

第三章 LoadRunner 场景设计

一、 场景设计

Controller 是 LR 的控制中心，包括场景设计和场景执行。

1 场景设计

- 场景设计是依据需求制定脚本如何执行的策略，使脚本运行更接近真实用户使用。
- 主要包括集合点、IP 欺骗、多机联合负载、手工场景和目标场景。

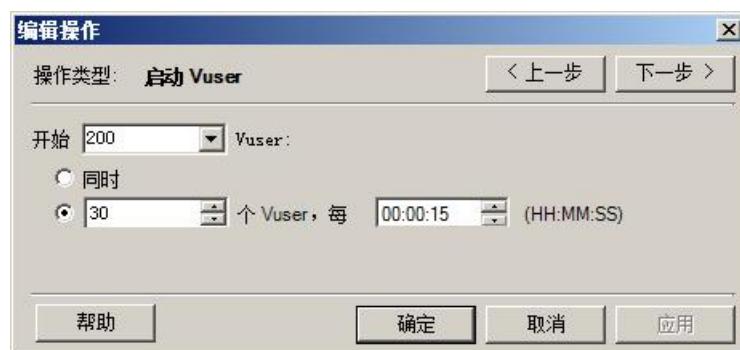
2 场景执行

- 场景执行是当场景设计完成后手动运行场景，在此过程中对场景产生的数据进行监控。
- 场景执行时会监控硬件和软件资源、统计事务的成功数、失败数以及出错信息。

3 场景状态

场景执行状态	含义
关闭(Down)	Vuser处于关闭状态
挂起(Pending)	Vuser初始化已就绪，正在等待可用的负载生成器或正在向负载生成器传输文件
初始化(Init)	Vuser正在进行初始化
就绪(Ready)	Vuser已经执行了脚本的初始化部分，可以开始运行
正在运行(Run)	Vuser正在运行。正在负载生成器上执行 Vuser 脚本
集合点(Rendezvous)	Vuser已经到达了集合点，正在等待释放
错误(Error)	Vuser发生了错误
逐步退出(Gradual Exiting)	Vuser正在运行退出前的最后一次迭代
退出(Exiting)	Vuser运行结束，正在退出
停止(Stopped)	设置“停止”命令后，Vuser 即停止

二、 虚拟用户加载策略



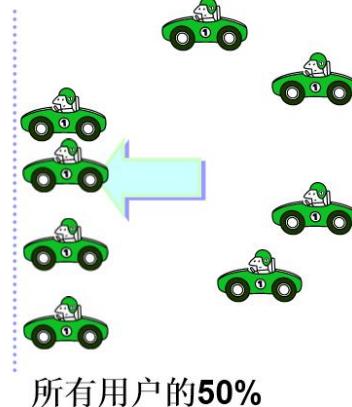
三、集合点

1 集合点的含义和意义

- 集合点用于保证真正的并发。
 - ✓ 当脚本运行到集合点函数时，将停止运行并等待其允许运行的条件(即集合点策略)，达到后才释放集合点开始运行。
 - ✓ 用来分析软件性能瓶颈。
- 一般将集合点放在开始事务代码之前。

2 集合点策略

用于设置到达集合点用户的标准时，系统就释放用户，继续运行。



四、IP 欺骗

1 使用 IP 欺骗的场合

- 应用程序服务器经常缓存来自同一台计算机的客户端信息，网络路由则尝试缓存来优化吞吐量，若许多用户使用同一个 IP，服务器和路由器都将尝试进行优化处理，使用同一 IP 的负载测试不反映真实情况。
- 有些应用程序禁止同一 IP 同时访问。

2 启用 IP 欺骗的过程

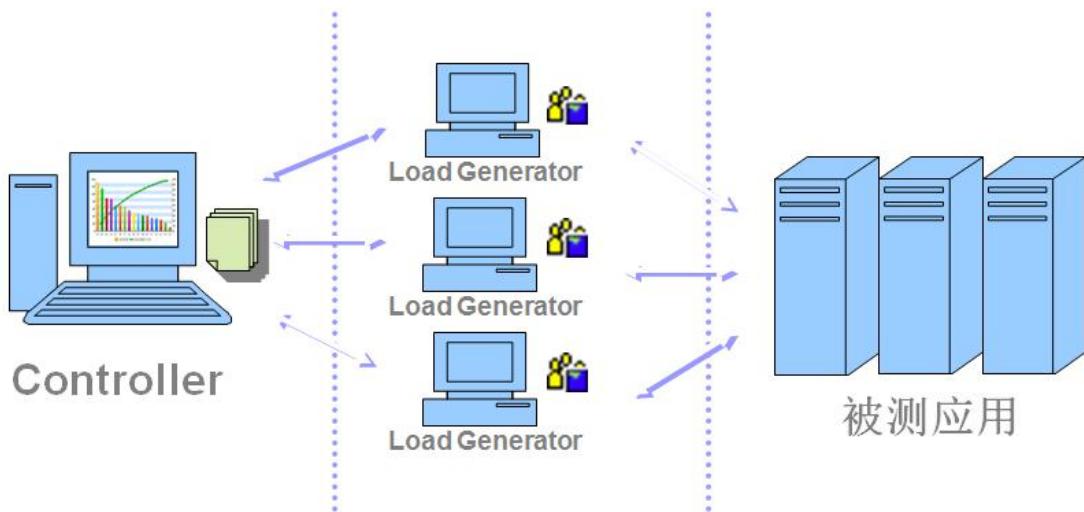
- 设置本机地址。
- 脚本中必须使用真实 IP。
- 配置 IP 地址池。
 - ✓ 本机 IP 也会被使用。
 - ✓ “要添加的数字”：输入总共要添加多少个 IP。
 - ↳ Verify that new IP address are not already: 表示验证新的 IP 地址未被使用，指示 IP 向导对新地址进行检查。这样只会添加未使用的地址。
 - ↳ 可以将 IP 保存为.ips 文件，以后直接加在此文件即可。

-
- 场景中启用 IP 欺骗。
 - 若要在日志中看到 IP，需要在脚本中进行运行时设置，开启扩展日志(至少一项)。
 - 日志中输出 IP
 - ✓ lr_get_vuser_ip()
 - ✧ 只能在场景运行时看到。

五、多机联合负载

1 负载均衡技术

为了尽可能减少或避免本身的测试机成为测试过程中的瓶颈，测试过程中需要使用所有的测试机产生 Vuser，对被测试系统进行施压。



2 Load Generator(负载发生器)

- 当控制器 Controller 发出执行命令时，Load Generator 负责和其它的负载机建立起联系并强制负载机执行。
- 一个 Controller 通过 Load Generator 来控制多台负载机。
- 控制器通过代理程序(Loadrunner Agent Process)控制负载机运行，需要负载机必须安装并启动代理程序。
- 启动代理程序后，场景初始化时，控制器会向负载机发送一个二进制文件，该文件中包括待运行的脚本信息。

3 计算负载机数

- 查看虚拟用户以进程还是线程方式运行
 - ✓ 设置→常规→其他
- 进程方式并发
 - ✓ 1 个虚拟用户对应 1 个 mmdrv.exe 进程。
- 线程方式并发
 - ✓ 一个 mmdrv.exe 进程最多可以支持 50 个线程并发用户数；

-
- ◆ 如 10 个虚拟用户，只出现 1 个 mmdrv.exe 进程，51 个虚拟用户，则会出现 2 个 mmdrv.exe 进程；
 - ◆ 线程方式虚拟用户占用的内存折合计算。
- 最后，用所有虚拟用户占用总内存除以每台负载机内存即可。

4 添加 Load Generators

- 准备负载机
 - ✓ 负载机先安装 LR 完整安装程序中的.net 部分。
 - ✓ 负载机只需安装 LR 功能中的 Load Generator。
- 所有负载机、控制机关闭防火墙
- 创建场景
 - ✓ 添加负载机
 - ◆ 改选为百分比模式。
 - Scenario->Convert Scenario to the Percentage Mode
- 连接负载机
 - ◆ Ready: 连接成功。
 - ◆ Connecting: 正在连接。
 - ◆ Active: 正在运行 Vusers。
 - ◆ Down: 未连接。
 - ◆ Failed: 连接失败。
- 注意
 - ✓ 录制代码时，应使用真实 IP，不能是 http://127.0.0.1 和 localhost。
 - ✓ 负载机使用 IP 欺骗时，需额外配置 IP 地址，且需要与控制机同网段。
 - ✓ 负载机如果不能运行，尝试打开小雷达。

六、 手工场景和目标场景

1 手工场景和目标场景

- 场景设计分为手动设计和面向目标两种。
- 手动测试场景
 - ✓ 灵活，更接近用户真实使用，使用较多。
- 面向目标
 - ✓ 测试性能是否达到预期目标，在能力规划和能力验证的测试过程中常用。首先定义要达到的目标，然后 LoadRunner 会自动基于这些目标创建场景，运行过程中，会不断的将结果与目标相比较，以决定下一步怎么走。

2 目标类型(Goal Type)

- Virtual Users 目标
 - ✓ 主要用来测试服务器对并发用户的处理能力。
 - ✓ 与手工场景相似。
- Hits per Second 目标

-
- ✓ 目标是点击数/秒。同时要设置最小虚拟用户数和最大虚拟用户数，当场景执行时，Controller 会使用最小的虚拟用户来达到定义的目标。如果最小的用户达不到目标，那么 Controller 将会增加虚拟用户数，直到定义的最大的虚拟用户数。

➤ Transactions per Second 目标

- ✓ 目标为每秒处理的事务数。注意，在脚本中一定要定义事务，否则事务名的地方为空白。
- ✓ 从业务角度看，每秒处理的事务数即为每秒处理的业务笔数，所以该指标更多的用于衡量系统每秒处理的业务数。
- ✓ 当虚拟用户数成倍增长时，处理的事务数并不会成倍增长。

➤ Transactions Response Time 目标

- ✓ 目标是设置事务的响应时间，该目标要求在脚本中一定要有事务。
- ✓ 如果事务响应时间没有达到目标值，说明当前事务响应时间还是可以接受的，也说明系统还可以支持更多的虚拟用户。假设达到目标时虚拟用户数为 M，则说明系统最多只能处理 M 个用户同时请求。
- ✓ 目前关于响应时间有一个广泛的应用原则就是"3-5-8"原则。"3-5-8"原则指的是，如果用户发出一个请求后，这个请求在 3s 之内得到响应，那么给客户的感觉是该系统性能十分优秀，5s 之内请求得到响应，用户会感觉还不错，但当请求响应时间超过 8s 甚至更长的时间以后，用户很有可能就失去信心，从此以后不再访问或者不再喜欢访问该网站、使用该程序等。

➤ Pages per Minute 目标

- ✓ 目标为每分钟处理的页面数。
- ✓ 一个事务可能由多个页面组成。