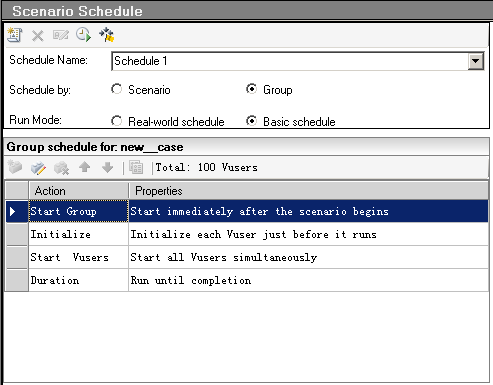
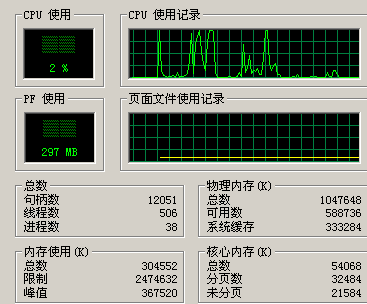
100虚拟用户

场景设置：



Cpu内存使用情况：



Cpu使用存在五个峰值，并且系统提示清理无关应用使用。

根据场景分析：五个峰值出现时，系统可能进行的时如下操作：

（1）用户登录系统成功

（2）新建bug请求

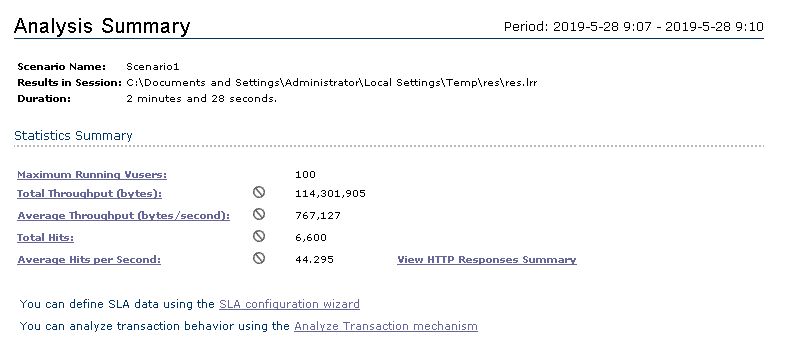
（3）新建bug内容编写

（4）保存内容

（5）返回bug列表

内存使用相对稳定，未出现异常情况

Analysis Summary：



最大运行用户为：100

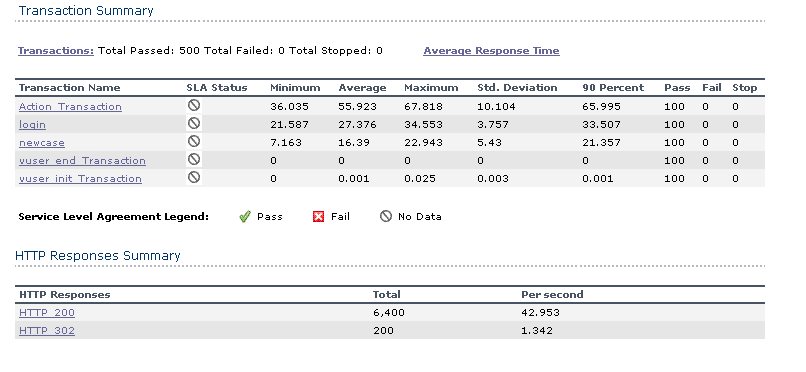
总吞吐量：114301905（bytes）

平均吞吐量：767127（bytes/s）

总点击量：6600

平均每秒点击量：44.295

Transaction Summary：



Action事务：

最小并发：36.035

最大并发：67.818

平均并发：55.923

通过：100

90 percent：65.995

标准差：10.104

Login事务：

最小并发：21.587

最大并发：34.553

平均并发：27.376

通过：100

90 percent：33.507

标准差：3.757

Newcase事务：

最小并发：7.163

最大并发：22.943

平均并发：16.39

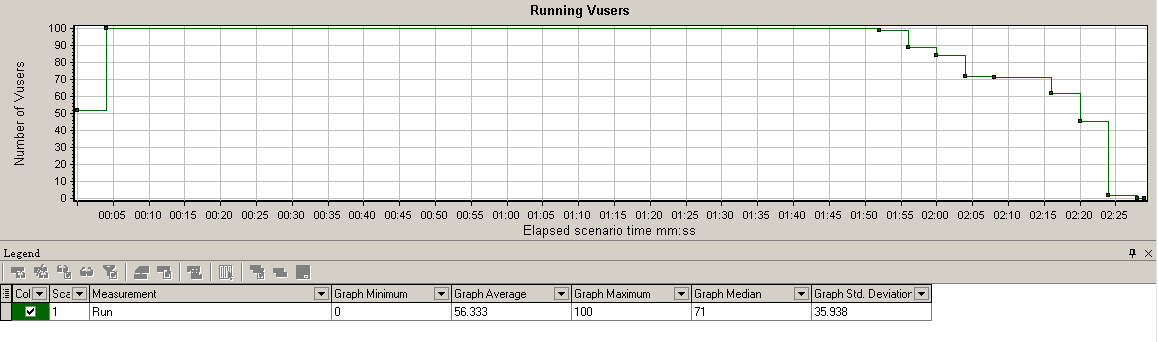
通过： 100

90ercent：21.357

标准差：5.43

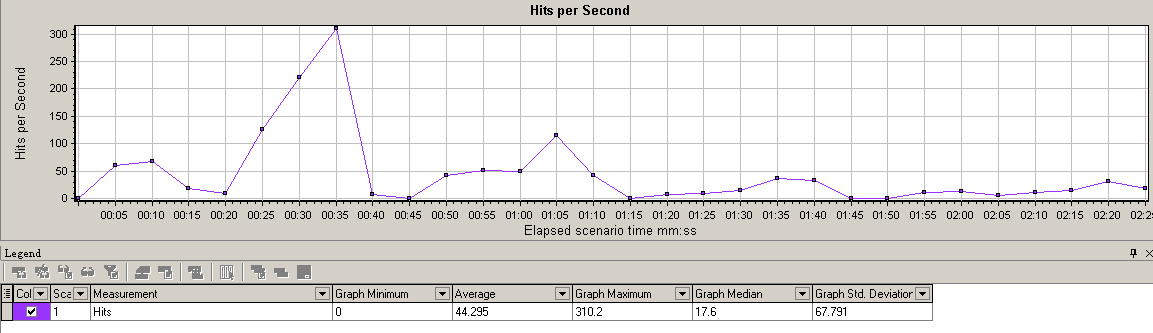
由此可以得出，登陆事务标准差较小。其操作相对于稳定。

Runner Vusers：



00：05s---01：50时间段内：100用户进行并发执行

Hits per Second：



最小点击量;

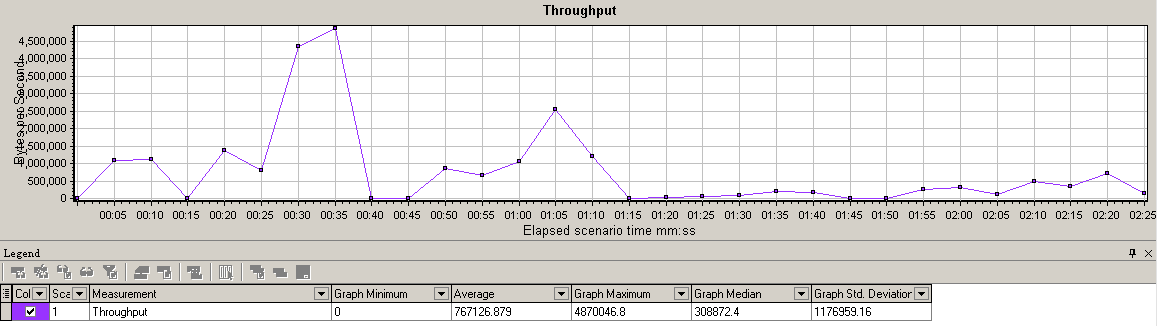
平均点击量：44.295

最大点击量：310.2（时间为：00：35）

中位数：17.6

标准差：67.791

Throughput：



最小吞吐量：0

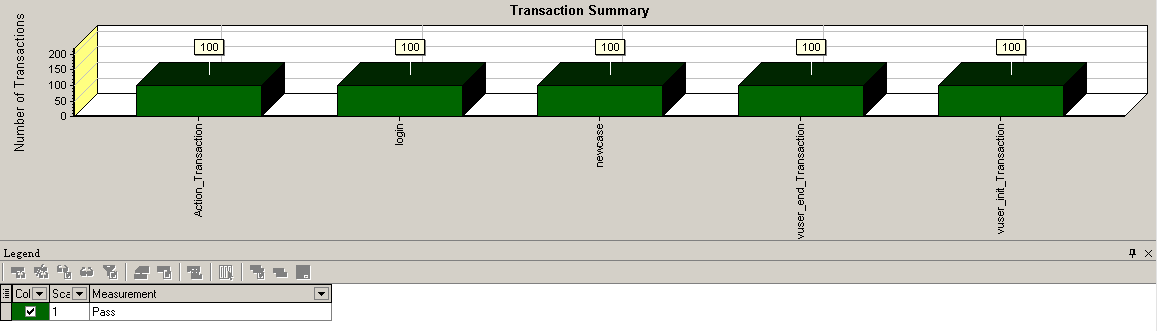
最大吞吐量：4870046.8

平均吞吐量：767126.879

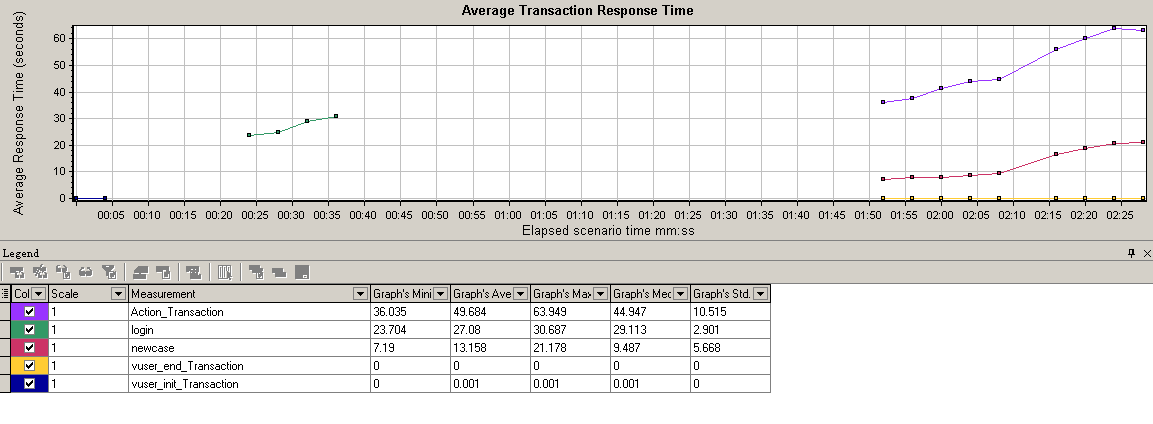
中位数：308872.4

标准差：1176959.16

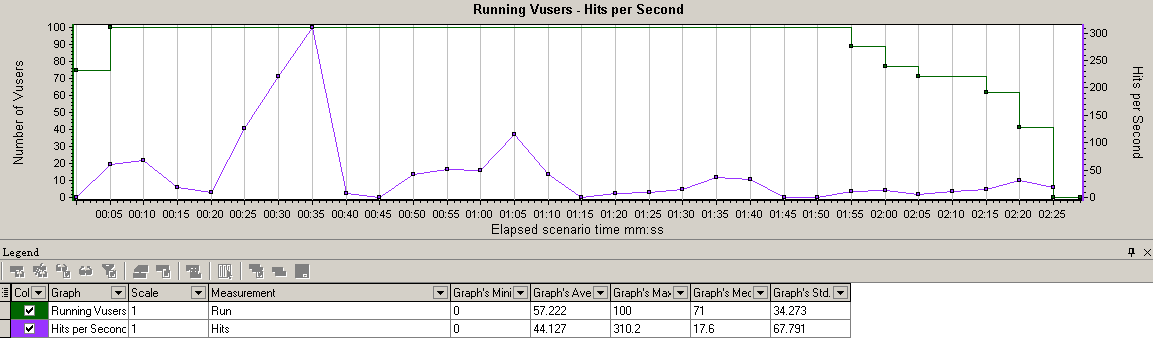
Transaction Summary：



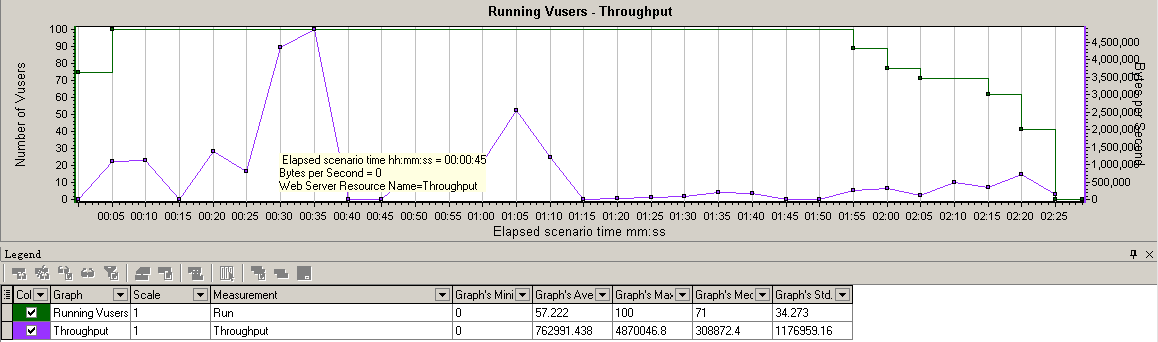
Average Transaction Response Time



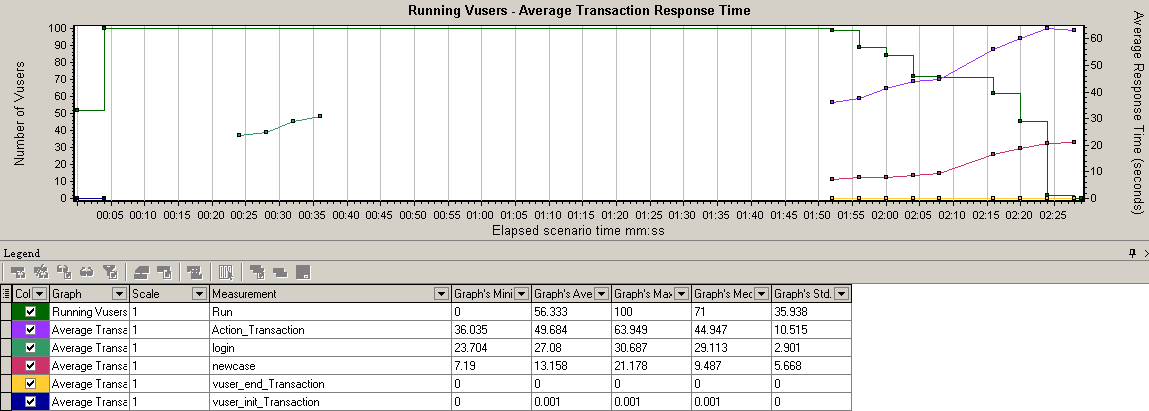
Running Vusers—Hits per Second：



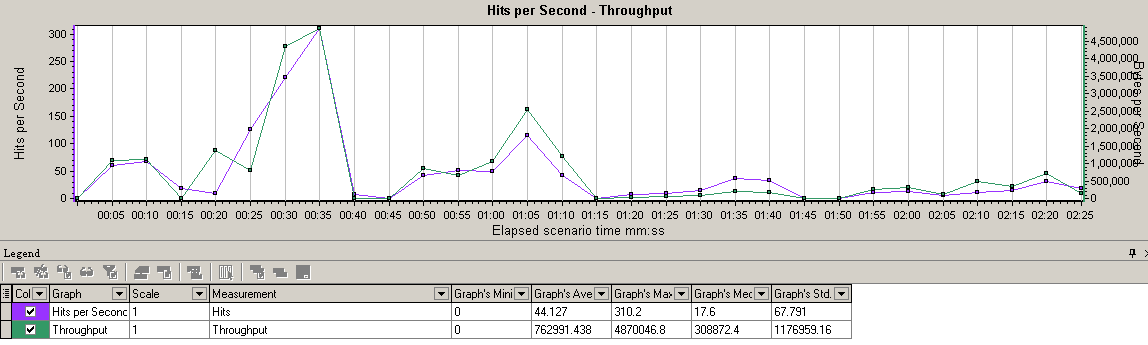
Running Vusers---Throughout：

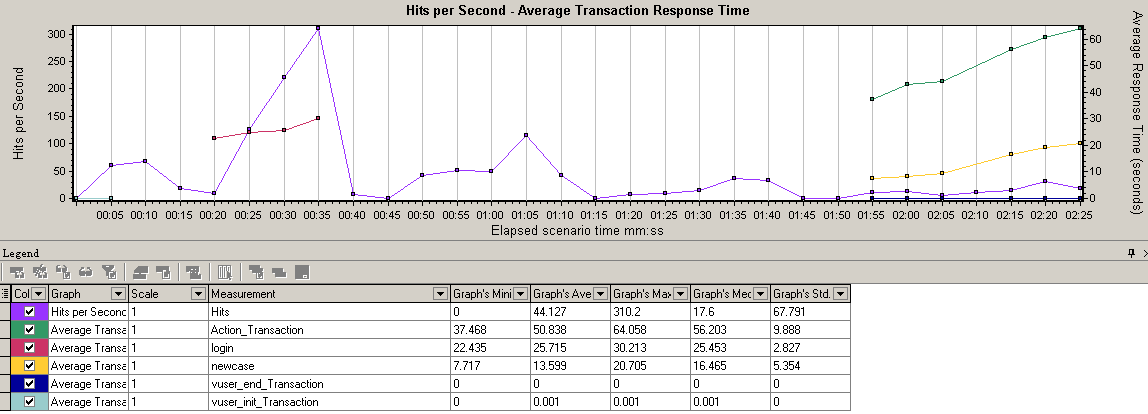


Running Vusers---Average Transaction Response Time

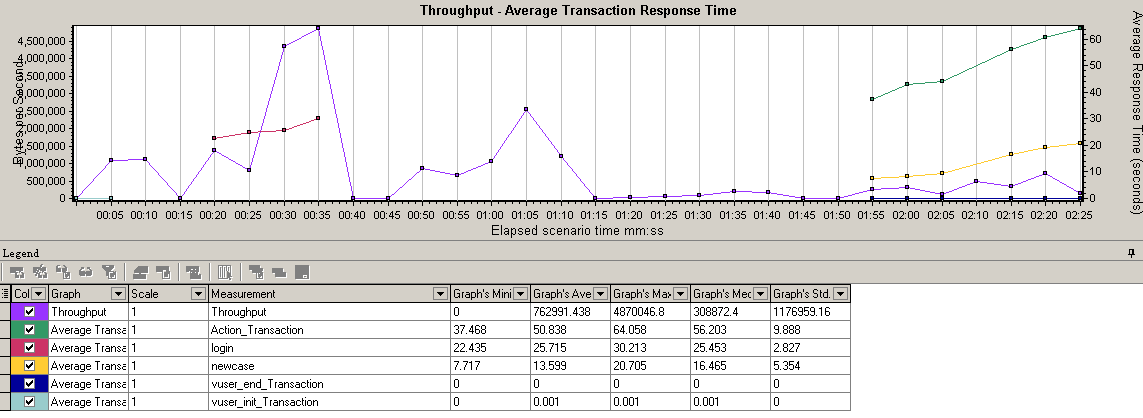


Hits per Second---Throughput



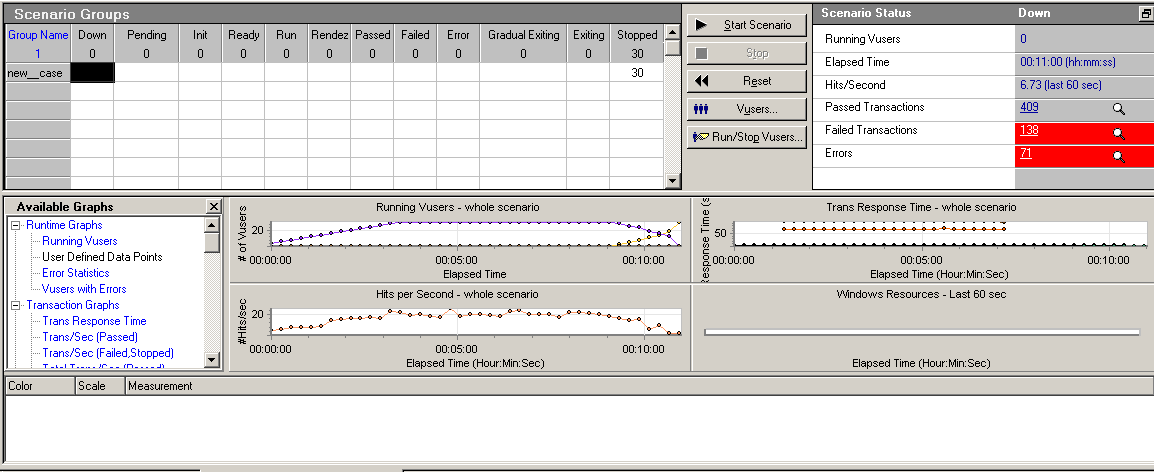


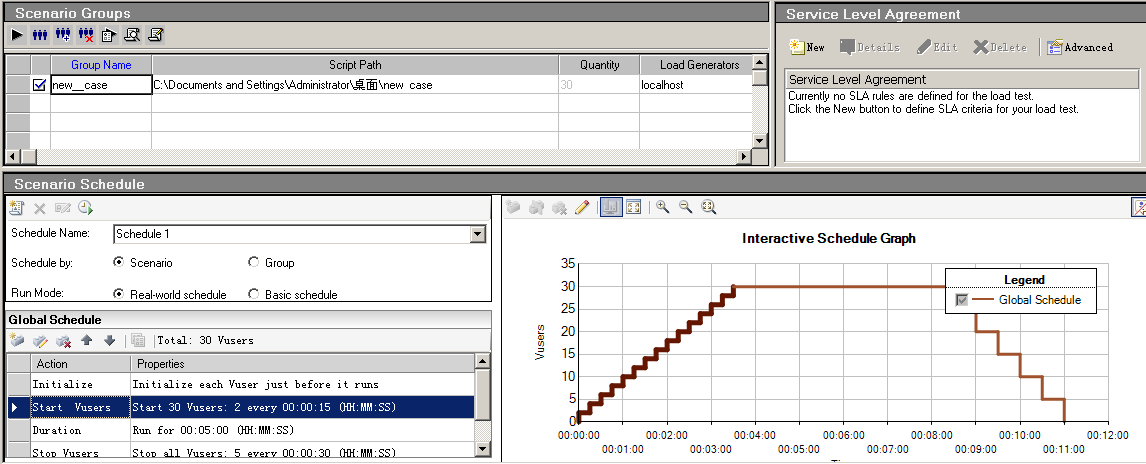
Throughput----Average Transaction response Time



30虚拟用户

场景设置：

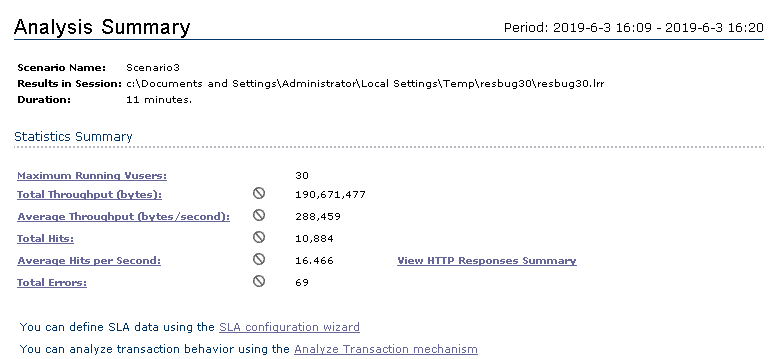




AnalysisiSummary

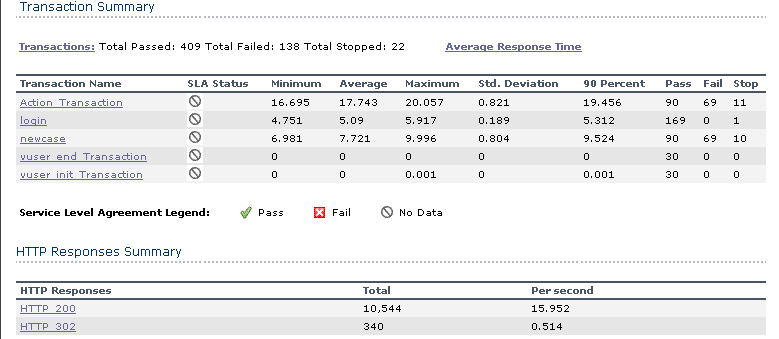
下图说明本次测试运行的最大并发数为30，总吞吐量为190671477字节，平均每秒吞吐量为288459字节，总的请求数为10884，平均每秒的请求为16.466。对于吞吐量，单位时间内吞吐量越大，说明服务器的处理能越好，而请求数仅表示客户端向服务器发出的请求数，与吞吐量一般是成正比关系。

---------------------



Transaction Summary

这部分给出了场景执行结束后相关Action的平均响应时间、通过率等情况.从该图中我们可以得到每个Action的平均响应时间与业务成功率。



该部分显示在场景执行中，每次发送HTTP请求所得到的状态。本次测试共发送10884次请求，其详情如上图其中“HTTP200”有10544次，“HTTP302”有340次，占整体请求数的3%，这说明基本所有的请求都能成功。

Running Vuser

“Running Vusers（运行的并发数）”显示了在场景执行过程中并发数的执行情况。它们显示Vuser的状态、完成脚本的Vuser的数量以及集合统计信息，将这些图与事务图结合使用可以确定Vuser的数量对事务响应时间产生的影响。从图中我们可以看到，Vusers的运行趋势与我们场景执行计划中的设置是一样，表明在场景执行过程中，Vusers是按照我们预期的设置运行的，没有Vuser出现运行错误，这样从另一个侧面说明我们的参数化设置是正确的，因为使用唯一数进行参数化设置，如果设置不正确，将会导致Vuser运行错误。在脚本中我们加入了这样一段代码：

if (atoi(lr\_eval\_string("{num}")) > 0){

lr\_output\_message("登录成功，继续执行.");

}

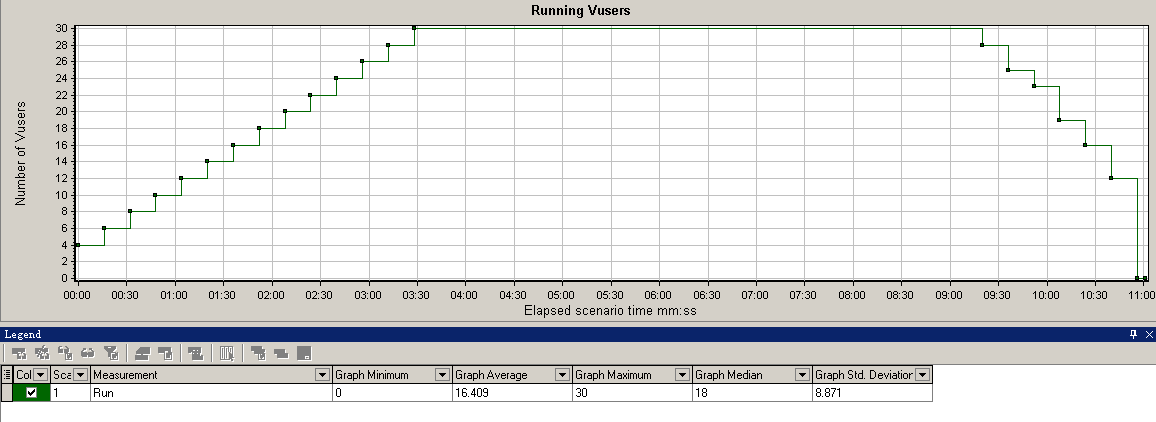
else{

lr\_error\_message("登录失败，退出测试");

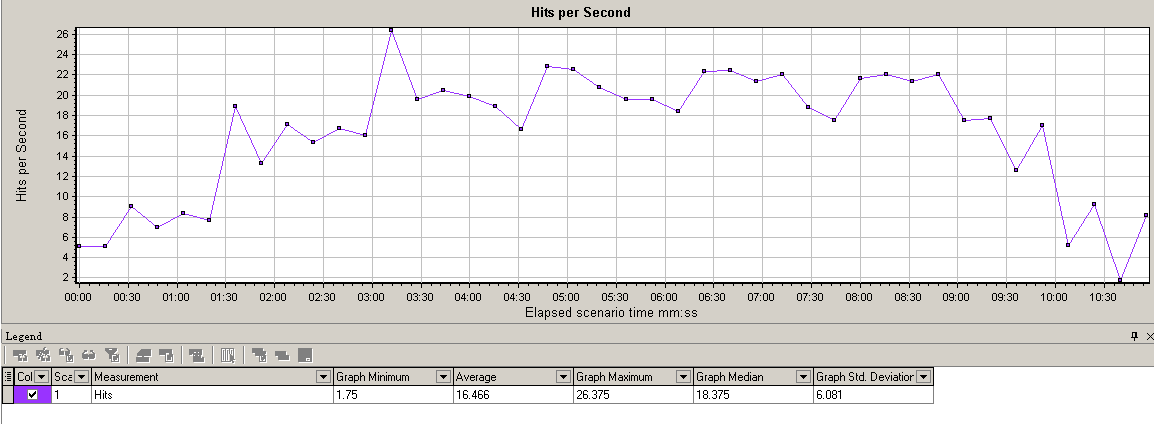
return -1;

}

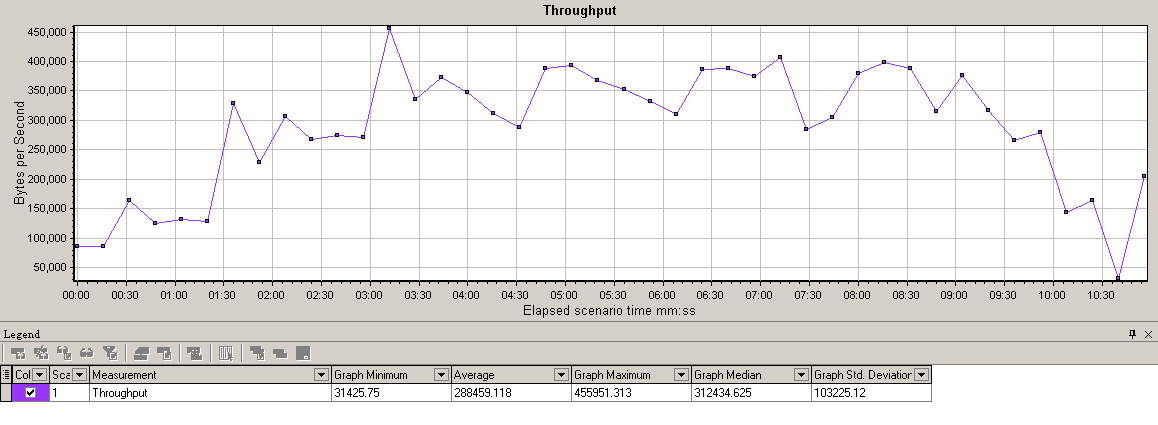
上述代码的意思是说，如果登录失败了，就退出脚本的迭代，那么什么原因可能会导致登录失败呢？就是我们前面参数化的设置，一旦Vuser分配不到正确的登录账号，就可能导致登录失败，从而引起Vuser停止运行。所以，从图的表现，可以认为参数化是没有问题的。



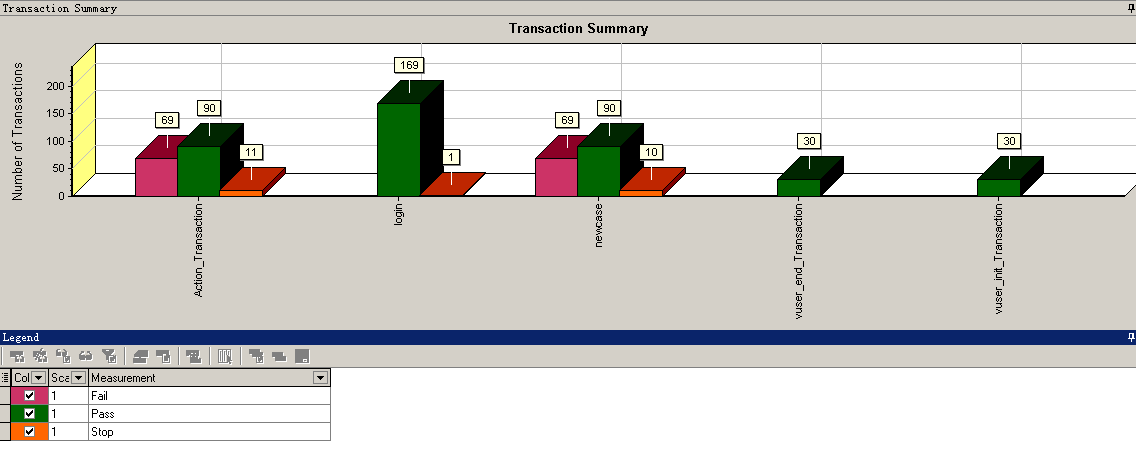
Hits Per Second



Throughput



Transaction Summary



业务成功率就是事务成功率，用户一般把一个Action当做一笔业务，在“Transaction Summary”中我们可以很明确地看到每个事务的执行状态。

从上图可以看出，所有的buser end/init-transaction都是绿色的。也就表示在场景执行的的时间里，共完成了30次操作，成功率为100%。

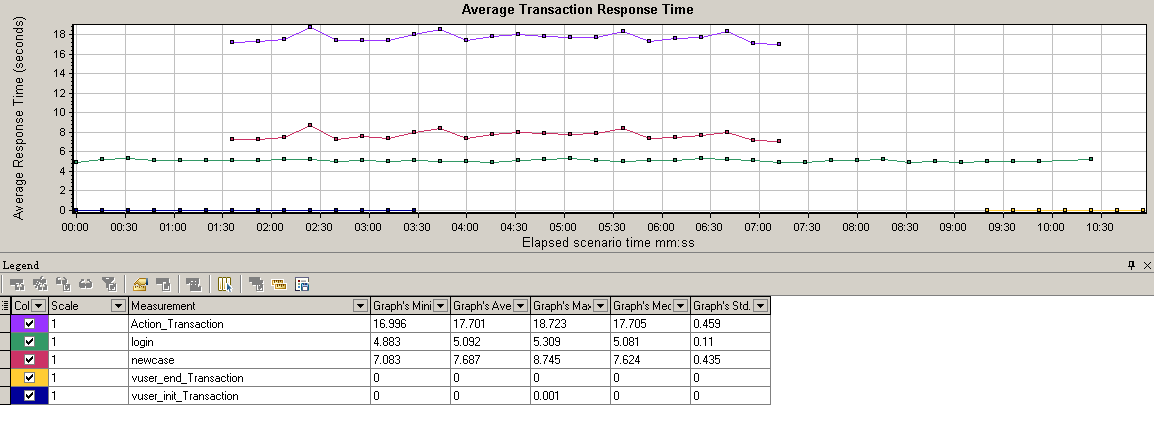
完成170次Action-transaction，其中成功90次，失败69次，停止11次。

完成170次login操作，成功169次，停止1次

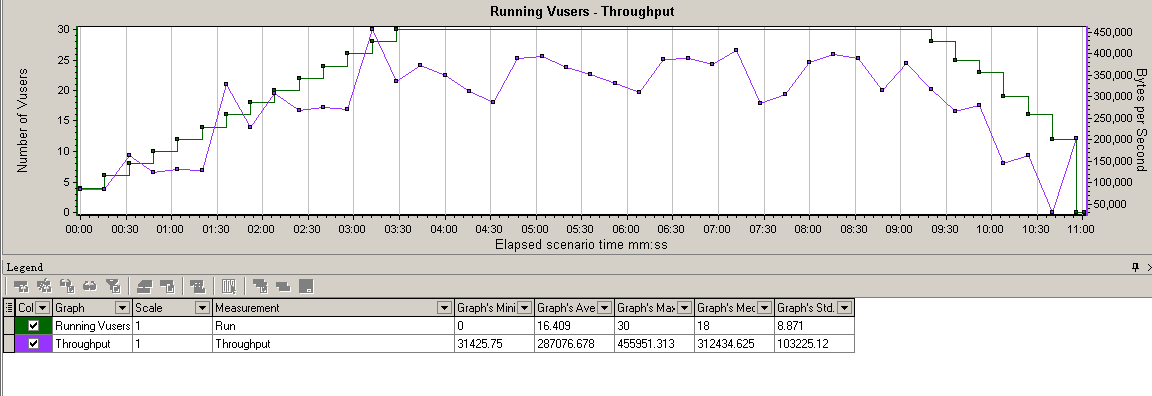
完成169次newcase操作，成功90次，失败69次，停止10次。

Average Transaction Response Time

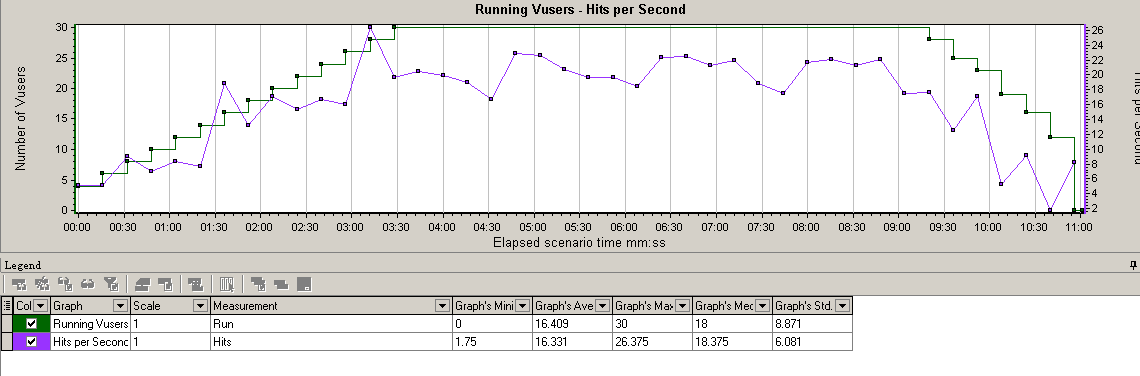
 通过图可以得到平均事务响应时间的详细信息，我们可以看到整体趋势比较平滑。因此结合图Analysis和Transaction可以得到，在登录事务中，“90 Percent”为5.081s，它表示90%的用户在登录时，服务器的响应都维持在这个5.081附近，响应速度相对较慢。



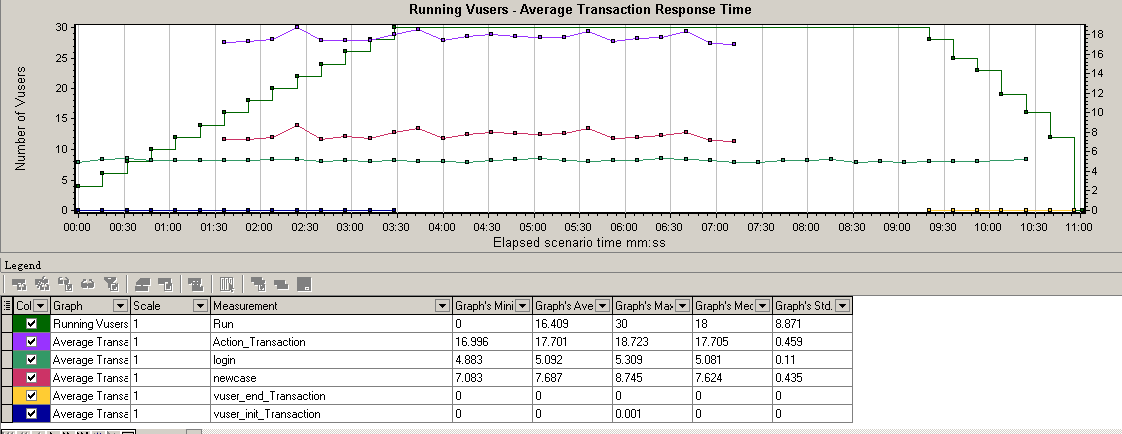




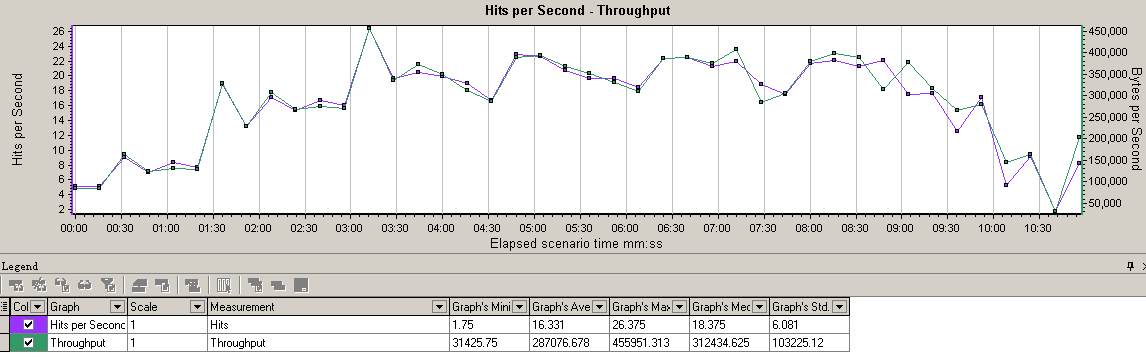




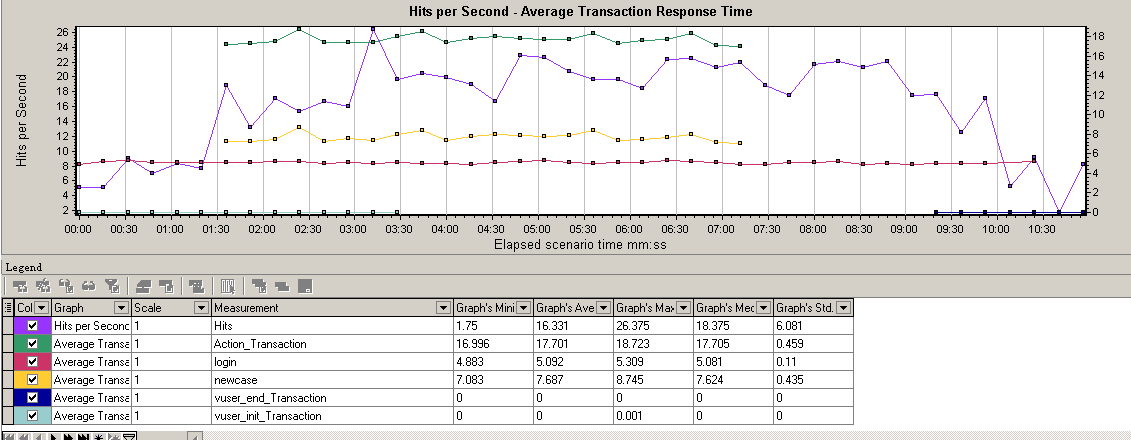




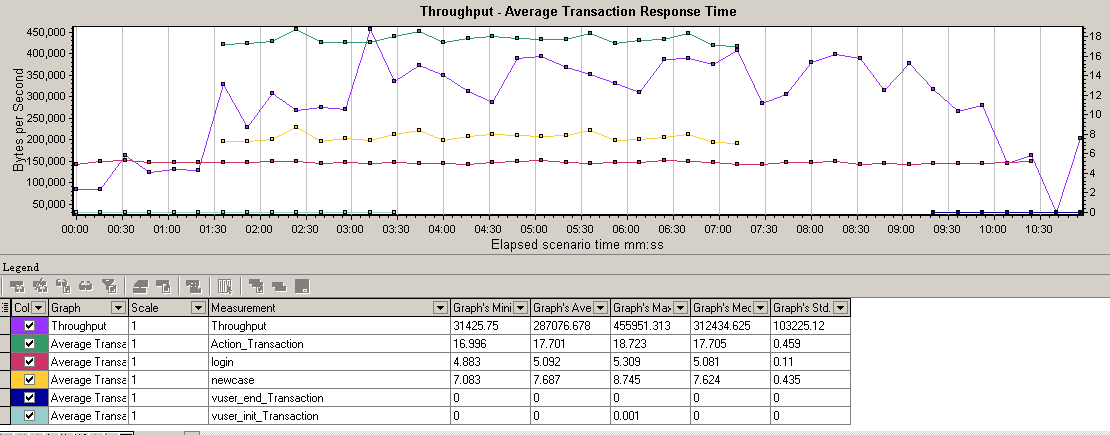




“Hits per Second（每秒点击数）”反映了每秒向服务器提交的请求数量。上图显示的是“Hits per Second”与“Throughput”的复合图。如果两种图像的曲线都正常且基本一致，说明服务器能及时接受客户端的请求，并能返回结果。从图中可以看出，整体表现还是不错的。







**不同的并发用户数对比得出的结论**

由上述两种不同的执行情况可以得出，30并发用户数，其运行时，CPU 内存不会存在崩溃现象，100用户在运行时会存在cpu崩溃现象，但并不是持续处于崩溃状态。因此在运行时需要注意注意并发用户数的选择，以及对系统各项的指标进行跟踪监测。