关键字：LoadRnner、协议选择、专题  
  
  
内容简介  
  
LoadRunner脚本开发过程中的协议选择作为脚本开发的第一个步骤，相当重要，只有选择了合理的正确的协议才能开发出好的测试脚本。在协议选择过程 中需要注意选择与被测对象相应的脚本，比如Web系统一般选择HTTP./HTML协议，FTP服务器一般选择FTP协议的脚本，另外在协议选择之前需要 确认被测系统使用了什么协议，这里说到的协议指的是应用层的协议，一般确认系统选择了什么样的协议一般有以下几种方法：  
  
1）  
可以直接确认  
  
这种情况下你可以通过你的测试常识直接判读系统使用了什么样的协议，比如前面说的Web系统使用了HTTP./HTML协议，Ftp服务使用了FTP协议等；  
  
2）  
通过研发人员了解被测系统使用了什么样的协议  
  
如果通过你的判断不能确定使用了什么样的协议这个时候，你可以跟研发人员进行沟通，确认他在开发的过程中使用了什么样的协议。  
  
3）  
使用常用的数据监听工具进行数据包分析  
  
有的时候可能你的研发人员也不能确定他在开发过程中使用了什么协议，这个确实是有可能的特别是现在的研发人员特别喜欢用别人的插件，或者开发环境封装的很 好，导致他们没有开发底层的协议栈，这个时候就需要你自己去判断，自己判断过程中可以借助协议分析工具，常用的协议分析工具如：sniffer Pro，ethreal等；这些工具可以通过抓取数据然后对数据包进行分析的方法分析出现在常用的协议。能力还是相当的强的。  
  
4）  
万能协议Winsocket协议  
  
如果上述方法都不行，但是你还想用LR完成你的性能测试，没有办法这个时候可以拿出这个Winsocket协议进行脚本的开发，这个协议只要是在 Windows上的通信的软件都可以将脚本开发出来，但是脚本调试的难度相当到，需要看大量的16进制的数据，并且有的时候还需要复杂的管理，本人建议能 不用，最好不用。  
  
另外可以通过以下的这些文章对协议选择进行更深入的了解：  
  
1.  
LoadRunner的协议选择  
  
1.LR支持多种协议，请大家一定要注意，这个地方协议指的是你的Client端通过什么协议访问的Server，Client一般是面向最终使用者 的，Server是第一层Server端，因为现在的体系架构中经常Server层也分多个层次，什么应用层，什么数据层等等，LR只管Client如何 访问第一层Server.  
  
2.特别要注意某些应用，例如一个Web系统，这个系统是通过ActiveX控件来访问后台的，IE只是一个容器，而ActiveX控件访问后台是通过 COM/DCOM协议的，这种情况就不能使用Web协议，否则你什么也录制不到，所以，LR工程师一定要了解应用程序的架构和使用的技术。  
  
  
  
3.谈谈多协议LR是支持在一个脚本里面使用多协议的，不过这个多协议是有一定的限制的，我有一个LR7.6上哪些协议支持多协议，LR8.0是不是取消 了这些限制我不清楚。象HTTPS，一般来讲一定要选择多协议，但在选择具体协议的时候一定只选Web协议，这时候才能作那个端口映射。  
  
  
  
4.谈谈Web协议中Option Web协议里面有一些Option，具体我记不清楚了，但有一个地方很重要：就是选择URL方式还是HTM方式录制，这里有一些原则a. 尽量使用HTM方式录制b.如果使用了Javascrīpt，并且javascrīpt里面与后台有交互，那一定要使用URL方式，例如 javascrīpt用于判断用户名和密码是否正确（与后台一定会有交互），那么就要使用URL方式录制；但如果Javascipt只是判断用户名或密码 的格式，那一般不会与后台发生交互，就不需要使用URL方式。至于什么情况下一定要用URL方式，我曾经写过一个文档，但现在我不记得了不好意思。  
  
  
  
5.谈谈并发点和结果检查很多人喜欢使用并发点和结果检查，我谈谈我自己的看法。第一我基本上从来不使用结果检查，压力测试的结果我一般是去后台检查存储 层的数据是否正确的。LR只能检查协议层上的错误，但如果在大量并发用户的情况下使用返回结果检查例如reg\_text\_check，会极大的占用客户端 的资源，我一般不这么作；至于并发点，我只有在客户强烈要求的情况下才会使用，否则基本不用。  
  
2. LoadRunner的协议选择、Winsocket、CS应用程序  
  
很多时候一提到不是基于浏览器的应用，很多人就会想到用WinSocket协议来录制，仿佛Form窗体都可以用Winsocket 。  
  
  
从道理上讲网络通讯的底层都是基于Socket的，例如TCP、UPD等，似乎所有的程序都可以用Socket协议来录制。但是事实不是这样的，因为选择的协议决定了LoadRunner如何捕获数据包。否则会多捕获很多无用的数据。  
  
  
因此，不是所有的程序都是适合WinSocket协议的。实际上，那些基于Socket开发的应用才真正适合Socket协议来进行录制。其他的，例如基于数据库的应用，就不太时候Socket协议，甚至可能录制不到脚本。  
  
  
很多C/S程序，一定要选择合适的协议。根据作者的经验，C/S的程序多数需要手工开发很多脚本，因为录制的很多回放时候或多或少都会有些问题，但是可以参考录制的结果。  
  
  
所以测试一个程序，一定要搞清楚开发人员用了什么技术、数据流是什么协议封装的。  
  
  
附件是我们自己开发的Controller，我们自己用面向对象实现了并发测试架构（目前支持并发、迭代、thinktime、参数文件、启动时间间隔，集合点功能正在开发中）。借助我们自己开发的Agent，能很好的测试我们的C/S架构的程序。  
这个工具和LoadRunner配合起来，可以完成大多数性能测试。  
这个工具主要为我们测试视频播放效果而开发，呵呵。这是LoadRunner不太擅长的。  
  
3.  
使用LoadRunner如何更好的选择协议  
  
LoadRunner支持的协议和应用非常广泛，测试B/S系统选择Web(Http/Html)。  
测试一个C/S系统要根据所用到的后台数据库来选择不同的协议：  
1.后台数据库是sybase，则采用sybaseCTlib协议。  
2.后台数据库是sql server,则使用MS sql server协议。  
3.后台数据库是oracle 数据库，就使用oracle 2-tier协议。  
4.没有数据库的c/s（ftp,smtp）系统，可以选择windowsSockets协议。  
5.其他的ERP，EJB（需要ejbdetector.jar），选择相应的协议即可。  
  
4.  
Loadrunner录制脚本，协议的选择  
  
一般对于常见的应用软件，可以根据软件的结构来选择协议：  
        B/S 结构，选择WEB（Http/Html)协议  
        C/S结构，可以根据后端数据库的类型来选择，如SybaseCTLib协议用于测试后台的数据库为Sybase的应用；MS SQL Server协议用与测试后台数据库为 SQL Server的应用；对于一些没有数据库的Windows应用，可选用Windows Sockets底层协议。  
  
  
总之，真确选择协议，就要熟悉被测试应用的技术架构。以下列出一些LoadRounner支持的协议：  
一般应用：C Vuser、VB Vuser、VB scrīpt  Vuser、JAVA Vuser、JavascrīptVuser  
        电子商务：WEB（Http/Html)、FTP、LDAP、Palm、Web/WinsocketDual Protocol  
        客户端/服务器：MS SQL Server、ODBC、Oracle、DB2、Sybase CTlib、Sybase DBlib、Domain Name Resolution(DNS)、Windows Socket  
        分布式组件：COM/DCOM、Corba-Java、Rmi\_Java  
        EJB:EJB、Rmi\_Java  
        ERP/CRP:Oracle NCA、SAP-Web、SAPGUI、SAPGUI/SAP-Web Dual Protocol、PropleSoft\_Tuxedo、Siebel Web、Siebel-DB2 CLI、Sieble-MSSQL、Sieble Oracle  
        遗留系统：Terminal Emulation (RTE)  
        Mail 服务：Internet Messaging(IMAP)、MS Exchange(MAPI)、POP3、SMTP  
        中间件：Jacada、Tuxedo 6、Tuxedo 7  
        无线系统：i-mode、voiceXML、WAP  
        应用部署软件:Citrix\_ICA  
        流：Media Plays(MMS)、Real  
  
5.  
LoadRunner与Winsock协议  
  
在讨论winsock解决方案之前，我们先讨论一下各种协议是如何工作。从前面的简介可以了解到很多的高级协议，例如FTP，HTTP协议等。以及所有基 于window的应用（例如IE，WS-FTP）底层都是在Winsocket层上通信，因此任何高级协议的底层都是用Winsocket通信。  
什么时候在LR中选择Winsocket协议呢？你要先了解LR是怎么样工作的：LR捕捉API请求然后再把它们回放。所以当你在创建LR WEB脚本的时候，VUGEN捕捉从IE出去的所有的HTTP请求。除此之外lr还支持其他很多协议，例如Oracle，ODBC等。在选择不同协议录制 脚本的时候，LR是依靠hooks捕捉正确的API请求。所以既然大部分网络协议都是架构在winsocket协议之上的，那对于lr不支持的协议，我们 都可以在winsocket层上录制脚本。所以当找不到合适协议的时候，可以选择winsocket来录制。  
  
  
录制WinSock协议脚本！  
   Lr录制新的虚拟用户脚本，选择winsock协议；  
在web虚拟用户脚本中录制的是URL信息，所以VUGEN启动流览器并运行就可以了，但选择Winsocket录制的时候，可能会是各种形式的应用，并不简简单单就是浏览器，所以在开始的时候我们不需要指定应用的地址。  
  
  
  
下面的例子我们选择winsock来录制web应用，正如上面说的我们开始要指定ie的地址。因为本来lr是支持http协议的，所以这个例子并无具体的意义，只是为了使例子简单。  
winsock脚本典型代码：  
lrs\_create\_socket("socket0"，"UDP"，"LocalHost="0""，"RemoteHost=doors:2084"，LrsLastArg);  
lrs\_create\_socket("socket1"，"TCP"，"LocalHost=0"，"RemoteHost=www2。yahoo。com:80"，LrsLastArg);  
lrs\_send("socket0"，"buf0"，LrsLastArg);  
lrs\_receive("socket0"，"buf1"，LrsLastArg);  
lrs\_send("socket1"，"buf2"，LrsLastArg);  
lrs\_send("socket0"，"buf3"，LrsLastArg);  
lrs\_receive("socket0"，"buf4"，LrsLastArg);  
这是访问雅虎的一个应用，正如我们看到的，winsock先打开一个winsocket连接，然后收发包。我们发现录制的脚本中比web脚本（三个文件）多一个文件。  
  
  
第四个文件是data.ws，它里面记载了在action里面收发所有包的内容。  
下面是一个data.ws的例子。  
send buf0  
"!"  
recv buf1 1  
"!"  
send buf2  
"GET / HTTP/1。1\r\n"  
"Accept: \*/\*\r\n"  
"Accept-Language: en-us\r\n"  
"Accept-Encoding: gzip，deflate\r\n"  
"User-Agent: Mozilla/4。0 (compatible; MSIE 5。01; Windows NT 5。0)\r\n"  
"Host: www。yahoo。com\r\n"  
"Connection: Keep-Alive\r\n"  
"Cookie: B="5oj83bst12l6m&b="2;"Y="v=1&n=8lln5lmi3f8g1&l=6ef8c0d34b0/o&p=m2a2s""  
"qa4110004&r=3f&lg=us&intl=us; T="z=4TVE6A4ZqE6A9dIIpt30。NQNTYGNDE3NTYwM081&""  
"a=AAE&sk=DAAEwinHlDtEm/&d=c2wBTWpFQk16WXdNakUzTkRneQFhAUFBRQF6egE0VFZFNkFn"  
"V0E-;I="i1=010g1q1u24252e2i2k2p2r494a4g4m4q55565b5g6g6t6u7172737678797a7f7""  
"g7k7n7o888f8k8p8q989c9f9i9k9l9n9qacanapb0b3bdbfbhblbqbrbuc0c1c4cgcmcscteie"  
"jgr&ir=73\r\n"  
"\r\n"  
send buf3  
"!"  
recv buf4 1  
"!"  
recv buf5 2048  
"HTTP/1。0 200 OK\r\n"  
"Content-Length: 16030\r\n"  
"Content-Type: text/html\r\n"  
"\r\n"  
"  
buf2包含了发给www。yahoo。com的包，buf5包含了从server收到的回应。正如看到的winsock录制的脚本并不像web脚本那样 具有可读性，因为winsock是我们最后的选择。在buf序列号的后面是buf的字节数。例如5号buffer后面的2048就是表示收到了2048个 帧。  
注意：在很多buffer里面你会发现仅仅包含了一个“！”，这是一个网络常识，这个帧对于应用来说没有任何的作用，完全可以把这些桢清除掉，在 data。ws清除这个帧是效果的，需要到action里把这些桢注释掉。譬如例子中收到和发出的buffer可以用//注释掉。这样可以使脚本运行得更 快。下边的例子：buffers 0，1，3 和4 可以注释掉:  
lrs\_create\_socket("socket0"，"UDP"，"LocalHost="0""，"RemoteHost=doors:2084"，LrsLastArg);  
lrs\_create\_socket("socket1"，"TCP"，"LocalHost=0"，"RemoteHost=www2。yahoo。com:80"，LrsLastArg);  
// lrs\_send("socket0"，"buf0"，LrsLastArg);  
// lrs\_receive("socket0"，"buf1"，LrsLastArg);  
lrs\_send("socket1"，"buf2"，LrsLastArg);  
// lrs\_send("socket0"，"buf3"，LrsLastArg);  
// lrs\_receive("socket0"，"buf4"，LrsLastArg);  
lrs\_receive("socket1"，"buf5"，LrsLastArg);  
lrs\_send("socket0"，"buf6"，LrsLastArg);  
  
、  
  
6.  
LoadRunner之协议选择  
  
在学习LoadRunner协议选择之前,我觉得我们有必要了解一下协议的基本概念。首先我们知道,计算机与计算机之间的通信都离不开通信协议，接着我们 来说说通信协议的概念。通信协议是什么，通信协议实际上是一组规定和约定的集合。说白了就是两台或者多台计算机在通信时必须约定好本次通信做什么，例如是 进行文件传输，还是发送电子邮件；然后约定怎样通信，什么时间通信等。因此，通信双方要遵从相互可以接受的协议（相同或兼容的协议）才能进行通信，如目前 因特网上广泛使用的TCP/IP协议等，任何计算机连入网络后只要运行TCP/IP协议，就可访问因特网。  
  
了解了协议的基本概念和作用之后，我们来说说LoadRunner的协议选择。LoadRunner首先是一个测试工具，其次是一个性能测试工具，然后是 该工具是一个基于协议，也就是说LoadRunner测试的对象都需要使用通信协议，对于那些不使用通信协议仅仅进行本地处理的软件例如 Microsoft Word，LoadRunner就不适用。  
  
说到通信协议我们来熟悉一下协议的分层，按照OSI的分层模型，分层结构如下：

|  |  |
| --- | --- |
| OSI七层模型 | TCP/IP协议 |
| 应用层 | 应用层（Application） |
| 表示层 |
| 会话层 |
| 传输层 | 传输层（Transport） |
| 网络层 | 网络层（Internet） |
| 数据链路层 | 网络接口层（Network） |

按照TCP/IP协议的分层，分层结构如下：

|  |  |
| --- | --- |
| TCP/IP协议 | 对应的服务 |
| 应用层 | WWW、SMTP、FTP、Telnet、Gopher、SNMP |
| Socket、NetBIOS |
| 传输层 | TCP、UDP |
| 网络层 | IP（[ARP](http://www.csai.cn/incsearch/search.asp?key=ARP)、RARP、ICMP） |
| 网络接口层 | Ethernet、X.25、PPP、SLIP |

   
    
  
   
  
第一个分层是由OSI制定但不实用，后一个是目前广泛使用且被业界认做既定标准的协议分层，下文探讨的LoadRunner协议选择即按TCP/IP协议的分层模型讨论。  
  
接着来说说LoadRunner VuGen  
中的协议分类，VuGen（LR8.1）中的协议分类如下表所示：

|  |  |
| --- | --- |
| [java](http://www.ltesting.net/html/19/%3CSTRONG%3E%3CA%20href=)script:;" target=\_self>**应用**程序部署[**解决方案**](http://www.ltesting.net/html/81/category-catid-381.html) | Citrix ICA |
| 客户端/ [**服务器**](http://www.ltesting.net/html/78/category-catid-378.html) | DB2 CLI、DNS、Informix、MS SQL Server、ODBC、**Oracle**（2 层）、Sybase Ctlib、Sybase Dblib 和 **Windows** Sockets 协议 |
| 自定义 | C 模板、Visual Basic 模板、**Java** 模板、Javascript 和 **VB**Script 类型的脚本。 |
| 分布式组件 | 适用于 COM/DCOM、Corba-Java 和 Rmi-Java 协议。 |
| 电子商务 | FTP、LDAP、Palm、PeopleSoft 8 mulit-lingual、**SOA**P、[**Web**](javascript:;)(HTTP/HTML) 和双 Web/WinSocket 协议。 |
| Enterprise Java Bean | EJB 测试和 Rmi-Java 协议。 |
| ERP/CRM | Baan、Oracle NCA、Peoplesoft-Tuxedo、SAP-Web、SAPGUI、Siebel-DB2 CLI、Siebel-MSSQL、Siebel-Web 和 Siebel-Oracle 协议 |
| 传统 | 终端仿真 (RTE)。 |
| 邮件服务 | Internet 邮件访问协议 (IMAP)、MS Exchange (MAPI)、POP3 和SMTP。 |
| [**中间件**](http://www.ltesting.net/html/85/category-catid-385.html) | Jacada 和 Tuxedo （6、7）协议。 |
| 流[**数据**](javascript:;) | Media Player (MMS) 和 Real 协议。 |
| 无线 | i-Mode、VoiceXML 和 WAP 协议。 |

仔细研究发现LoadRunner VuGen中的协议与文章开头所说的通信协议还是有一定的区别的，例如像LoadRunner VuGen中的C 模板、Visual Basic 模板、Java 模板、Javascript 和 VBScript 类型的脚本均为开发语言，非通信协议，但LoadRunner即把它列在这儿，我们也就暂且认可。  
  
  
了解了LoadRunner的协议类型之后,我们进入正式话题，即测试时如何选择协议。  
  
  
正式测试之前，测试人员都需要预先熟悉被测对象，我们需要知道我们的被测对象是一个什么样的结构，是B/S结构还是C/S结构，了解这个之后，我们还需要 了解被测对象所使用的协议是什么，也许有的人说我们的程序使用的是TCP/IP协议，其实他的回答跟我们需要知道的差别很大，因为我们知道，所有的通信软 件都需要使用TCP/IP协议，为什么呢，因为这个协议是底层协议，所有应用层数据都必须经过这个协议封装之后才能向更底一层传输。我们需要知道的是被测 对象在应用层使用的是什么协议，就像我们使用邮件客户端发送邮件一样，我们知道使用的应用层协议是SMTP，使用邮件客户端接收邮件时使用的时POP3协 议。了解上述信息之后，我们的定位也就准确了，也就是说在LoadRunner中所说的协议基本上都是应用层协议（也有底层协议，例如 Windows Sockets），知道这点之后，我们在询问开发人员时就避免了不少麻烦，省的开发人员告诉你我们使用的是TCP/IP协议或者其他底层协议。  
  
  
了解了LoadRunner中的协议之后，我们就进入协议确定阶段，协议的确定，通常有如下几种方法：  
  
1、通过询问开发人员获知所使用的协议，通常这是最简单也是最直接的方法；因为没有人比开发人员更清楚他们所开发的应用程序使用的什么通信协议了；  
  
2、通过概要或详细设计手册获知所使用的协议，在没有开发人员支持的情况，通过概要设计或详细设计获知所使用的协议不失为第二简便方法；  
  
3、通过协议分析工具捕包分析，然后确定被测对象所使用的协议。在使用协议分析工具分析协议过程当中一定要摒除底层协议，不要被底层协议所迷惑；  
  
4、通过以往测试经验确定被测对象所使用的协议，当然通过这种方法确定的协议有一定的不准确性；  
  
通过以上四种方法我们基本就确定了录制时应该选择什么协议，光确定协议是没有用的，最主要的是付诸行动，确定了协议之后，我们进入VuGen开始录制脚本，录制完成后看看是否生成相应脚本，如果脚本内容为空可能我们选择的协议不正确，我们可以尝试选择其他协议。  
  
一般来说协议选择有如下原则：  
  
B/S 结构，选择WEB（Http/Html)协议；  
  
C/S结构，可以根据后端数据库的类型来选择，如SybaseCTLib协议用于测试后台的数据库为Sybase的应用；MS SQL Server协议用与测试后台数据库为 SQLServer的应用；对于一些没有数据库的Windows应用，可选用Windows Sockets底层协议；使用了数据库但使用的是ODBC连接的数据则选择ODBC协议；  
  
对于有些使用纯JAVA编写的C/S结构的东东，采用JAVA，而且不能录制只能手工编写代码（工作量和难度还是有的）。同样不能录制的还包括C、VB Script、VB、VBNet User协议。  
  
对于Windows Sockets协议来说，最适合的那些基于Socket开发的应用程序；但是由于网络通讯的底层都是基于Socket的，因此几乎所有的应用程序都能够通 过Socket来录制，哪可能有人会问，哪既然Socket都能录制下来，还要那么多协议做什么，价格还贼贵，其实最主要的原因就是Socket录制的代 码可读性较差，如果Socket的脚本可读性较高的话，实话就没有其他协议出现的必要性了。  
  
对于邮件来说，首先要看你收邮件的途径，如果你通过WEB页面收发邮件，毫无疑问，你选择协议时就需要选择HTTP协议，如果你通过邮件客户端，像 OutLook、FoxMail之类的，则需要根据操作不同选择不同的协议了，例如发邮件你可能要选择SMTP、收邮件你可能需要选择POP3。