5.4.2 测试结果分析

LoadRunner 性能测试结果分析是个复杂的过程，通常可以从结果摘要、并发数、平均事务响应时间、每秒点击数、业务成功率、系统资源、网页细分图、Web 服务器资源、数据库服务器资源等几个方面分析，如图 5- 1 所示。性能测试结果分析的一个重要的原则是以性能测试的需求指标为导向。我们回顾一下本次性能测试的目的，正如**错误！未找到引用源。** 所列的指标，本次测试的要求是验证在 30 分钟内完成 2000 次用户登录系统，然后进行考勤业务，最后退出，在业务操作过程中页面的响应时间不超过 3 秒，并且服务器的 CPU 使用率、内存使用率分别不超过 75%、70%，那么按照所示的流程，我们开始分析，看看本次测试是否达到了预期的性能指标，其中又有哪些性能隐患，该如何解决。

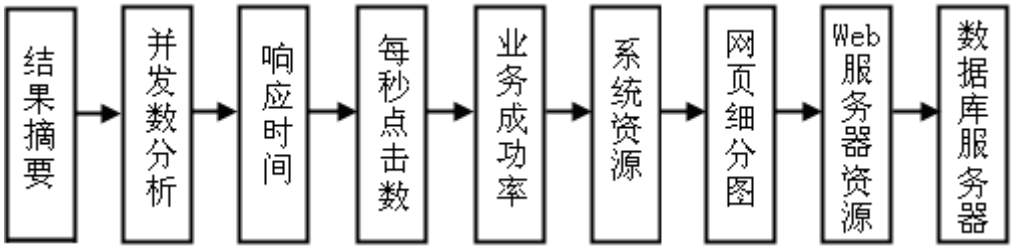
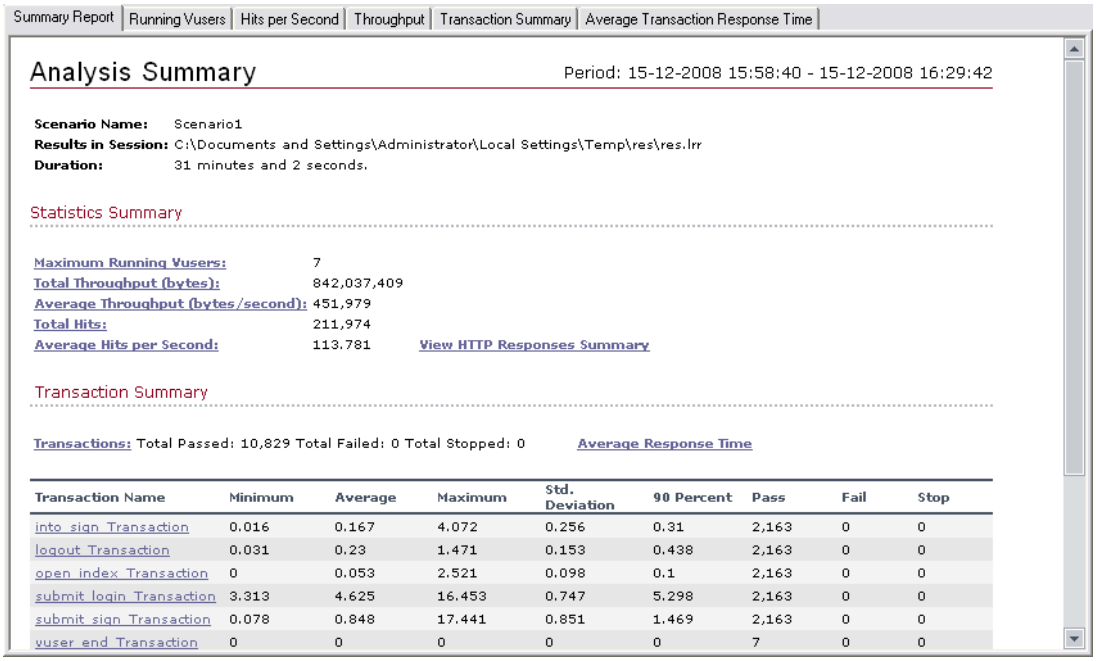


图 5- 1 性能测试结果分析流程图

结果摘要

LoadRunner 进行场景测试结果收集后，首先显示的该结果的一个摘要信息，如图 5- 2 所示。概要中列出了场景执行情况、“Statistics Summary（统计信息摘要）”、“Transaction Summary（事务摘要）”以及“HTTP Responses Summary（HTTP 响应摘要）”等。以简要的信息列出本次测试结果。



Summary Report | Running Vusers | Hits per Second | Throughput | Transaction Summary | Average Transaction Response Time

Analysis Summary　　Period: 15-12-2008 15:58:40 - 15-12-2008 16:29:42

Scenario Name: Scenario1
Results in Session: C:\Documents and Settings\Administrator\Local Settings\Temp\res\res.lrr
Duration: 31 minutes and 2 seconds.

Statistics Summary

Maximum Running Vusers: 7
Total Throughput (bytes): 842,037,409
Average Throughput (bytes/second): 451,979
Total Hits: 211,974
Average Hits per Second: 113.781　　View HTTP Responses Summary

Transaction Summary

Transactions: Total Passed: 10,829 Total Failed: 0 Total Stopped: 0　　Average Response Time

Transaction Name	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
into_sign_Transaction	0.016	0.167	4.072	0.256	0.31	2,163	0	0
logout_Transaction	0.031	0.23	1.471	0.153	0.438	2,163	0	0
open_index_Transaction	0	0.053	2.521	0.098	0.1	2,163	0	0
submit_login_Transaction	3.313	4.625	16.453	0.747	5.298	2,163	0	0
submit_sign_Transaction	0.078	0.848	17.441	0.851	1.469	2,163	0	0
vuser_end_Transaction	0	0	0	0	0	7	0	0

图 5- 2 性能测试结果摘要图

场景执行情况

该部分给出了本次测试场景的名称、结果存放路径及场景的持续时间，如图 5- 3 所示。从该图我们知道，本次测试从 15:58:40 开始，到 16:29:42 结束，共历时 31 分 2 秒。与我们场景执行计划中设计的时间基本吻合。

Analysis Summary　　Period: 15-12-2008 15:58:40 - 15-12-2008 16:29:42

Scenario Name: Scenario1
Results in Session: C:\Documents and Settings\Administrator\Local Settings\Temp\res\res.lrr
Duration: 31 minutes and 2 seconds.

图 5- 3 场景执行情况描述图

Statistics Summary（统计信息摘要）

该部分给出了场景执行结束后并发数、总吞吐量、平均每秒吞吐量、总请求数、平均每秒请求数的统计值，如图 5- 4 所示。从该图我们得知，本次测试运行的最大并发数为 7，总吞吐量为 842,037,409 字节，平均每秒的吞吐量为 451,979 字节，总的请求数为 211,974，平均每秒的请求为 113.781，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。

Statistics Summary

Maximum Running Vusers:	7	
Total Throughput (bytes):	842,037,409	
Average Throughput (bytes/second):	451,979	
Total Hits:	211,974	
Average Hits per Second:	113.781	View HTTP Responses Summary

图 5- 4 统计信息摘要图

Transaction Summary（事务摘要）

该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，如图 5- 5 所示。从该图我们得到每个 Action 的平均响应时间与业务成功率。

> 注意:
>
> 因为在场景的“Run-time Settings”的“Miscellaneous”选项中将每一个 Action 当成了一个事务执行，故这里的事务其实就是脚本中的 Action。

Transaction Summary

Transactions: Total Passed: 10,829 Total Failed: 0 Total Stopped: 0　　Average Response Time

Transaction Name	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
into_sign_Transaction	0.016	0.167	4.072	0.256	0.31	2,163	0	0
logout_Transaction	0.031	0.23	1.471	0.153	0.438	2,163	0	0
open_index_Transaction	0	0.053	2.521	0.098	0.1	2,163	0	0
submit_login_Transaction	3.313	4.625	16.453	0.747	5.298	2,163	0	0
submit_sign_Transaction	0.078	0.848	17.441	0.851	1.469	2,163	0	0
vuser_end_Transaction	0	0	0	0	0	7	0	0
vuser_init_Transaction	0	0	0.001	0	0.001	7	0	0

图 5- 5 事务摘要图

HTTP Responses Summary（HTTP 响应摘要）

该部分显示在场景执行过程中，每次 HTTP 请求发出去的状态，是成功还是失败，都在这里体现，如图 5- 6 所示。从图中可以看到，在本次测试过程中 LoadRunner 共模拟发出了 211974 次请求（与“统计信息摘要”中的“Total Hits”一致），其中“HTTP 200”的是 209811 次，而“HTTP 404”则有 2163，说明在本次过程中，经过发出的请求大部分都能正确响应了，但还是有部分失败了，但未影响测试结果，“HTTP 200”表示请求被正确响应，而“HTTP 404”表示文件或者目录未能找到。有朋友可能会问，这里出现了 404 的错误，为什么结果还都通过了。出现这样问题的原因是脚本有些页面的请求内容并非关键点，比如可能请求先前的 cookie 信息，如果没有就重新获取，所以不会影响最终的测试结果。

HTTP Responses Summary

HTTP Responses	Total	Per second
HTTP 200	209,811	112.62
HTTP 404	2,163	1.161

图 5- 6 HTTP 响应摘要

常用的 HTTP 状态代码如下：

400 无法解析此请求。
401.1 未经授权：访问由于凭据无效被拒绝。
401.2 未经授权: 访问由于服务器配置倾向使用替代身份验证方法而被拒绝。
401.3 未经授权：访问由于 ACL 对所请求资源的设置被拒绝。
401.4 未经授权：Web 服务器上安装的筛选器授权失败。
401.5 未经授权：ISAPI/CGI 应用程序授权失败。
401.7 未经授权：由于 Web 服务器上的 URL 授权策略而拒绝访问。
403 禁止访问：访问被拒绝。
403.1 禁止访问：执行访问被拒绝。
403.2 禁止访问：读取访问被拒绝。
403.3 禁止访问：写入访问被拒绝。
403.4 禁止访问：需要使用 SSL 查看该资源。
403.5 禁止访问：需要使用 SSL 128 查看该资源。
403.6 禁止访问：客户端的 IP 地址被拒绝。
403.7 禁止访问：需要 SSL 客户端证书。
403.8 禁止访问：客户端的 DNS 名称被拒绝。
403.9 禁止访问：太多客户端试图连接到 Web 服务器。
403.10 禁止访问：Web 服务器配置为拒绝执行访问。
403.11 禁止访问：密码已更改。
403.12 禁止访问：服务器证书映射器拒绝了客户端证书访问。
403.13 禁止访问：客户端证书已在 Web 服务器上吊销。
403.14 禁止访问：在 Web 服务器上已拒绝目录列表。
403.15 禁止访问：Web 服务器已超过客户端访问许可证限制。
403.16 禁止访问：客户端证书格式错误或未被 Web 服务器信任。
403.17 禁止访问：客户端证书已经到期或者尚未生效。
403.18 禁止访问：无法在当前应用程序池中执行请求的 URL。
403.19 禁止访问：无法在该应用程序池中为客户端执行 CGI。
403.20 禁止访问：Passport 登录失败。

404 找不到文件或目录。
404.1 文件或目录未找到：网站无法在所请求的端口访问。
需要注意的是 404.1 错误只会出现在具有多个 IP 地址的计算机上。如果在特定 IP 地址/端口组合上收到客户端请求，而且没有将 IP 地址配置为在该特定的端口上侦听，则 IIS 返回 404.1 HTTP 错误。例如，如果一台计算机有两个 IP 地址，而只将其中一个 IP 地址配置为在端口 80 上侦听，则另一个 IP 地址从端口 80 收到的任何请求都将导致 IIS 返回 404.1 错误。只应在此服务级别设置该错误，因为只有当服务器上使用多个 IP 地址时才会将它返回给客户端。
404.2 文件或目录无法找到：锁定策略禁止该请求。
404.3 文件或目录无法找到：MIME 映射策略禁止该请求。
405 用于访问该页的 HTTP 动作未被许可。
406 客户端浏览器不接受所请求页面的 MIME 类型。
407 Web 服务器需要初始的代理验证。
410 文件已删除。
412 客户端设置的前提条件在 Web 服务器上评估时失败。
414 请求 URL 太大，因此在 Web 服务器上不接受该 URL。
500 服务器内部错误。
500.11 服务器错误：Web 服务器上的应用程序正在关闭。
500.12 服务器错误：Web 服务器上的应用程序正在重新启动。
500.13 服务器错误：Web 服务器太忙。
500.14 服务器错误：服务器上的无效应用程序配置。
500.15 服务器错误：不允许直接请求 GLOBAL.ASA。
500.16 服务器错误：UNC 授权凭据不正确。
500.17 服务器错误：URL 授权存储无法找到。
500.18 服务器错误：URL 授权存储无法打开。
500.19 服务器错误：该文件的数据在配置数据库中配置不正确。
500.20 服务器错误：URL 授权域无法找到。
500 100 内部服务器错误：ASP 错误。
501 标题值指定的配置没有执行。
502 Web 服务器作为网关或代理服务器时收到无效的响应。

并发数分析

“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。图 5-7 显示了在 OA 系统考勤业务性能测试过程中 Vusers 运行情况，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 5- 7 的表现，可以认为参数化是没有问题的。

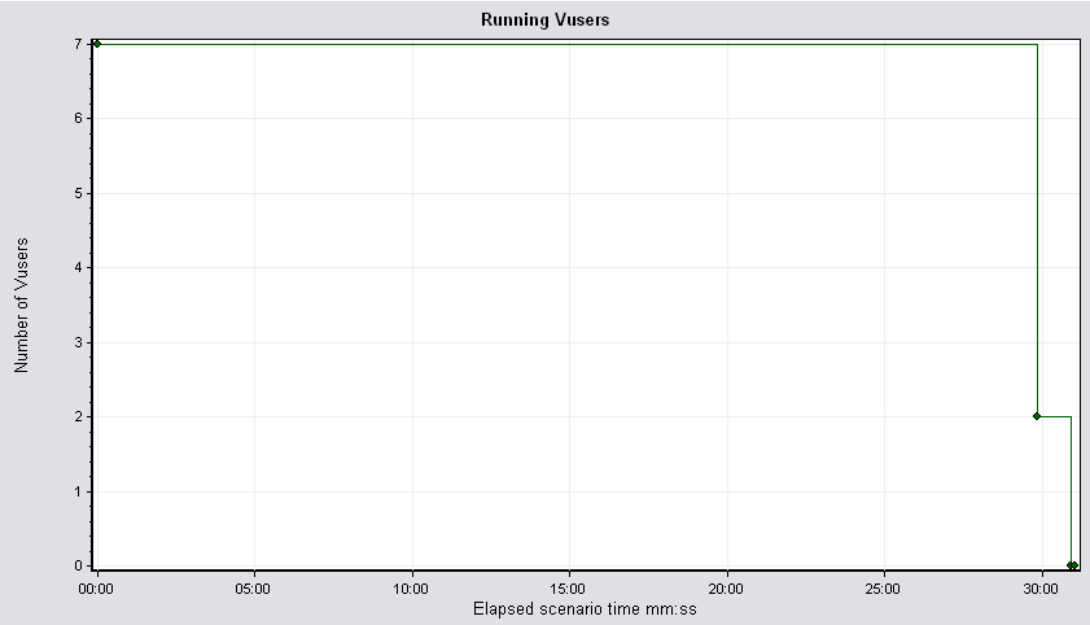


图 5- 7 运行的并发数图

测试脚本中我们还使用了集合点，那么这里还可以看看集合点在场景执行过程中的表现，点击左边的“New Graph”，出现图 5- 8，展开“Vusers”前的加号，双击“Rendezvous”，出现集合点的图形后，点击【Close】，关闭添加新图界面。

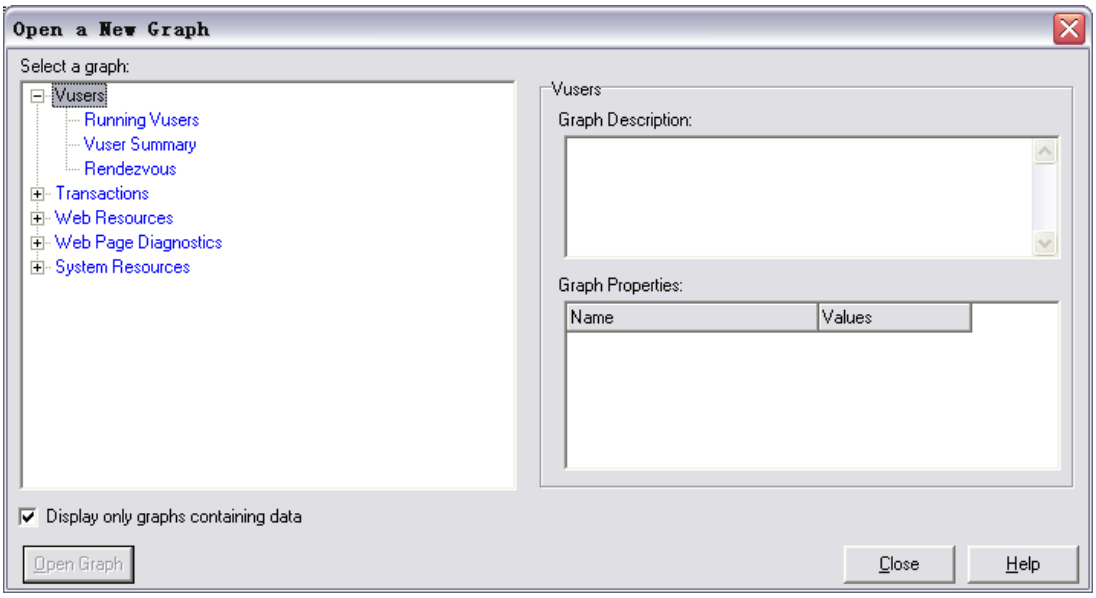


图 5- 8 添加集合点统计图

集合点的图形如图 5- 9 所示，从图中可以看到，所有用户到达集合点后，立刻就释放了。与之前设定的集合点策略设置“所有运行用户到达后释放“是一致的。假设这样的一种情况，Running 的 Vusers 有 10 个，集合点策略设置是“所有运行用户到达后释放”，而集合点图形显示的最大释放 Vusers 是 7 个，那么就表示有些 Vuser 超时了，引起超时的原因可能是 Vuser 得到的响应超时了，可以结合平均事务响应时间再详细分析原因。

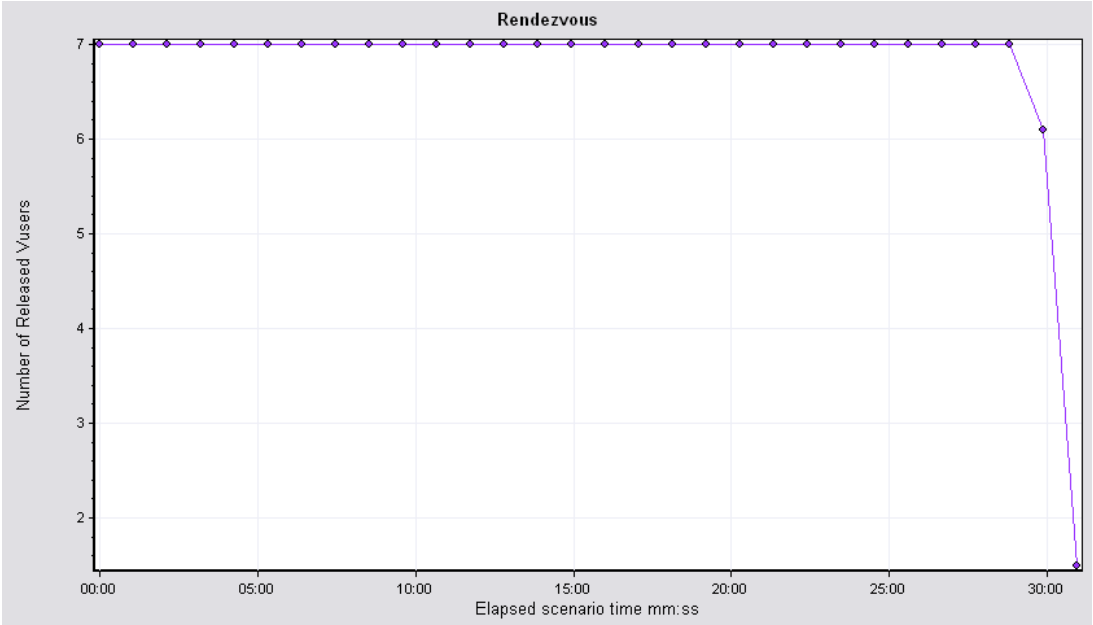
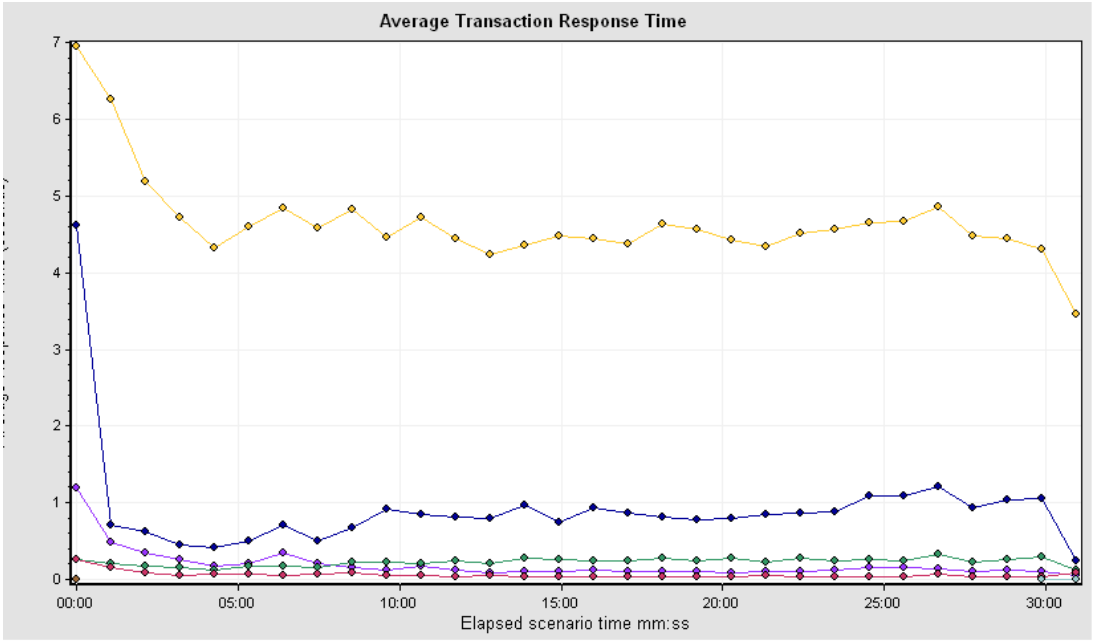


图 5- 9 集合点状态图

我们本次测试 Running Vusers 与集合点是一致，说明整个场景执行过程中，并发数用户的执行正确，OA 系统测试服务器能够应付 7 个并发用户的业务操作。

响应时间

在性能测试要求中我们知道，有一项指标是要求登录、考勤业务操作的页面响应时间不超过 3 秒，那么本次测试是否达到了这个要求呢？我们先来看“Average Transaction Response Time（平均事务响应时间图）”（图 5- 10），这张图是平均事务响应时间与结果摘要中的“Transaction Summary”合成的。



Transaction Name	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
into_sign_Transaction	0.016	0.167	4.072	0.256	0.31	2,163	0	0
logout_Transaction	0.031	0.23	1.471	0.153	0.438	2,163	0	0
open_index_Transaction	0	0.053	2.521	0.098	0.1	2,163	0	0
submit_login_Transaction	3.313	4.625	16.453	0.747	5.298	2,163	0	0
submit_sign_Transaction	0.078	0.848	17.441	0.851	1.469	2,163	0	0
vuser_end_Transaction	0	0	0	0	0	7	0	0
vuser_init_Transaction	0	0	0.001	0	0.001	7	0	0

图 5- 10 平均事务响应时间图

从图形下部我们可以看到，登录部分对应的 Action 是“submit_login”，考勤业务提交对应的 Action 是“submit_sign”，他们的“Average Time（平均响应时间为）”分别是 4.425 秒与 0.848 秒，从这两个数值来看，考勤业务的事务响应时间 0.848 秒小于预期的 3 秒，达到了要求，而登录是 4.425 秒，大于预期的 3 秒，不符合要求。这样的结果是不正确的，因为在统计的登录业务的时候，我们没有去除思考时间，所以，登录功能的实际事务时间应该是 4.425 秒-3 秒=1.425 秒，小于预期的 3 秒，故登录业务的事务响应时间也达到了我们的要求。在平时的性能测试活动中，统计结果的时候需要去掉思考时间，加上思考时间是为了真实的模拟用户环境，统计结果中除去思考时间是为了更真实的反映服务器的处理能力，两者并不矛盾。

看完了“Average Time”，我们再看“90 Percent Time”，这个时间从某种程度来说，更准确衡量了测试过程中各个事务的真实情况，表示 90%的事务，服务器的响应都维持在某个值附近，“Average Time”值对于平均事务响应时间变动趋势很大的情况统计就不准确了，比如有三个时间：1 秒、5 秒、12 秒，则平均时间为 6 秒，而另外一种情况：5 秒、6 秒、7 秒，平均时间也为 6 秒，显然第二种比第一种要稳定多了。所以，我们在查看平均事

务响应时间的时候，先看整体曲线走势，如果整体趋势比较平滑，没有忽上忽下的波动情况，取“Average Time”与“90 Percent Time”都可以，如果整体趋势毫无规律，波动非常大，我们就不用“Average Time”而使用“90 Percent Time”可能更真实些。

从图 5- 10 可以看出，所有 Action 平均事务响应时间的趋势都非常平滑，所以使用“Average Time”与“90 Percent Time”差别不是很大，用哪个都可以。这里是使用最常用的统计方法“90 Percent Time”。登录业务的“90 Percent Time”是 5.298 秒-3 秒（思考时间）=2.298 秒，考勤业务的“90 Percent Time”是 1.469 秒，没有思考时间，那么就是实打实的啦。根据上面的计算，本次测试结果记录如表 5- 1 所示。

测试项	目标值	实际值	是否通过
登录业务响应时间	<=3 秒	2.298 秒	Y
考勤业务响应时间	<=3 秒	1.469 秒	Y
登录业务成功率	100%		
考勤业务成功率	100%		
登录业务总数	30 分钟完成 2000		
考勤业务总数	30 分钟完成 2000		
CPU 使用率	<75%		
内存使用率	<70%		

表 5- 1 测试结果对照表一

每秒点击数

“Hits per Second（每秒点击数）”反映了客户端每秒钟向服务器端提交的请求数量，如果客户端发出的请求数量越多，与之相对的“Average Throughput (bytes/second)”也应该越大，并且发出的请求越多会对平均事务响应时间造成影响，所以在测试过程中往往将这三者结合起来分析。图 5- 11 显示的是“Hits per Second”与“Average Throughput (bytes/second)”的复合图，从图中可以看出，两种图形的曲线都正常并且基本一致，说明服务器能及时的接受客户端的请求，并能够返回结果。如果“Hits per Second”正常，而“Average Throughput (bytes/second)”不正常，则表示服务器虽然能够接受服务器的请求，但返回结果较慢，可能是程序处理缓慢。如果“Hits per Second”不正常，则说明客户端存在问题，那种问题一般是网络引起的，或者录制的脚本有问题，未能正确的模拟用户的行为。具体问题具体分析，这里仅给出一些建议。

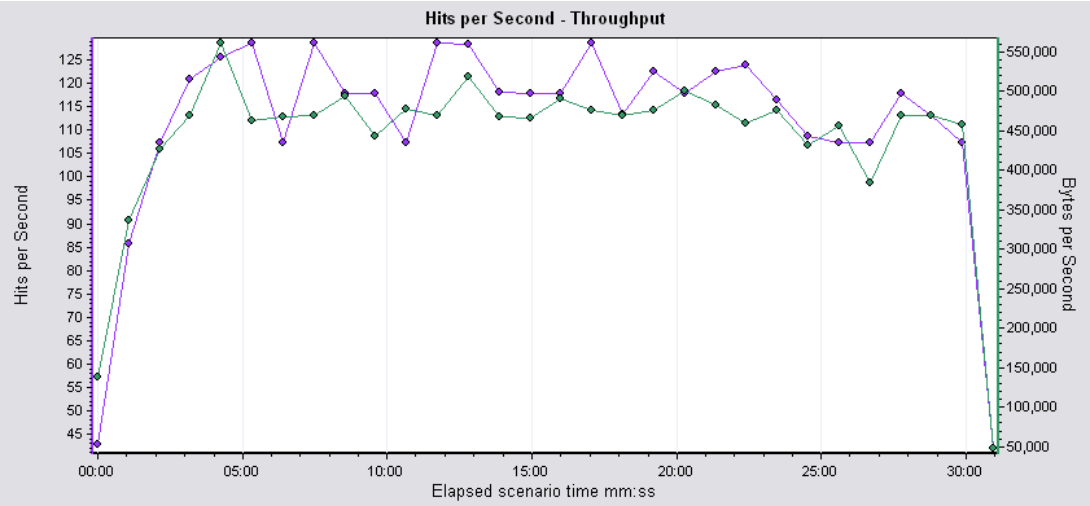


图 5- 11 每秒点击数与每秒吞吐量复合图

对于本次测试来说，“Hits per Second”与“Average Throughput (bytes/second)”都是正常的，而且整体表现还是不错的。

一般情况下，这两种指标用于性能调优，比如给定了几个条件，去检测另外一个条件，用这两个指标衡量，往往起到很好的效果。比如要比较某两种硬件平台的优劣，就可以使用相同的配置方法部署软件系统，然后使用相同的脚本、场景设计、统计方法去分析，最终得出一个较优的配置。

业务成功率

“业务成功率”这个指标在很多系统中都提及到，比如电信的、金融的、企业资源管理的等等。举个例子，我们楼下的建行，假如每天的业务类别是这样的：20 个开户，5 个销户，300 个存款，500 取款，100 个汇款等，那么在做他们的营业系统测试时就需要考虑业务成功率了，一般不得低于 98%。具体的业务成功率是什么意思呢？

排除那些复杂的业务，比如异步处理的业务（移动的套卡开通就是异步的），业务成功率就是事务成功率，用户一般把一个 Aciton 当做一笔业务，在 LoadRunner 场景执行中一笔交易称为一个事务。所以，说业务成功率其实就是事务成功率、通过率的意思。在“Transaction Summary”中我们可以很明确的看到每个事务的执行状态，如图 5- 12 所示。

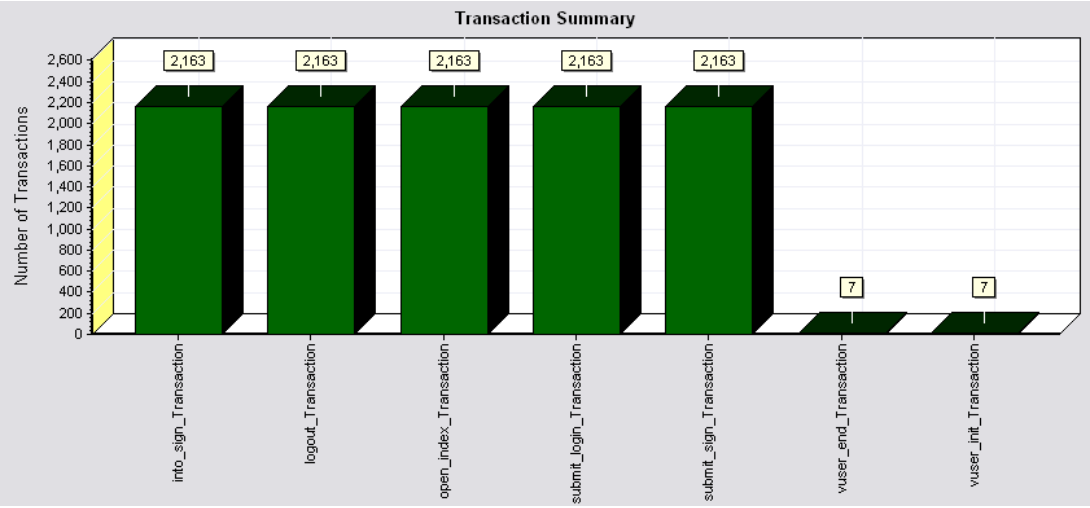


图 5- 12 事务状态统计图

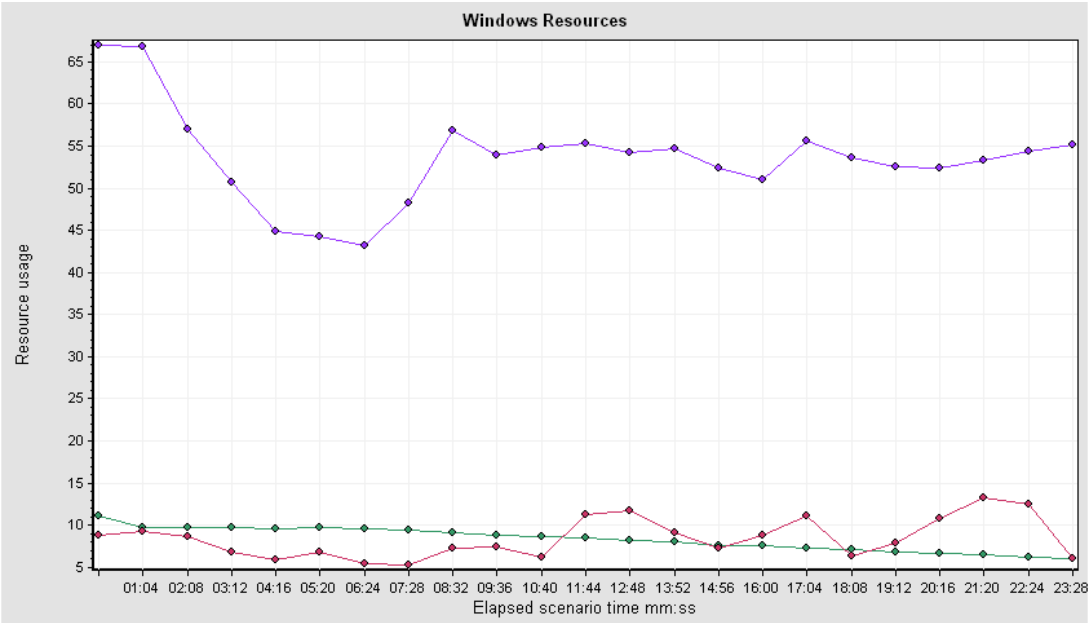
从图中可以看出，所有的 Aciton 都是绿色的，即表示为 Passed，同时除了 vuser_init 与 vuser_end 两个事务，其他的事务通过数为 2163，也就表明在 30 分钟的时间里，共完成了 2163 次登录考勤业务操作。那么根据这些可以判断本次测试登录业务与考勤业务的成功率是 100%，再次更新测试结果记录表如表 5- 2 所示。

测试项	目标值	实际值	是否通过
登录业务响应时间	<=3 秒	2.298 秒	Y
考勤业务响应时间	<=3 秒	1.469 秒	Y
登录业务成功率	100%	100%	Y
考勤业务成功率	100%	100%	Y
登录业务总数	30 分钟完成 2000	2163	Y
考勤业务总数	30 分钟完成 2000	2163	Y
CPU 使用率	<75%		
内存使用率	<70%		

表 5- 2 测试结果对照表二

系统资源

系统资源图显示了在场景执行过程中被监控的机器系统资源使用情况，一般情况下监控机器的 CPU、内存、网络、磁盘等各个方面。本次测试监控的是测试服务器的 CPU 使用率与内存使用率，以及处理器队列长度，具体的数据如图 5- 13 所示。



Color	Scale	Measurement	Minimum	Average	Maximum	Std. Deviation
☑	1	% Processor Time (Processor _Total):192.168....	3.797	53.582	100	25.875
☑	0.1	Available MBytes (Memory):192.168.0.52	51	83.456	141	14.079
☑	1	Processor Queue Length (System):192.168.0.52	0	8.45	40	9.684

图 5- 13 测试服务器系统资源监控结果图

从图中可以看出，CPU 使用率、可用物理内存、CPU 的队列长度三个指标的曲线逗较为平滑，三者的平均值分别为：53.582%、83.456M、8.45，而测试服务器总的物理内存为 384M，那么内存使用率为（384-83.456）/384=78.26%，根据本次性能测试要求的：CPU 使用率不超过 75%，物理内存使用率不超过 70%这两点来看，内存的使用率 78.26%大于预期的 70%，故内存使用率不达标。根据 Windwos 资源性能指标的解释，一般情况下，如果“Processor Queue Length（处理器队列长度）”一直超过二，则可能表示处理器堵塞，我们这里监控出来的数值是 8.45，而且总体上保持平衡，那么由此推断，测试服务器的 CPU 也可能是个瓶颈。同时在测试过程中，场景执行到 23 分半钟的时候，报出了**错误！未找到引用源。**的错误，意思是说被监控的服务器当前无法再进行计数器数据的获取了，所以，本次操作系统资源的监控只得到了场景执行的前 23 分半钟的数据。这样对本次测试结果有一定的影响。

获得上述数据后，最新的测试结果记录表如表 5- 3 所示。

测试项	目标值	实际值	是否通过
登录业务响应时间	<=3 秒	2.298 秒	Y
考勤业务响应时间	<=3 秒	1.469 秒	Y
登录业务成功率	100%	100%	Y
考勤业务成功率	100%	100%	Y
登录业务总数	30 分钟完成 2000	2163	Y
考勤业务总数	30 分钟完成 2000	2163	Y
CPU 使用率	<75%	53.582%	Y

内存使用率	<70%	78.26%	N

表 5- 3 测试结果对照表三

从上表数据来看，本次测试总体上已经达到了预期的性能指标，但从其他的数据，比如 CPU 的队列长度、内存使用率来看，被测服务器的硬件资源需要提升。

网页细分图

网页细分图可以评估页面内容是否影响事务响应时间。使用网页细分图，可以分析网站上有问题的元素（例如下载很慢的图像或打不开的链接）。

我们这里查看一下网页细分图中的“Page Download Time Breakdown”，点击**错误！未找到引用源。**左边的“New Graph”，出现图 5- 14，展开“Web Page Diagnostics”前的加号，双击“Page Download Time Breakdown”，待出现“Page Download Time Breakdown”监控图后，点击【Close】按钮关闭添加监控图界面。

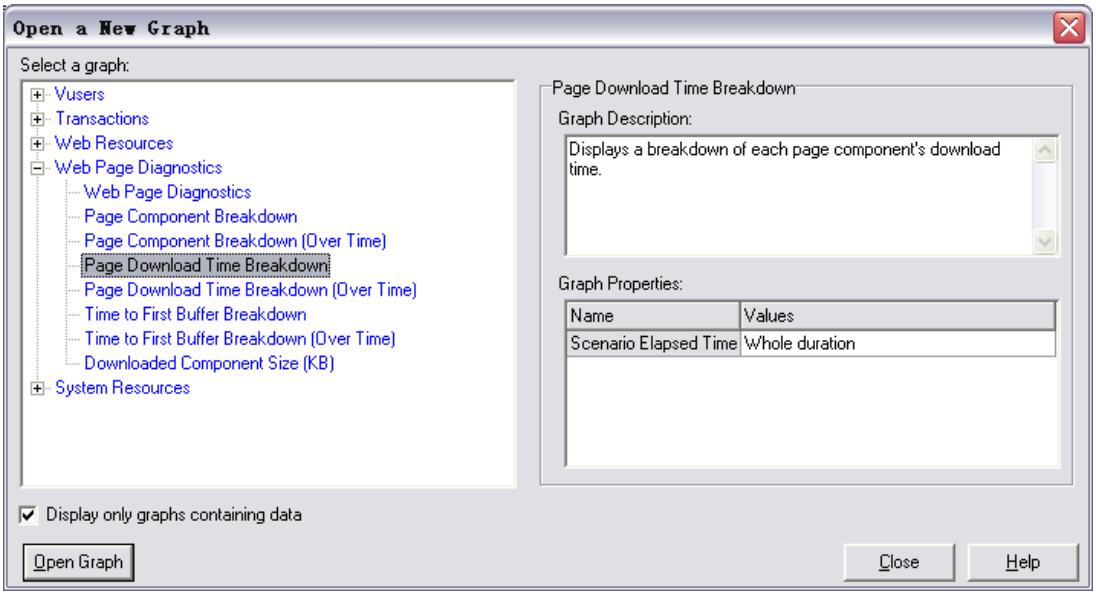


图 5- 14 添加网页细分图

在监控图列表中，我们看到图 5- 15，从图中我们看到，在所有的页面中，登录后的用个人面页面“http://192.168.0.52:8080/oa/oa.jsp”的下载时间最长。

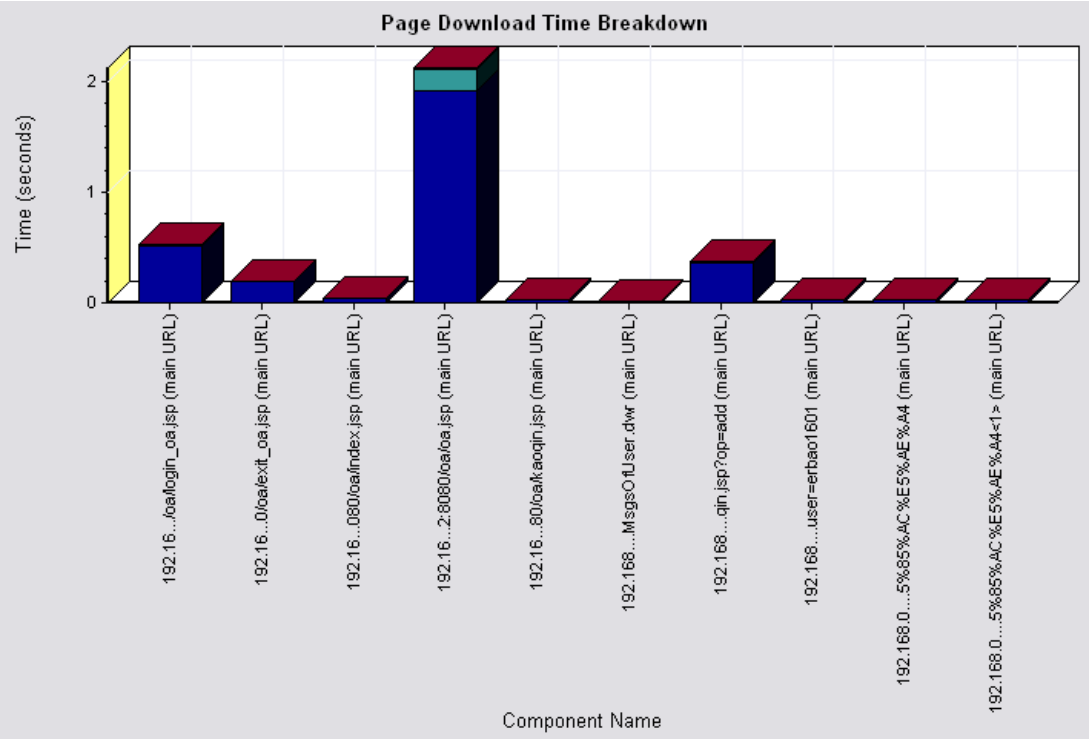


图 5- 15 网页下载时间细分图

图 5- 16 详细列出了每个页面所消耗的时间分布，图中每一个指标含义见表 5- 4 所示。该表由 LoadRunner 使用手册提供。通过这些指标的数据，我们可以轻易的判断是哪个页面、哪个请求导致了响应时间变长，甚至响应失败。

Color	Scale	Measurement	Minimum	Average	Maximum	Std. Deviation
☑	1	Client Time	0	0.001	2.424	0.029
☑	1	Connection Time	0	0	0.125	0.003
☑	1	DNS Resolution Time	0	0	0	0
☑	1	Error Time	0	0	0	0
☑	1	First Buffer Time	0	0.15	18.929	0.5
☑	1	FTP Authentication Time	0	0	0	0
☑	1	Receive Time	0	0.01	4.92	0.091
☑	1	SSL Handshaking Time	0	0	0	0

图 5- 16 oa.jsp 页面下载时间分布图

名称	描述
Client Time	显示因浏览器思考时间或其他与客户端有关的延迟而使客户机上的请求发生延迟时，所经过的平均时间。
Connection Time	显示与包含指定 URL 的 Web 服务器建立初始连接所需的时间。连接度量是一个很好的网络问题指示器。此外，它还可表明服务器是否对请求做出响应。
DNS Resolution Time	显示使用最近的 DNS 服务器将 DNS 名称解析为 IP 地址所需的时间。DNS 查找度量是指示 DNS 解析问题或 DNS 服务器问题的一个很好的指示器。
Error Time	显示从发出 HTTP 请求到返回错误消息（仅限于 HTTP 错误）这期间经过的平均时间。
First Buffer Time	显示从初始 HTTP 请求（通常为 GET）到成功收回来自 Web 服务器的第一次

	缓冲时为止所经过的时间。第一次缓冲度量是很好的 Web 服务器延迟和网络滞后指示器。 （注意：由于缓冲区大小最大为 8K，因此第一次缓冲时间可能也就是完成元素下载所需的时间。）
FTP Autherntication Time	显示验证客户端所用的时间。如果使用 FTP，则服务器在开始处理客户端命令之前，必须验证该客户端。FTP 验证度量仅适用于 FTP 协议通信
Receive Time	显示从服务器收到最后一个字节并完成下载之前经过的时间。接收度量是很好的网络质量指示器（查看用来计算接收速率的时间 / 大小比率）。
SSL Handshaking Time	显示建立 SSL 连接（包括客户端 hello、服务器 hello、客户端公用密钥传输、服务器证书传输和其他部分可选阶段）所用的时间。此时刻后，客户端和服务器之间的所有通信都被加密。SSL 握手度量仅适用于 HTTPS 通信。

表 5- 4 网页下载时间细分指标说明

对于本次测试，从网页细分图来看，基本上每个页面的加载时间都是预期范围内，oa.jsp 页面因为集成了用户的个人工作平台，需要检索很多的数据，并合成了很多图片，所以相应的加载时间较长，这是正确的。

Web 服务器资源

上述所有的监控图形 LoadRunner 都可以提供，但对于某些测试监控图来说，LoadRunner 就没有提供了，期望其新版支持这些功能，当然想监控 Tomcat、Jboss 或者其他的 Web 服务器可以 SiteScope 工具，这个工具配置较为复杂，根据个人需要吧。我这里监控 Tomcat 使用的是 ManageEngine Applications Manager 8 的试用版，测试结束后得出 Tomcat 的 JVM 使用率如图 5- 17 所示。

图 5- 17 Tomcat JVM 使用率监视图

从图中我们可以明显看出，Tomcat 的 JVM 使用率不断上升，配置 Tomcat 时共分配了

100M 左右的物理内存给其，测试初期使用的 JVM 相对来说较少，我们的测试场景是从 15:58:40 开始，到 16:29:42 结束，共历时 31 分 2 秒。从图中看到，从 16:00 到 16:30 这个时间内，也就是测试场景执行期间，JVM 的使用率不断上升，并没有在请求达到均衡状态后也呈现一种平衡状态，所以，从这点可以推断，如果测试场景继续执行，或者加大并发数，最终必将导致 Tomcat 内存不够用而报出“Out Of Memory”内存溢出的错误。在正常情况下，内存的使用应该与“Hit per Second”、“Average Throughput (bytes/second)”等监控图的图形走势是一致的。

从上述过程可以得出一个结论，出现图 5- 17 中的问题，可能有两个原因：

1、 Tomcat 的内存分配不足；

2、 程序代码有错误，可能导致内存泄露。

解决方法：

1、 为 Tomcat 分配更多的内存，如果是使用的 catalina.sh 或 Catalina.bat 启动的 Tomcat，则可在这两个文件中添加“SET CATALINA_OPTS= -Xms300m －Xmx300m”，如果使用的 winnt 服务方式启动的 Tomcat，则可在“运行”中输入“regedit”进入注册表，然后在“HKEY_LOCAL_MACHINE-->SOFTWARE-->Apache Software Foundation-->Process Runner 1.0-->Tomcat5-->Parameters”修改两个属性，一个是 JvmMs，另外一个是 JvmMx，如图 5- 18 所示。

2、 检查程序代码，使用一些内存泄露检查工具进行清查。

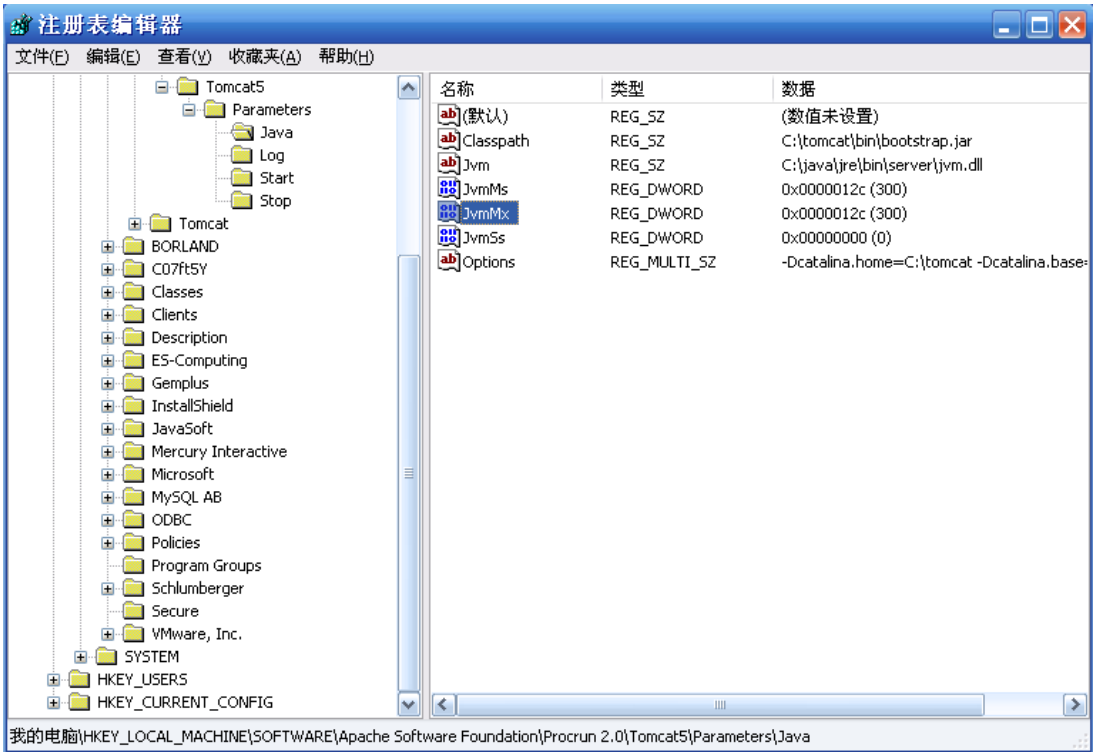


图 5- 18 修改 Tocat 的 JVM 数据

数据库服务器资源

数据库服务器资源监控相对来说就复杂的多了，现在常用的数据有 Mysql、SQL Server、Oracle、DB2 等，LoadRunner 提供对后面几种数据库的监控方法，但对 Mysql 没有提供对应的监控方法。他不提供，咱们就自己找监控工具，我这里使用的是 Spotlight，该工具监控数据库的好处是配置连接简单，不仅能监控数据库，还能监控操作系统的资源，监控结果直观明了。**错误！未找到引用源。**显示了 Mysql 数据库在场景执行过程中 SQL 语句的执行情况，从图中可以看到，“Selects（查询）”与“Inserts（插入）”两种语句执行的趋势在场景执行过程中是比较平滑，并且测试中没有错误发现，也就说明在处理相关业务时 Mysql 的处理是正常的。假如这两种 SQL 语句任何一个出现波动很大的情况，就可以推出在场景执行过程中存在页面错误，因为这些语句不执行，就表明某些页面未被加载或者某些功能未被使用。在本次测试中，OA 系统的“oa.jsp”页面有大量的“Selects（查询）”语句，而考勤操作则是“Inserts（插入）”，所以，只要有一方出问题，必然表示测试过程中存在页面打不开或者考勤不成功的错误。

通过前面的分析，在查看**错误！未找到引用源。**中的 SQL 语句执行状态，本次测试在页面访问、功能执行方面是没有问题的。