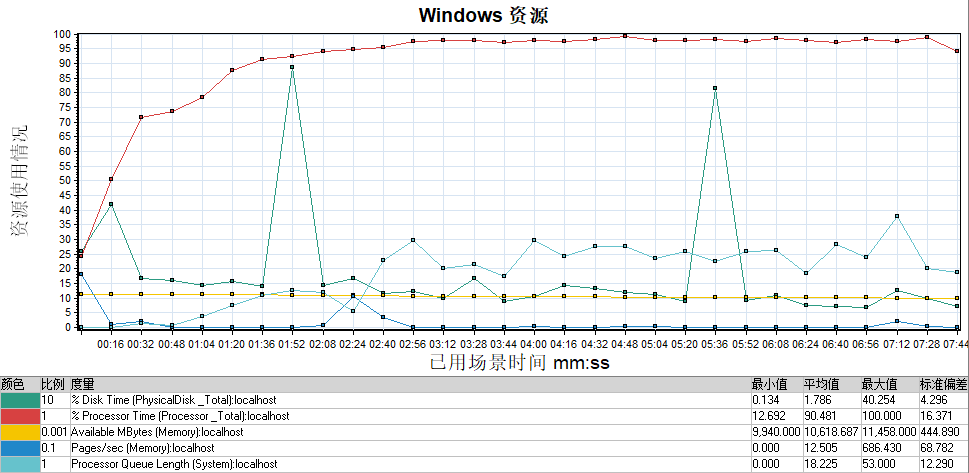
25并发用户数



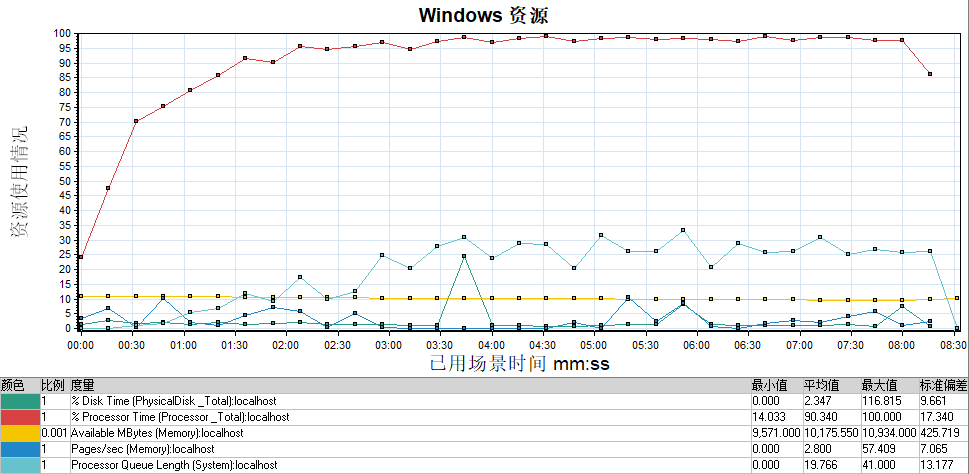
30并发用户数



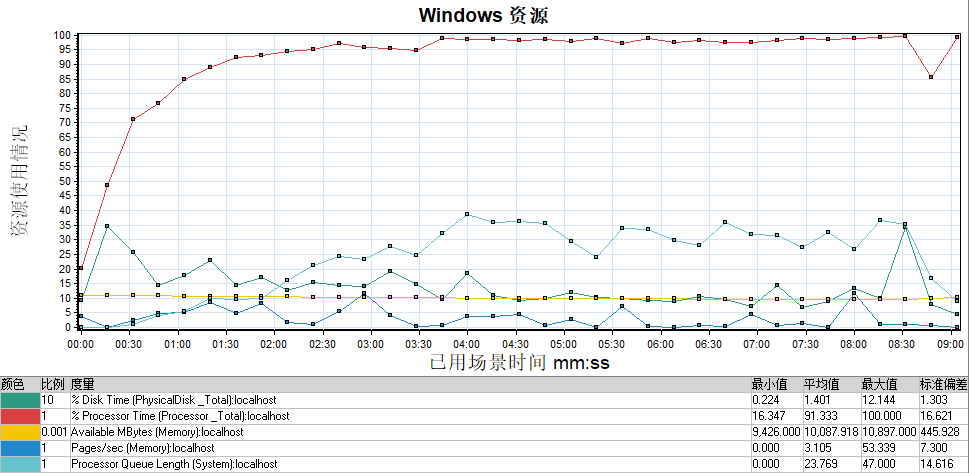
35并发用户数



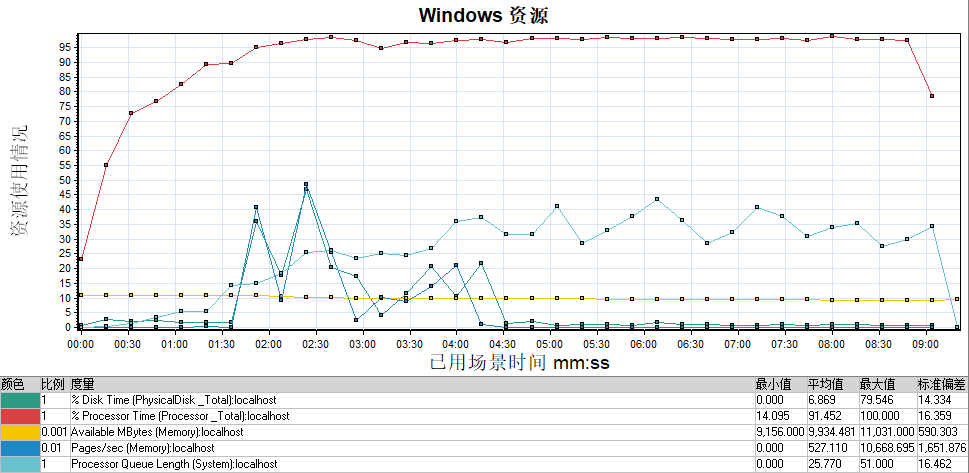
40并发用户数



45并发用户数



50并发用户数



## 事务出错分析

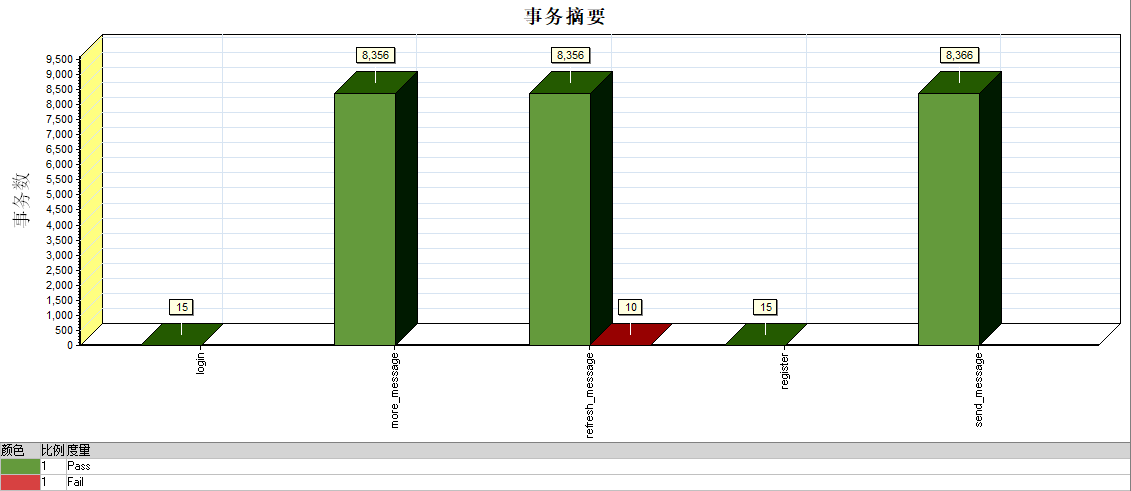
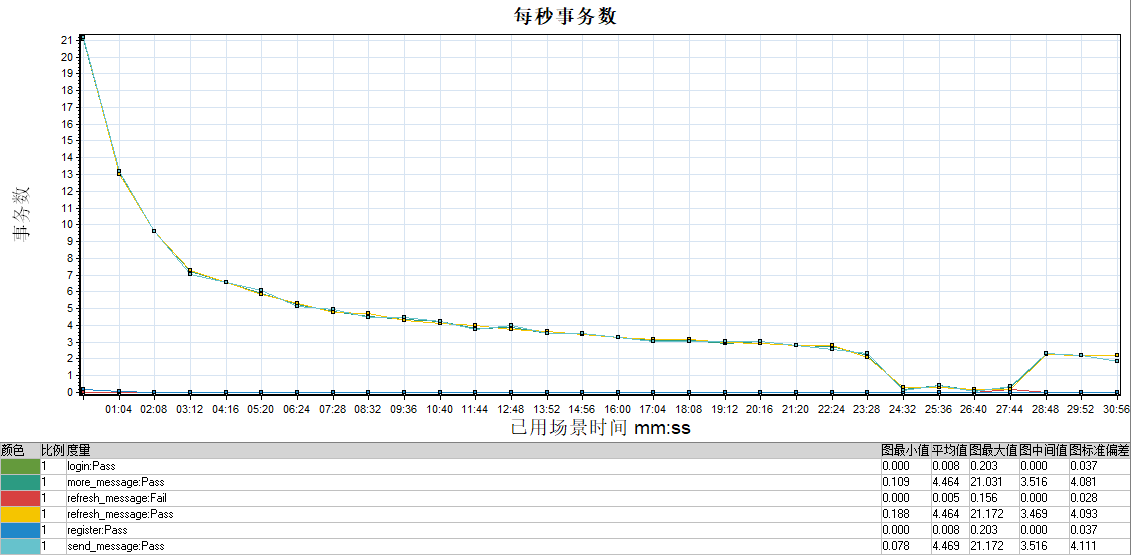
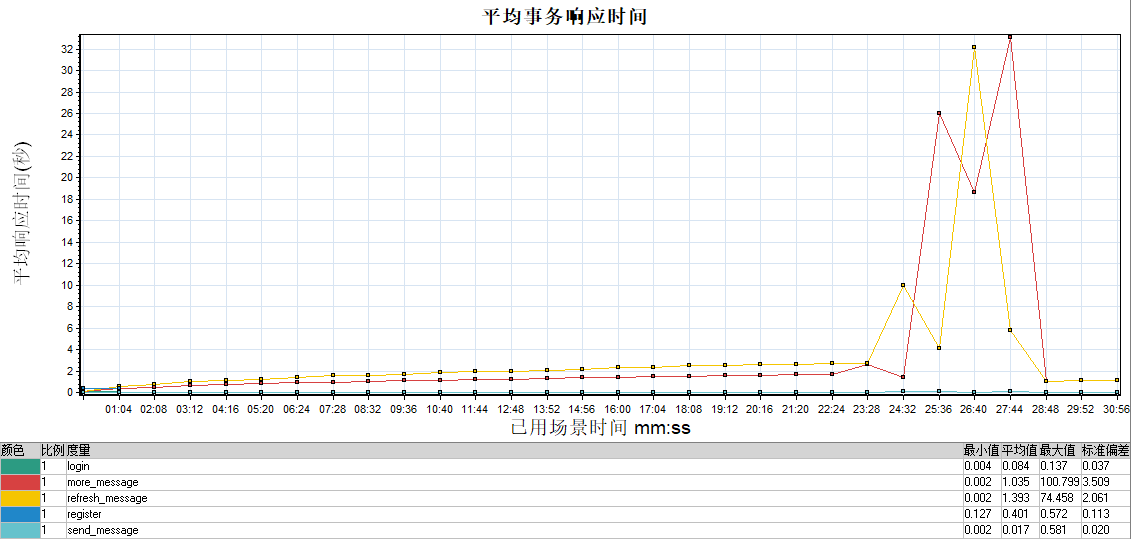
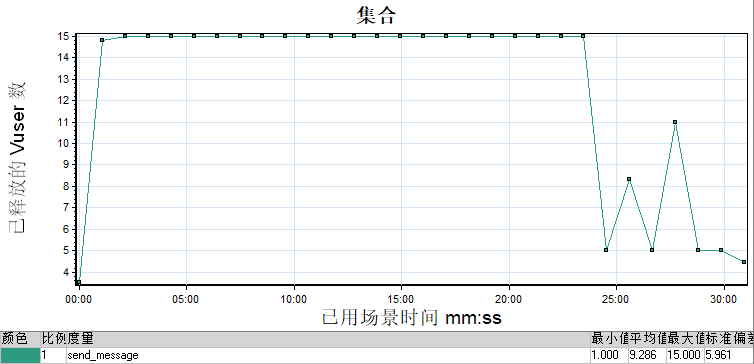
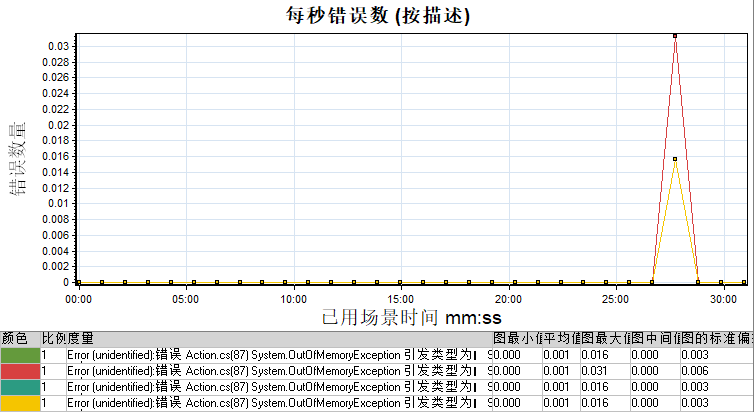
错误类型为System.OutOfMemoryException，为.Net内存溢出。可能需要优化桌面端.Net项目的内存使用。

# 疲劳场景测试报告

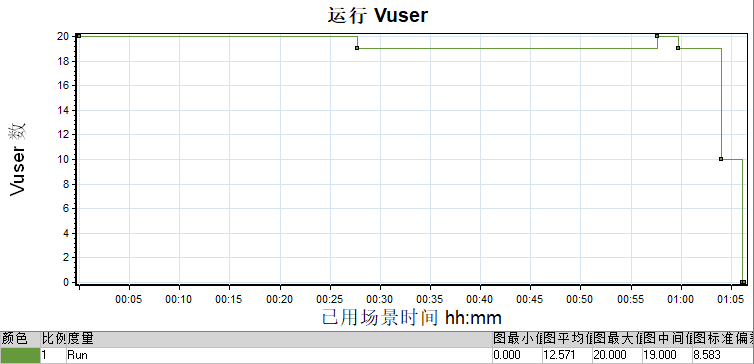
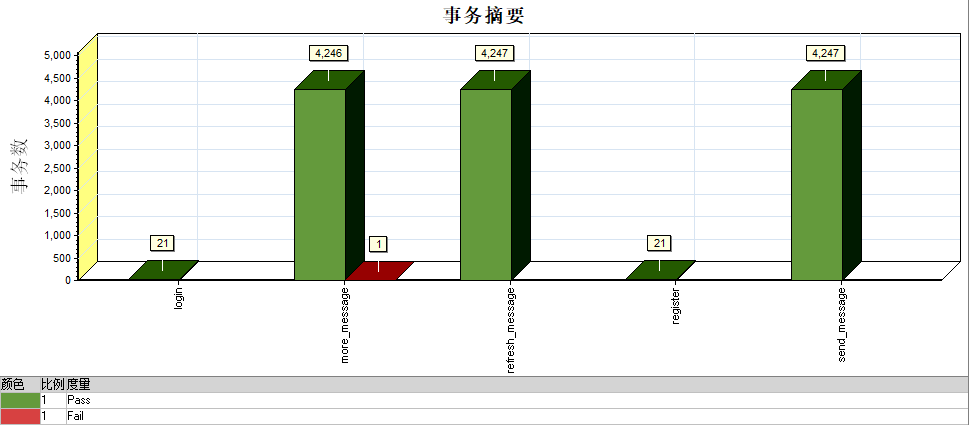
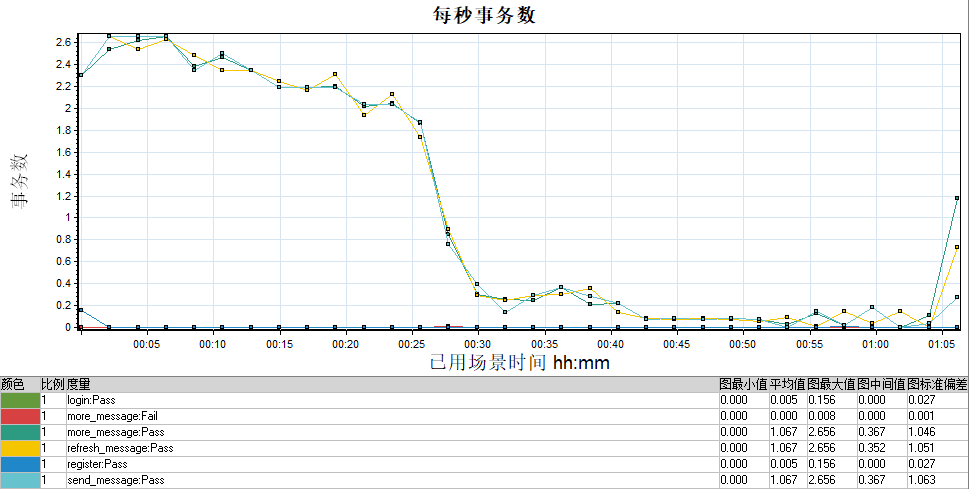
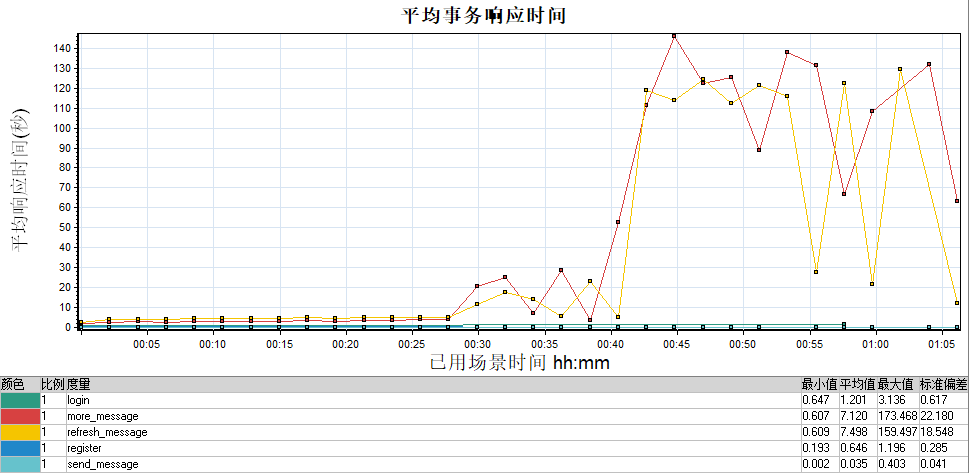
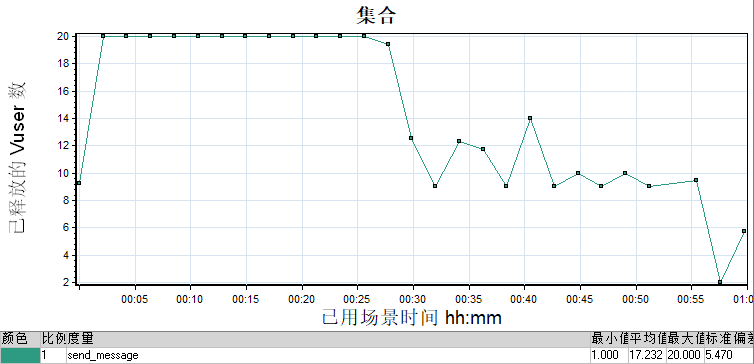
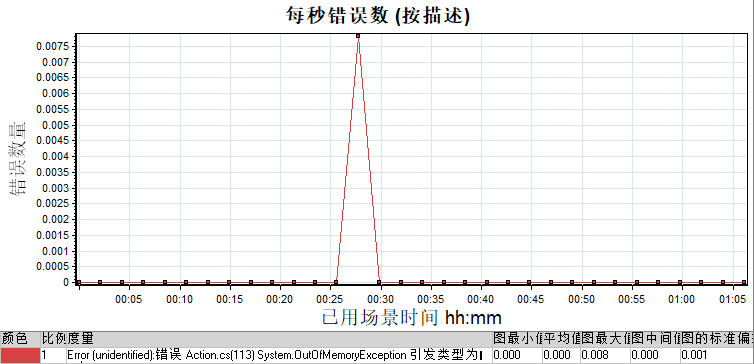
## 测试结果摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试日期 | 成功处理事务数 | | 失败笔数 | 事务通过率 | 事务通过率（去除数据问题） | TPS（笔/秒） | | 事务吞吐率 |
| 账务类 | 查询类 | 平均值 | 峰值 |
| 2019/1/1 | 15136 | 30251 | 11 | 99.97% | 99.97% | 6.877 | 25.703 | \ |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

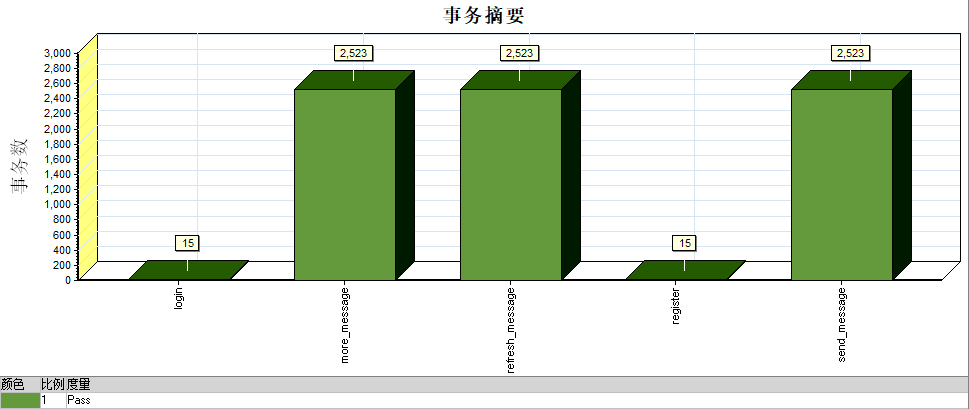
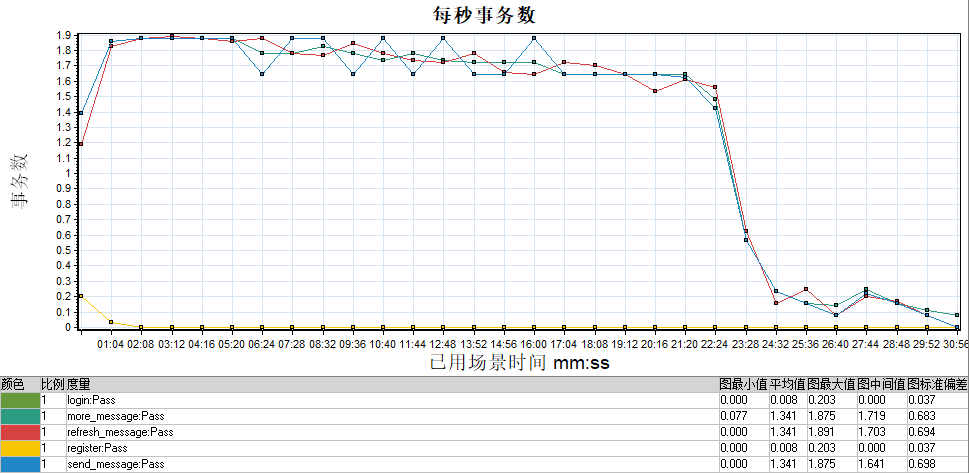
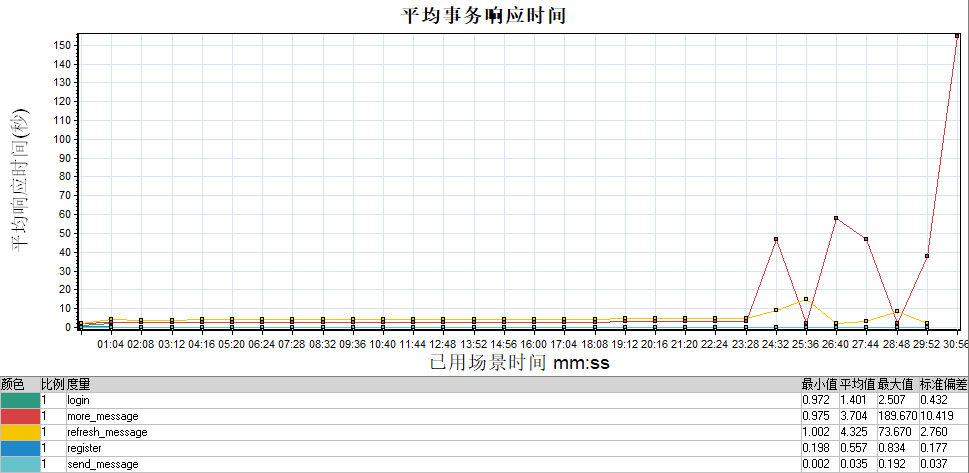
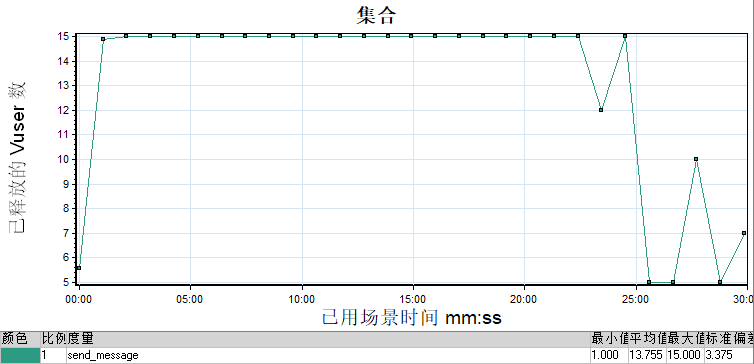
15并发用户运行30分钟

20并发用户运行1小时

15并发用户运行30分钟

## 测试结论

长时间运行系统，能保证事务成功率，但事务的平均响应时间需要优化。

## 响应时间结果

### 第一疲劳响应时间

27分44秒。

### 末一次疲劳响应时间

1小时57分44秒。

## 处理能力结果

### 第一疲劳处理能力

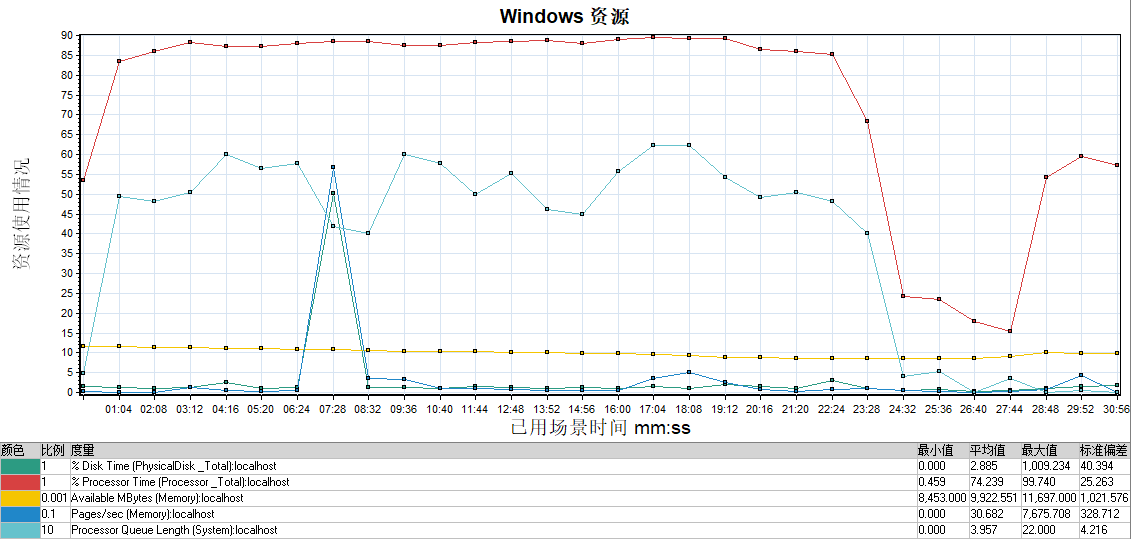
more\_message耗时33秒，refresh\_message耗时6秒，其余正常。

### 末一次疲劳处理能力

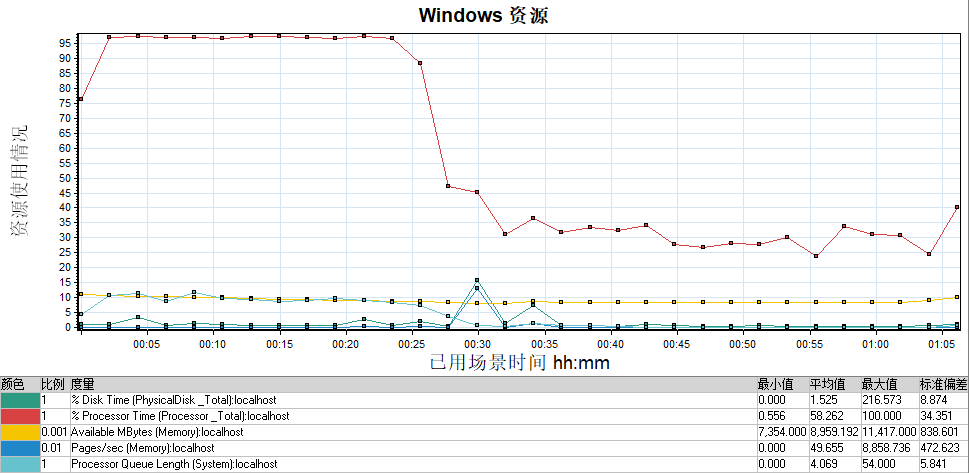
more\_message耗时58秒，其余正常。

## 系统性能监控

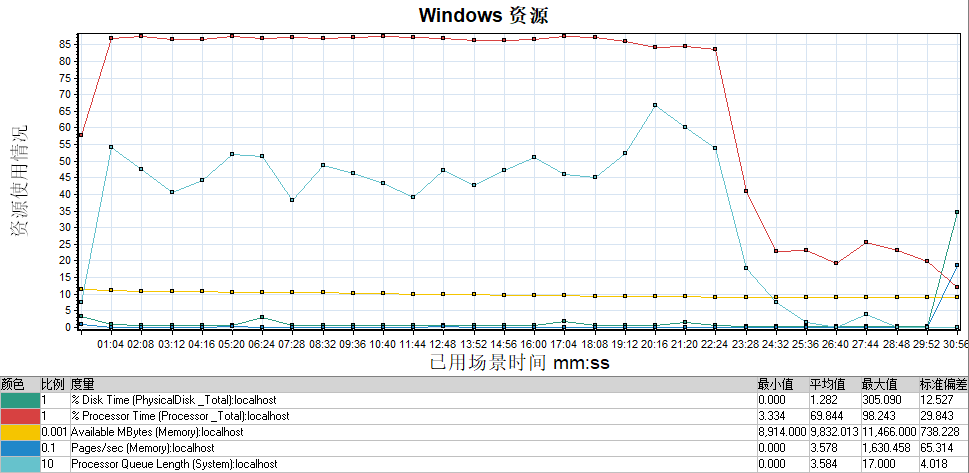
15并发用户运行30分钟



20并发用户运行1小时



15并发用户运行30分钟



## 事务出错分析

### 第一次疲劳分析

报错信息为System.OutOfMemoryException，且此时cpu平均占用率接近90%持续一段时间。事务出错的原因为.Net内存溢出，且可能存在cpu瓶颈问题。

### 末一次疲劳分析

报错信息为System.OutOfMemoryException，且此时cpu平均占用率超过95%持续一段时间。事务出错的原因为.Net内存溢出，且存在cpu瓶颈问题。

# 性能缺陷统计分析

## 缺陷严重程度

小缺陷。

## 缺陷类型分析

1. System.OutOfMemoryException（.Net内存溢出）。
2. 平均事务响应时间随着news表的增大而增大。

## 缺陷原因分析

1. 服务器端的物理内存不够。cpu存在瓶颈。
2. 后端的实现存在一些性能问题。

# 测试环境分析

## 系统架构设计

C#开发的.net引用。(.net framework 4.6.1)

采用了观察者模式。

## 测试环境配置

|  |  |
| --- | --- |
| 硬件配置 | 软件配置 |
| Intel® Core™ i7-6700HQ  @ 2.60Hz 2.59GHz  8GB内存  HGST HTS721010A9E630 机械硬盘1T | Windows 10 64位  Loadrunner 12.55  Mysql 5.7 |

## 环境差异分析

无测试环境差异。