

软件测试教程 性能测试Loadrunner篇(二)

在上一节中讲述了如何编写一个测试脚本，这节课将讲述如何设计场景、执行场景并进行结果分析。

创建场景

分析以及监控场景

利用Analysis 分析结果

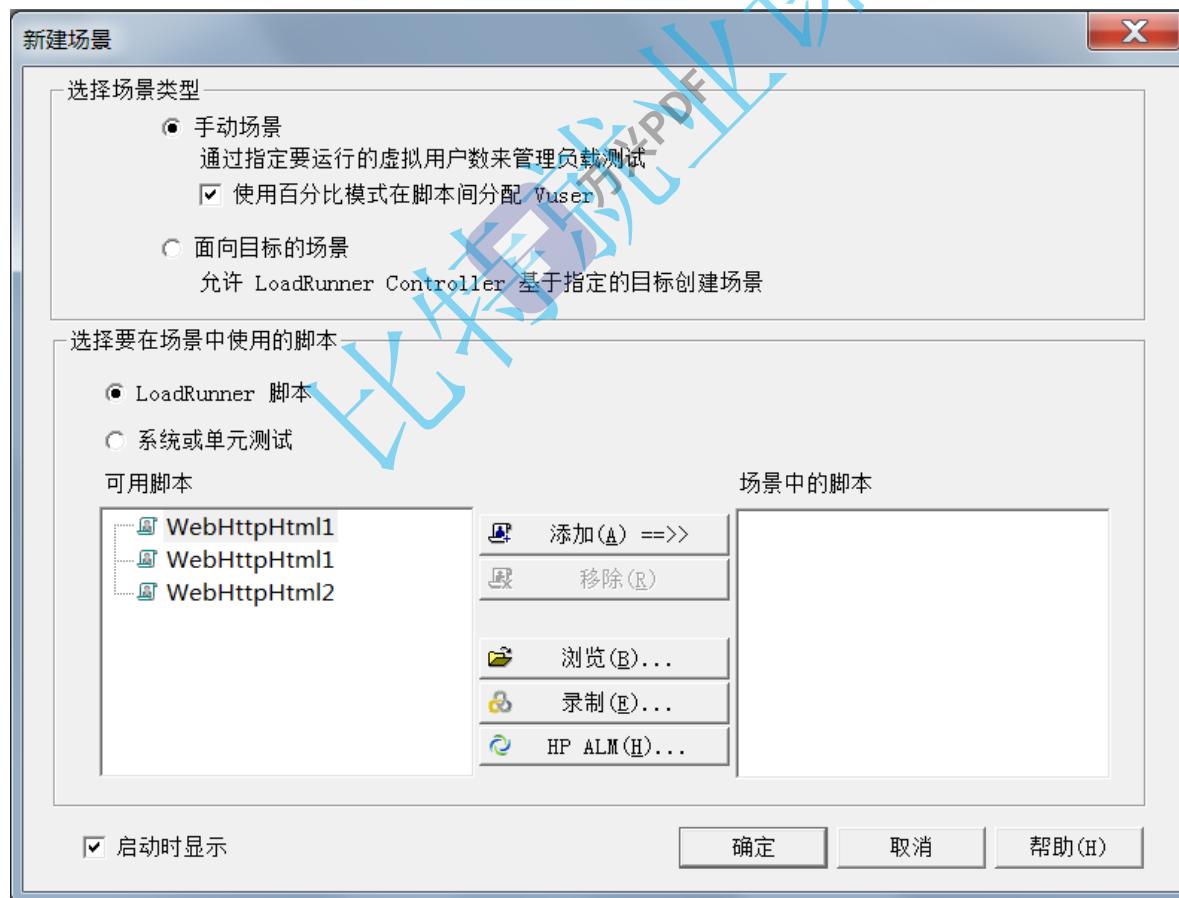
性能测试报告的编写

创建运行场景

运行场景描述在测试活动中发生的各种事件。一个运行场景包括一个运行虚拟用户活动的Load Generator 机器列表，一个测试脚本的列表以及大量的虚拟用户和虚拟用户组。

创建运行场景使用 Controller。

打开controller，出现如下页面：



在新建场景的窗口，选择一种场景类型。下面对三种类型进行简单的说明。

1、手动场景：该项要完全手动的设置场景。

1.1 使用百分比模式在脚本间分配Vuser：该项只有在“手动场景”选中的情况下才能选择。选择该项后，在场景中我们需要定义要使用的虚拟用户的总数，然后我们为每一个脚本分配要运行的虚拟用户的百分比。如果不按百分比的话每一个脚本分配的用户按数量分配。

2、面向目标的场景：在测试计划中，一般都包括性能测试要达到的目标。选择该项后，基于这个目标，自动为你创建一个场景。在场景中，我们只要定义好我们的目标即可。

选择场景类型为手动场景

选择可用脚本并确定之后，就进入手动场景设置页面，下面对设置页面进行说明

Load Generator

Load Generator是执行场景的机器，默认安装的是本机，如果在其他电脑中安装了loadrunner agent，那么也可以进行调用。

添加 LoadGenerator 后，执行“连接”操作，使状态为“就绪”，表示该机器联接正常，如果为Failed，表示该机器不能联接，请检查原因。

Vuser组模式

在菜单--场景中，可以将场景转化为Vuser组模式或者百分比模式。在Vuser组模式下，可以按照Vuser数目来设置每一个脚本

百分比模式

在百分比模式下，用户数在全局计划中进行设置，然后按照百分比进行分配

设置全局计划

这里的设置是非常重要的，也是三种场景类型最重要的区别之处。

全局计划		
	操作	属性
	初始化	在每个 Vuser 运行之前将其初始化
▶	启动 Vuser	启动 2 (个) Vuser: 每隔 00:00:15 (时:分:秒) 启动 2 个
	持续时间	运行00:05:00 (时:分:秒)
	停止 Vuser	停止 全部 (个) Vuser: 每 00:00:30 (时:分:秒) 停止 5 个
*		

初始化：说明每一个Vuser在什么时候进行初始化，Vuser只有初始化之后才能运行，一般选择默认的在每个Vuser运行之前将其初始化即可。

启动Vuser：说明Vuser在什么时候启动，同时启动Vuser会对服务器造成更大的压力，一般是按照阶梯化启动

持续时间：说明本场景共执行多长时间。“完成前一直运行”指的是所有虚拟用户只运行一遍脚本就停止。

停止Vuser：说明Vuser的停止策略。

设置结果文件保存路径

通过菜单结果--结果设置，设置结果文件的保存路径，该路径最好在每次场景运行前重新设置一下。

运行时设置

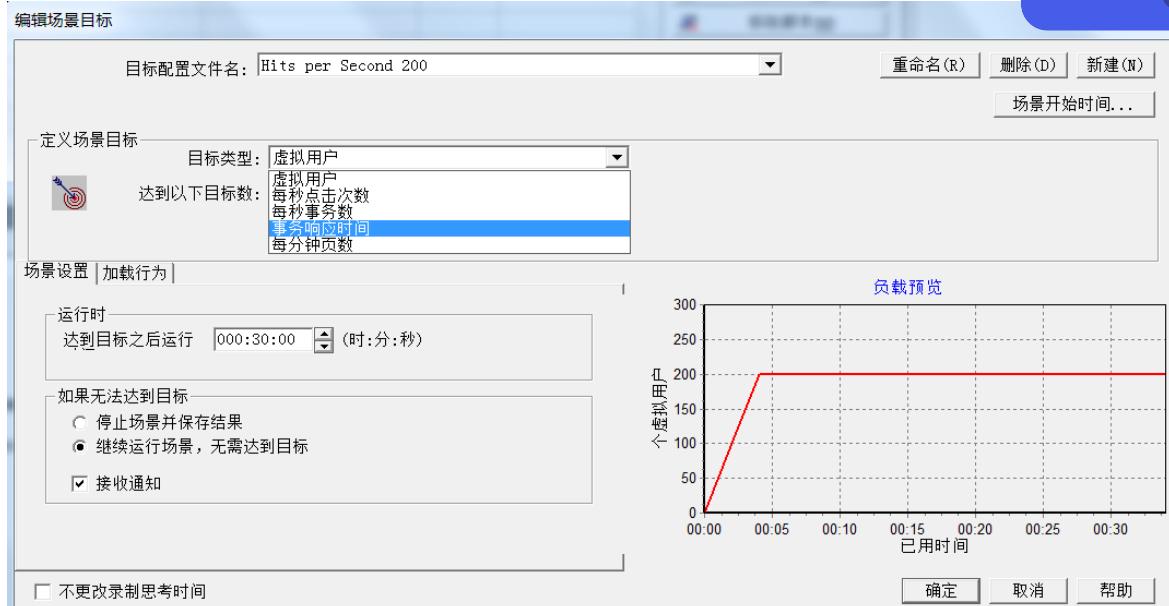
与上一节的上述的运行时设置一样。

注意会影响性能的设置项：日志和思考时间，如果想获得更高的tps，这两项都需要关闭

会影响tps统计的项：常规--其他--自动事务选项。

选择场景类型为面向目标的场景

场景目标的主要设置页面如下：



虚拟用户：

如果需要测试多少人可以同时运行Web 应用，那么推荐定义Virtual Users Goal。

每秒点击次数： 如果想测试 Web Server 的真正实力，推荐定义目标类型为：Hits per Second、Pages per Minute 或者Transactions per Second，这些类型都需要指定一个虚拟用户的最小值和最大值的范围。

Controller 试图使用最少的虚拟用户来达到定义的目标。如果使用最少的用户，不能达到目标，Controller 增加用户数，直到定义的最大值。如果使用了最多的虚拟用户数，定义的目标还没有实现，那么需要增加最大用户数，重新执行场景。

事务响应时间：

如果想知道在多少用户并发访问网站时，事务的响应时间达到性能指标说明书中规定响应时间的最大值，那么推荐使用Transactions Response Time 类型。指定需要测试的事务的名称，虚拟用户数量的最小值和最大值，还有预先定义好的事务的响应时间。

在场景运行中，如果使用了最多的虚拟用户，还不能达到定义的最大响应时间，说明Web Server 还有能力接纳定义的虚拟用户的最多数量；如果在使用了部分虚拟用户，就达到了定义的最大的响应时间，或者LoadRunner 提示如果使用最多数量的虚拟用户时将要超过最大响应时间，那么需要重新设计或者修补应用程序，同时可能需要升级Web Server 的软硬件。

如果你定义的类型是 Pages per Minute、Hits/Transactions per Second，Controller 首先用最小用户数除以定义的目标，得到一个值，然后确定每个用户应该达到的hits/transactions或者pages per minute，然后controller 开始按照以下的策略加载用户：

- 如果选择的是自动的加载虚拟用户，LoadRunner 会首先加载50 个用户。如果定义的最大用户数小于50，LoadRunner 就会一次加载所有的虚拟用户。
- 如果选择的是在场景运行一段时间后达到目标，LoadRunner 就会尝试在定义的这段时间内达到目标，根据时间限制和计算出的每个用户的hits、transactions 或者pages，LoadRunner 确定第一批加载多少用户。
- 如果选择的是按照一定的阶段达到目标（也就是先在x 长时间内达到y pages/hits，然后再达到下一个目标），LoadRunner 计算每个用户应该达到的数字后，再确定第一批加载多少用户。

每加载一批用户后，LoadRunner 会判断是否达到这批用户的目标。如果这批用户的目标没有达到，LoadRunner 重新计算每一个用户应该达到的目标数字后，重新调整下一批加载用户的数量。默认情况下，LoadRunner 每两分钟加载一批用户。

如果 Controller 加载了最多数量的用户还没有达到预定的目标，LoadRunner 会重新计算每个用户的目标，然后同时运行最大数量的用户，尝试达到预定的目标。

如果出现以下情况，Pages per Minute、Hits/Transactions per Second类型的场景会置为失败状态：

- Controller 使用了指定的最大数量的用户，并且两次都没有达到目标
- 所有的用户运行都失败
- 没有足够的 Load Generators 机器（现有的机器已经超载运行的情况下）
- Controller 增加了几批用户后，pages per minute 或者hits/transactions per second没有增加
- Controller 加载第一批用户后，定义的目标没有被捕捉到

##分析以及监控运行场景

在运行过程中，可以监视各个服务器的运行情况（DataBase Server、Web Server 等）。监视场景通过添加性能计数器来实现。实际中更多的会使用第三方工具，例如nmon来监控linux等。这里以windows服务器为例：

在运行页面，系统资源图中选中windows资源，在windows资源图中点击右键，选择“添加度量”。添加localhost为监控的服务器。之后可以查看各个计数器的值，通过这些值可以分析系统的瓶颈在哪里。

分析实时监视图表

事务响应时间是否在可接受的时间内？哪个事务用的时间最长？

看“事务响应时间”图，可以判断每个事务完成用的时间，从而可以判断出那个事务用的时间最长，那些事务用的时间超出预定的可接受时间。

网络带宽是否足够

“吞吐量”图显示在场景运行期间的每一秒钟，从Web Server 上接受到的数据量的值。拿这个值和网络带宽比较，可以确定目前的网络带宽是否是瓶颈。如果该图的曲线随着用户数的增加，没有随着增加，而是呈比较平的直线，说明目前的网络速度不能够满足目前的系统流量。

TPS值

“每秒事务数”图可以显示TPS的值

利用Analysis 分析结果

在场景执行完毕后，可以进行分析，在Controller选择菜单结果--分析结果，会调用Analysis 进行结果分析。

菜单图--添加新图，可以选择所有结果图。

事务摘要

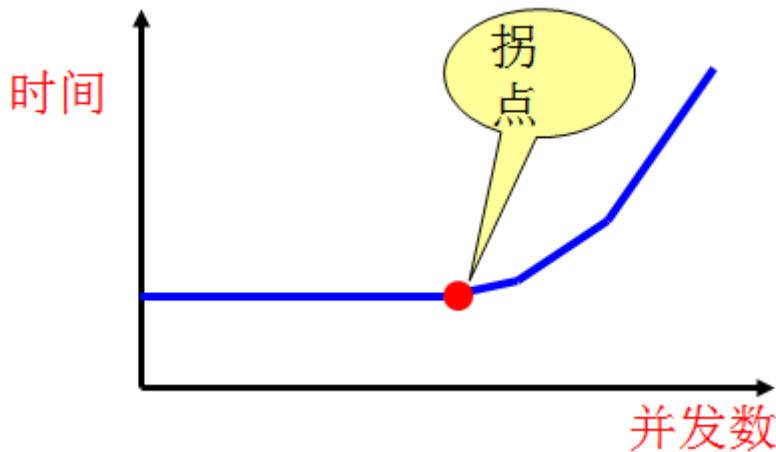
显示了整个测试期间的事务摘要，包括每一个事务的pass，fail等。对事务进行综合分析是性能分析的第一步，通过分析测试时间内用户事务的成功与失败情况，可以直接判断出系统是否运行正常。事务综合分析结束后，就可以更深入地分析用户事务的细节。

平均事务响应时间

“平均事务响应时间”图显示在测试场景运行期间的每一秒内事务执行所用的平均时间，通过它可以分析测试场景运行期间应用系统的性能走向。此外，“平均事务响应时间”图表还统计出测试场景运行时间内各个事务的最大值、最小值、平均值等信息。

通过分析平均响应时间，可以看出系统的性能走向，确定是否存在问题，如果存在问题，可以进行下一步的分析。

拐点出现后，交易将会出现排队等待，直接导致ART快速上升，此时服务器出现瓶颈，需要进一步分析。



每秒通过事务数分析图(TPS Transactions per Second)

“每秒通过事务数(TPS)”图显示在场景运行的每一秒中，每个事务通过、失败以及停止的数量，是考查系统性能的一个重要参数。通过它可以确定系统在任何给定时刻的实际事务负载。通过分析单位时间内通过的事务数，可以直接看出系统的性能变化趋势。可以将“每秒事务数(TPS)”图与平均事务响应时间图进行对比，以分析事务数目对执行时间的影响。

负载下的事务响应

“负载下的事务响应”图是“正在运行的虚拟用户”图和“平均事务响应时间”图的组合，通过它可以看出在任一时间点事务响应时间与用户数目的关系，从而掌握系统在用户并发方面的性能数据，为扩展应用系统提供参考。此图可以查看虚拟用户负载对执行时间的总体影响，对分析具有渐变负载的测试场景比较有用。

事务响应时间（百分比）

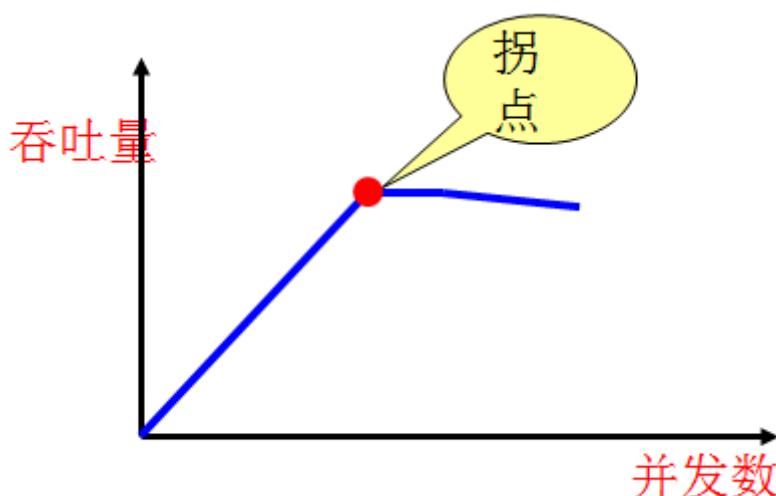
“事务响应时间（百分比）”图是根据测试结果进行分析而得到的综合分析图，也就是工具通过一些统计分析方法间接得到的图表。通过“事务响应时间（百分比）”图可以分析在给定事务响应时间范围内能够执行的事务百分比。

点击率/吞吐率

点击率的下降说明服务器的响应时间在变慢，需要对应用系统进行进一步的分析。

吞吐率和点击率的图基本上会保持一致。

拐点预示着资源耗尽或出现瓶颈，此后TPS将不再上升。同样，当点击数、TPS不再增加时，也可能出现系统瓶颈。



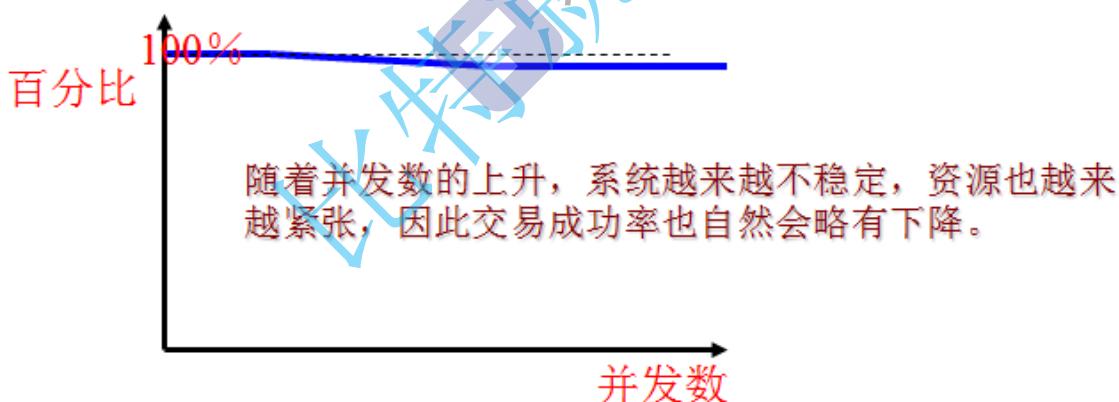
主要用来评估页面内容是否影响事务响应时间，通过它可以深入地分析网站上那些下载很久的图片，断的链接等有问题的元素。

可以按下面四种方式进行进一步的细分：下载时间细分、组件细分（随时间变化）、下载时间（随时间变化）、第一次缓冲时间（随时间变化）。通过这些细分图可以分析具体事务每个网页元素的运行情况。

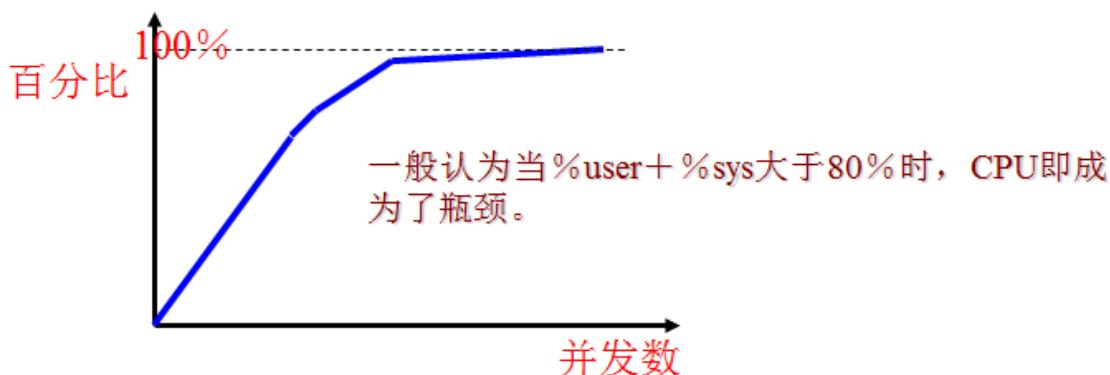
- DNS Resolution : 浏览器向 Web Server 发送请求，一般情况下，该请求首先发送到DNS Server 把DNS名字解析成IP 地址。解析的过程的时间就是DNS Resolution。。这个度量时间可以确定 DNS 服务器或者DNS 服务器的配置是否有问题。如果DNS Server 运行情况比较好，该值会比较小。
- Connection: 解析出 Web Server 的IP 地址后，请求被送到了Web Server，然后浏览器和 WebServer 之间需要建立一个初始化连接，建立该连接的过程就是Connection。这个度量时间可以简单的判断网络情况，也可以判断Web Server 是否能够响应这个请求。如果正常，该值会比较小。
- First buffer: 建立连接后，从Web Server 发出第一个数据包，经过网络传输到客户端，浏览器成功接受到第一字节的时间就是First buffer。这个度量时间不仅可以表示Web Server 的延迟时间，还可以表示出网络的反应时间。
- Receive: 从浏览器接受到第一个字节起，直到成功收到最后一个字节，下载完成止，这段时间就是Receive，这个度量时间可以判断网络的质量。
- 其他的时间还有 SSL Handshaking (SSL 握手协议)、Client Time (请求在客户端浏览器延迟的时间)、Error Time (从发送了一个HTTP 请求，到Web Server 发送回一个HTTP 错误信息所经过的时间)、Ftp Authentication (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。)

其他分析

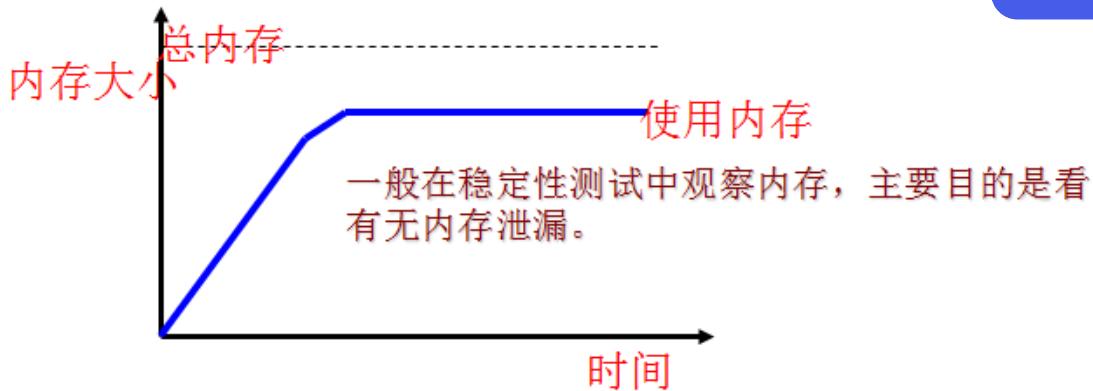
交易成功率曲线



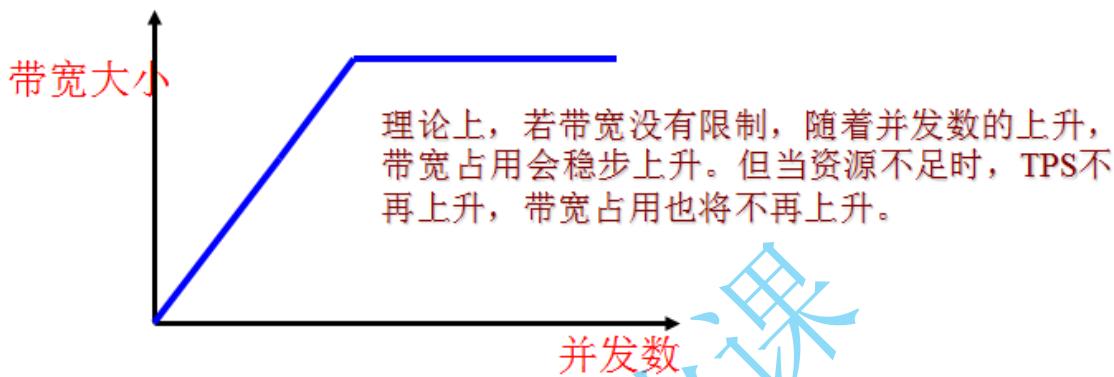
CPU使用率曲线



内存使用率曲线



网络带宽曲线



性能测试报告的编写

性能测试报告一般会包括如下部分：

- 测试目标
- 测试概要描述
- 测试结果和数据
- 测试结论

测试目标：

指标要求：本次测试预期达到的性能要求。（TPS, ART, 交易成功率, 并发数等）

测试概要描述：

系统结构

测试时间

测试地点和测试人员

工具和环境

测试过程简介

测试结果和分析：

测试场景

测试结果

结果分析

测试结论：

遗留问题

缺陷列表

测试结论

建议

测试结果的限制

比稿就用万兴PDF