【引言】本文介绍了Load Runner和JMeter等性能测试工具在进行性能测试时对脚本所进行的至关重要的**关联**操作，并针对现代信息系统本身及其性能测试所模拟的用户操作进行分析、总结，得出如下结论：

1. 脚本的“参数化”和“关联”是模拟不同的用户进行大量并发网站操作，实现性能测试的两种不同操作；
2. 对于现代信息系统，没有对脚本进行关联操作几乎无法实现性能测试；
3. 要实现脚本的关联，对系统的内部信息的了解是必不可少的；
4. 对于某些“动静两相宜”的过程参数，可以通过参数化的方式来赋值，也可以通过关联来抓取，但采用关联更为高效；
5. 通过添加性能测试扩展模块来增加可关联的参数，可以提高性能测试的效率，在J2EE网站应用中，还可以采用“类条件编译”的机制，使得该模块仅作用于性能测试环境。
6. 性能测试概念及工具

性能测试工具对Web应用进行性能测试的过程是先录制用户的一系列网站操作的HTTP请求和响应，然后启动大量线程重放脚本来模拟不同用户的网站操作以测试Web应用在一定压力下的性能表现。

常见的性能测试工具有国产的KylinPET、HP的Load Runner和Apache免费开源项目JMeter，

现代信息系统面向多用户、实现不同用户的身份认证和权限控制，各用户彼此业务独立，因此需要对脚本进行参数化和关联两种不同的操作，才能够实现正确的用户并发操作，实现性能测试。

1. 脚本的参数化和关联

首先理解一下什么是性能测试工具脚本的“参数化”和“关联”，以及为什么需要关联。

用户的网站操作通常包括登录网站，执行一定的事务，退出登录。

当模拟不同用户的网站登录操作时，需要为每个线程设置不同的用户名、密码，因此一个录制好的脚本首先需要进行用户名、密码的参数化工作，给不同线程所使用的用户名、密码进行赋值。脚本的“参数化”可以理解为对所模拟的不同用户的HTTP请求所使用的静态参数进行**赋值**，通常采用文本文件、SQL查询等方式实现。

JMeter参数化实例如下：

实例中，使用CSV文件中不同的用户名、密码给登录HTTP请求中的用户名和密码参数赋值， 以实现不同用户的并发登录。

然而，对现代信息系统的操作**仅仅参数化脚本的用户名、密码还不够**，**还需要对脚本进行关联**。

关联是性能测试脚本制作和调试中极为重要的一个环节，其过程是在HTTP请求的顺序执行过程中，从前面的HTTP响应中**抓取**动态数据并用作后续HTTP请求的参数，脚本的“关联”可以理解为对性能测试工具所模拟的**不同用户在不同时刻、不同场景**下的HTTP请求所使用的**动态参数**进行**抓取**。

这是因为在网站操作过程中，会产生大量的动态数据，如鉴权认证码、动态组件号码、会话变量、时间戳、不同的业务数据等，这些动态数据可能会作为后续请求的参数传回服务器，**没有进行脚本关联，不正确的请求参数会导致后续请求的失败，无法实施性能测试**。

下面是一个脚本关联的实例，从实例中，我们可以看到，

常用性能测试工具如LR、JMeter等均有参数关联相关的功能，