基于 JMX 的软件自动化测试模型①

林洁文　葛　红 (华南师范大学 计算机学院 广东 广州 510631)

摘　要： 简要分析了现在自动化软件测试系统的需求和 JMX 网络管理体系结构，在此基础上提出了使用 JMX 框架来构建的软件自动化测试模型的方法，最后结合 JMX 规范和实现方法，分析了基于 JMX 的软件自动化测试模型的优势。

关键词： 自动化测试;　JMX；管理；模型

JMX-Based Software Auto Testing Model

LIN Jie-Wen, GE Hong (School of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract: This paper makes an introduction to requirement of software auto testing system and JMX network management architecture. It proposes an approach to build software auto testing system with JMX framework. In the end, the advantages of JMX-based software auto testing model are pointed out.

Keywords: auto testing; JMX; management; model

1　引言

在现在的自动化软件测试中，随着软件规模的庞大，不仅测试用例的数量和复杂度在不停地增长，而且，为了在不同平台上完成测试，测试机的数量也是越来越多，测试工作变得越加艰巨。虽然现在的很多自动化软件测试可以在一定程度上减少测试开销，同时增加在有限时间内的测试量，但是却很难很好管理各测试机上的测试任务、测试状态和测试机本身的一些状态等等。例如 opengl 驱动测试框架 piglit，它把各种测试用例和套件组合起来测试，但不能对测试机进行管理，没有图形化的操作管理界面或是界面不够友好，没有对测试机的状态进行监控等。同时，在现在的大部分自动化软件测试系统中，一个测试机往往只是做一种软件或软件系统的测试，这样就不仅浪费了资源，还使用得测试时间比较长。如果有一种可以把各个测试机和各种测试任务统一管理起来，就可以解决上述的所有问题了。Java Management eXtension (JMX) 是对 Java 平台的扩充 ,提供了一种可扩充的、低实现成本和与完善管理机制的管理框架。JMX 将网络上一切资源(包括应用程序、设备、服务等)通过网络进行管理，使资源可以动态地建立、安装 、激活和卸载，从而建立了一个动态的、可伸缩的和便携的第三代管理体系结构。当今很多的研究和项目采用 JMX 实现其管理框架，例如 JBoss、WebLogic、Adventnet Manager 和 JAIN SLEE 等等。JMX 管理框架完全可以实现对软件自动化测试系统中资源的有效管理，从而节省测试机数量和测试时间。

目前，还没有出现基于 JMX 管理框架的软件自动化测试系统，也几乎没有这方面的研究。本文提出一种基于 JMX 体系结构的软件自动化测试模型，把所有测试机集中在一起组成一个测试网络，然后通过测试管理服务器把各个测试机上的所有测试任务统一集成管理起来，并通过 WEB 实现可视化管理。

2　JMX网络管理框架

Java Management eXtensions(JMX)[1]是 SUN 公司开发的基 Java 的一系列技术规范和开发工具的集合，致力于解决分布式系统管理和软件集成的问题。JMX 是一种管理架构、应用编程接口、可扩充对象和方法的集合体，它可用于跨越一系列不同的异构操作系统平台、系统体系结构和网络传输协议，灵活地开发无缝集成系统、网络及网络管理应用的规范，通过

① 收稿时间:2009-10-14;收到修改稿时间:2009-11-10

© 中国科学院软件研究所 http://www.c-s-a.org.cn

JMX 技术，用户可以轻松实现资源管理及其管理应用程序的即插即用。如图 1 所示，JMX 体系结构是一个三层模型，分别是分布式服务层(Distributed Services Level)、代理层(Agent Level)和设备层(Instrumentation Level)。

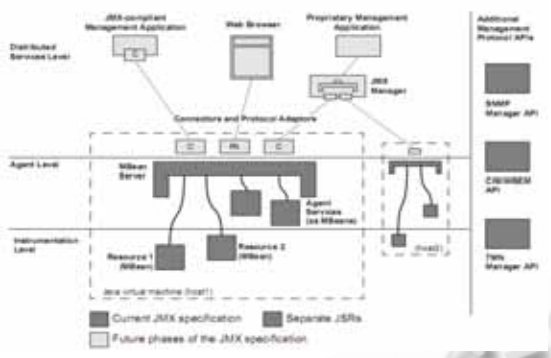

图 1 三层 JMX 体系结构

设备层 提供了实现 JMX 可管理资源的规范。这是三层中最接近可管理资源的一层，它由四种装备管理资源方法和一个接收和发送通知的模型组成。一个 JMX 可管理交资源可以是它可以是一个应用程序、一个设备、一种服务或者策略的实现，被管理资源通过封装成一个符合 JMX 规范的 Java 对象 MBeans (Managed MBeans)后就可以被任何符合 JMX 规范的应用程序管理。在这一层次的 JMX 可管理资源的设计与 JMX 其他层的设计完全独立。

代理层 代理层是在三层的中间层，它由主要由 MBeanServer(提供可管理资源处理程序和代理服务的注册管理)、一组表示被管理资源的 MBeans(包括代理服务)和至少一个协议适配器或者是连接器组成。它向远程管理组件公开一组标准化服务并通过 JMX 可管理资源的 MBean 接口直接控制这些资源。在 JMX 代理内可通过能够动态装入和卸装 MBean 的 MBean 服务器来管理 MBeans。

分布式服务层 JMX 最上面一层，是由连接代理[U1]和管理应用程序的中间件组成。中间件分协议适配器和连接子两种类型。这层的主要目标是指定了 JMX Manager 组件提供的接口。JMX Manager 可以访问代理或代理组来管理由代理公开的 JMX 可管理的资源。例如通过 HTTP 协议适配器，就可以通过网页浏览器连接到 JMX 代理和管理注册在代理上的 MBeans。

这三层的结合实现了一个完整的针对设计和开发完善网络管理方案的架构，从而解决了一个完整的管理方案应该处理好的三个基本问题：

(1) 如何让资源可管理？

(2) 一旦资源可管理，如何让这些资源对管理是可用的(可见的)？

(3) 一旦这些资源对管理是可见的，管理程序应该如何访问它们？

此外，JMX 还包括一套开放的接口，通过这套接口，一个 JMX 代理及其所管理的资源可与多种管理模型协同工作。同时，JMX 规范定义了一套基于 Java 事件模型的通用告警模型。

3 基于JMX的软件自动化测试模型

3.1 软件自动化测试系统的需求

随着软件规模的日益庞大，软件测试的工作量也越来越大。很多的测试系统基于人工测试和自动化测试相结合的方式进行，要编写更多的测试用例，这样的测试工作量较大，对于需求和软件的频繁变更适应能力较弱。提高自动化测试水平，建立一种新的统一管理测试自动化测试框架，这种框架能容易和灵活地把基于需求的配置、自动执行测试用例、实时显示测试数据、对大量的测试日志进行保存和分析、自动产生测试报告等等自动机制集成在一起，并对它们实现管理。这样的测试自动化测试管理框架对于提高测试效率和质量至关重要。

基于文献[2]和对软件自动测试的分析，目前的软件自动化测试系统的几个需求：

(1) 测试的可配置性。为了提高测试系统的灵活性，在进行实际的测试过程之前，可以对测试过程中的参数、测试流程和测试机环境进行配置等等。例如，测试人员可以通过 WEB 网页设置测试的时间阀值，测试用例的数量和种类，对测试机平台的选择和环境的配置等。另外，测试的可配性还包括可配置不同测试任务，如测试 UI、测试数据库等等。

(2) 测试状态信息实时显示和测试结果信息显示及持久化。在测试过程中要及时显示当前的测试进度、测试状态及其中间结果，以让测试人员随时了解整个系统的工作状态测试完后的结果和结论也是必不可少的。同时，对测试产生的日志信息保存到数据库里面，方便测试人员或开者人员对系统 BUG 的定位和修复。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(3) 测试过程的可控制性。为了节省资源，把可以把多个不同的测试任务放在同一台测试机上进行，这些测试任务是对立的，即不能同时间做两种或以的测试，一定是某个时间段只有能一个测试任务在进行，当其完成或是被终止或是长暂停时才能执行下一个测试任务。所以要完成每个测试任务都需要对整个测试流程进行控制。例如启动一个新测试任务，暂停或终止一个执行中测试任务等等。对同一个测试任务里头也有着很多对立的测试用例，也需要对它们的执行进行控制。

(4) 测试系统功能模块的扩展性。一个有完好集成功能的自动化测试系统应该有良好的可扩展性.当系统需要添加新的测试功能时,不能影响现有功能的正常运行,同时现有的系统设计不需要进行改动,否则一旦添加新功能带来的是系统的重构,不仅违背了系统设计的重用原则,而且会浪费人力物力和时间,导致成本的上升和缓慢了整个软件测试进度，甚至对整个软件开发过程造成严重的影响。同样，去除某些功能模块时也不能影响系统正常工作。

(5) 资源的可共享性。主要是指测试机资源，当然还包括一些共用性测试和配置脚本。把不同项目的测试机器都放在一起组成一个测试机群，然后通过一台服务器把这些测试机管理起来，通过服务器对各测试进行任务分配。这样就可以实现资源的有效利用，当一台测试完成它的测试后，就可以用来做其他项目的测试工作。

3.2 基于 JMX 的软件自动化测试模型

为了解决 3.2 中所述的需求，结合 JMX 网络管理框架的高可扩展性、完善的网络管理机制、与平台无关性和分布式管理等优点，本文把 JMX 技术运用到软件自动化测试系统中，整个系统框架图如图 2。

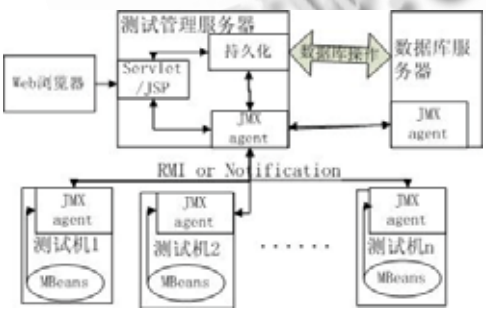

图 2 基于 JMX 的软件自动化测试模型

如图 2 所示，根据 JMX 体系结构的分层特点（分三层），把软件自动化测试系统框架也分为测试模块(图中的 MBeans)、测试代理服务(图中的 JMX agent)和测试管理服务程序(图中的 Servlet/JSP)三层，分别对应 JMX 体系结构的设备层、代理层和分布式服务层。

测试模块 这一层具备了 JMX 体系结构的设备层的所有特点，主要的工作就对测试程序进行 JMX 规范的封装，包括资源的属性(测试程序要受管理的信息，如状态)、构造函数(管理应用程序或者其他 JMX 代理来创建资源的实例)、操作(由管理应用程序或者其他 JMX 代理服务来调用来执行，如远程的操作可以通过 JMX 里面的远程调用 RMI 机制来执行)、参数(为操作提供必需的参数)、通知(由资源发出 ,由 JMX 通知基础设施来完成 ,发到任何感兴趣即设置为该资源的监听者的代理服务，一般来说不一定要有通知)，把测试程序封装成 MBeans。然后把封装好的 MBeans 注册到 JMX agent 中的 MBean Server 上，就可以通过这个代理上的 MBean Server 对 MBeans 进行管理了。例如一个测试任务，可以把测试的状态信息、进度数据当成属性，测试启动、终止等操作封装在一个 MBean 中。同时，如果对一些测试任务，在测试进行时还需要对系统资源或是一些硬件设备进行监控，也可以很方便把系统资源或是一些硬件设备封闭成一个个的 MBean，注册到测试代理服务层中的 MBean Server 中，这样就很方便实现了功能的扩展。如果想去掉某些测试或是监控功能，只要在 MBean Server 中解注册对应的 MBean 就可以了。

测试代理服务层 测试代理服务层主要包括一个登记 MBean 的 MBean Server 和一些也要在在封装为了 MBeans 的代理服务。这里的 MBean Server 就是 JMX 规范里的 MBean Server 的一个实例化，而代理服务 MBeans 就按照 JMX 规范封装了一些对测试模块的操作，然后注册到 MBean Server 上，这样就可以为其他测试代理服务和上层测试管理服务程序提供一些操作调用或是实现一些对测试过程进行控制的操作。例如可以在这一层实现对一些测试脚本的运行、测试机环境的配置，测试数据或测试脚本文件的传输等等。

测试管理服务程序 测试管理服务程序完成整个自动化测试系统管理工作，工作在 JMX 的最高层分布式服务层，是整个系统框架最接近用户(如测试人员)的一层。为了完全与 JMX 体系融合在一起，测试管理服务程序可用 Java 的相关技术来实现，如图 2 中的

© 中国科学院软件研究所 http://www.c-s-a.org.cn

Servlet/JSP。测试管理服务程序通过 JMX 提供的高度灵活的远程接口(Remote Interface),根据每个测试代理服务的 JMX Service URL 来获到到各个测试机上的测试代理服务的连接 MBeanServerConnection,通过这个 MBeanServer-Connection 就可以对测试代理进行管理，主要就是对测试模块进行设置、控制和配置，收取或采集各个测试任务的信息，增加和删除测试任务和一些功能的增减等等。同时通过 WEB 技术把这些功能友好的可视化，测试人员只要打开浏览器访问测试管理服务器就可以进行一切的测试管理工作和测试相关信息的查看。除此之外，也可把对测试重要数据在这一层进行持久化，如采用 Java 技术中比较主流的数据持久化框架 Hibernate 把重要数据写到数据库中。

3.3 基于 JMX 软件自动化测试模型的主要优势

根据 JMX 网络管理框架的优点，跟一般的自动测试相比，基于 JMX 软件自动化测试系统在系统设计、系统使用、系统扩展等都有很好的优势：

(1) 系统设计开发简单、快速。Sun 公司根据 JMX 规范开发了一套开放的接口，里面包括四种 MBean 操作、远程调用接口、通知机制、动态管理机制等等的实现，通过这套接口，一个 JMX 代理及其所管理的资源可与多种管理模型协同工作。因此，在设计开发过程完全可以省略掉这些工作，直接使用 JMX 提供的接口。

(2) 框架提供可伸缩的管理。每一个代理服务和 MBean 都是一个独立的模块，它可以根据需要动态的加载或卸载，这种基于组件化的耦合方式使解决方案更加具有弹性，可以自由切换各种规模的测试模块。

(3) 测试任务集成更简单，可执行测试任务更多。把测试程序、测试控制封装成管理的对象 MBean，注册到管理服务器或测试代理服务上。这样可以利用管理框架提供的标准接口去管理测试工作。

(4) 灵活的可配置性。JMX 中的被管理对象 MBean 的属性是支持配置的。通过在 WEB 服务里调用 JMX 的管理接口，测试人员可以在浏览器上灵活地对测试参数、测试环境进行配置。

(5)更高效的资源利用和集中管理。JMX 是一个完善的、适应未来的网络管理理念的网络管理框架。把所有测试机组成一个测试机群，然后采用 JMX 框架来管理。对于不同的测试任务，对其测试环境进行说明或配置后就可以加入到测试管理服务器上的测试队列中，如果有合适的测试机空闲，服务器就把测试任务放到该测试机上执行，并把服务器上的相关测试套件下载到该测试机上。反过来，如果某测试机完成任务，它会告诉服务它已经完成任务，可以接受新的任务。这样就可以充分利用测试机和集中管理测试任务。

(6)更好的用户交互，测试工作更方便。通过最上层的 WEB 服务，测试人员在远程通过浏览器的友好网页界就可以轻松完成测试工作。不必亲自跑到测试实验室里或是去找哪一台可用的测试机，也不会因特定测试机有问题而影响测试工作的进行。同时，通过 JMX 来实现对测试机的状态监控和 JMX 的通知机制能很方便、及时知道测试机的健康状况，对健康有问题的测试机及时修复。

4 结语

随着软件测试的工作量和力度的增加，软件自动化测试系统的功能就要更全面，使用要更方便，功能要更好可伸缩性。对一种新的集成管理的自动化测试系统的需求越来越强烈。本文在研究 JMX 体系结构和一些自动化软件测试分析的的基础上，提出一种基于 JMX 的自动化软件测试模型，使得自动化软件测试可管理性更强、系统设计更简便、可扩展性更好、测试自动化程度更高和资源共享更充分。

参考文献

1 Java Management Extensions Specification, version 1.4. Sun Microsystems,Inc. 2006.

2 张曙光,刘亚斌. 基于软件框架的控制分组件测试系统的研究与实现.微计算机信息, 2006,22(5-1):32－34.

© 中国科学院软件研究所 http://www.c-s-a.org.cn