Linux平台下的自动化测试工具

在Linux平台下，借助工具或者其他手段实施测试的自动化时，需要关注以下几点：   
　　(一)各种技术应用的前提。对于在开源社区和一些开源项目中获得的测试工具，首先需要了解工具适用于哪些类型应用的测试，以及工具发布后的发布说明和FAQ。开源的工具通常不像商业工具那样成熟稳定，因此找出工具的适用范围以及探索工具的实现程度是进行自动化测试应用的前提。   
　　(二)各种技术应用的环境需求。对于各类工具，需要关注编译和运行时对各种包和库及其版本的依赖关系以及对预先安装的应用的依赖关系。这些在用户手册中都有详尽的说明。   
　　(三)服务器性能监视器。大部分测试工具没有提供服务器端的性能监控功能，测试工程师需要根据实际的需求编写性能监控脚本来配合工具的使用。   
　　下面结合曾经参与进行过的Linux平台下的自动化测试的研究，面向不同类别的测试用例自动化的需求，将主要从功能测试，如GUI测试、命令行客户端的测试，以及性能测试等几个方面对Linux平台下的测试工作的自动化进行分析和说明。   
　　   
　　GZW自动化测试   
　　   
　　对于GUI测试的自动化，通常的测试工具所使用的捕捉／回放技术有两种，一种是通过记录界面的鼠标事件(如点击、移动)和键盘事件来完成录制和回放，另外一种则是录制和回放都是基于控件的识别和操作进行的，每个脚本的执行都是控件对象的属性改变或事件触发。我们从开源社区可以获得如上两种类型的运行于Linux平台之上的典型测试工具，如Knee和LDTP等。   
　　   
　　(一)Xnee工具   
　　在Linux操作系统的xll环境下，Xnee能够录制、回放和分发用户的动作。Xnee的捕捉／回放技术是记录鼠标事件和键盘事件。进入录制模式时，Xnee记录发送至和来自X server之间的协议数据拷贝，并生成Xneesession文件。在回放模式下，Xnee读取Xnee Session中的事件，模仿整个录制过程(即用户操作过程)完成和x server之间的通讯，被录制的应用软件(Xclient)则接收来自xserver的消息，完成预设的动作。   
　　(二)LDTP测试工具／框架   
　　Linux Desktop Testing Project(LDTP)测试工具／框架能够基于用户在应用界面的选择进行脚本的录制。LDTPI具使用了Gnome环境下的Accessibility库即辅助选项库(at-spi)。使用辅助选项能够获得应用通过AT-SPI协议提供的关于用户界面的信息和界面控件的当前状态或者属性。LDTPI具／框架的体系结构如下：   
　　AT-SPI的基础思想就是为用户界面的可视化元素提供对应的辅助对象，而录制完成的每个脚本的执行都是基于这些辅助对象进行的。对于希望利用LDTPI具进行测试的应用，需要激活辅助选项。   
　　(三)GUI自动化测试工具的应用   
　　在实际的GUI自动化测试中，LDTPI具应用的场景会更广泛一些。LDTPI具可以识别窗口中的对象(如按钮)，测试脚本使用LDTP的API接口，每个API接口对UI对象进行操作存在两个最基本的入口，即窗口和对象，窗口通过窗口的类型和名称(即标题)识别，对象通过希望操作的控件的类型和名称(标签或者关联的标签)识别。我们同样可以通过at-pokel具展现激活了辅助选项的应用程序窗口的对象及对象属性。在测试Linux桌面产品和服务器产品的过程中，使用LDTPI具可以测试任何启用辅助选项的Gnome应用，如Mozilla，OpenOffice.org、Evolution邮件客户端，Nautilus文件浏览器等等，此外还可以测试UI界面基于Swing的Java应用，以及KDE4.O上基于QT4.0的应用等等。   
　　而Xneel具所针对的应用程序类型就没有特别的限制，对于一些简单的窗口验证测试和界面的稳定性测试等则比较有效。Xnee相对于基于控件方式捕获和回放的工具而言，不用担心存在控件不能被识别的问题。 

　　从使用的情况来看，各个工具也都因为实现技术而存在一定的缺陷，如两个工具均不能插入验证点，从而不能实现用例级别的结果验证；LDTP对于界面的个别元素捕获不到以及不能对不支持辅助选项的应用进行测试等等；而Xneel具生成的脚本可编辑性差，同时由于录制生成的脚本中的事件和屏幕坐标相关，因此当出现窗口弹出位置发生变化等问题时，就需要考虑回放时应该如何来处理这些变化。