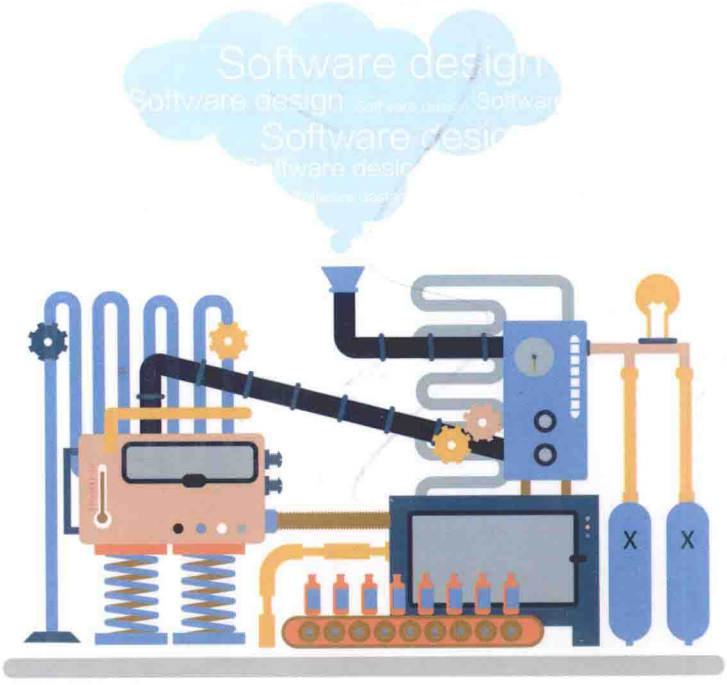
Broadview®
www.broadview.com.cn

自动化运维
软件设计实战

面对众多的设备类型如何实现自动化运维?
如果Ansible、Puppet、SaltStack都无法满足你对自动化运维的需求，那么本书将会带给你一种全新的思路!

吴文豪 著



中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

自动化运维软件设计实战

吴文豪 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书主要讲解采用 OSGi 技术来设计一款可插拔式的运维软件的方法与思想，为读者提供一种不一样的运维软件设计与自动化运维解决方案。

本书分三部分，第一部分讲解开源社区中比较流行的三款集中化运维软件，第二部分与读者一起分享为什么要采用 OSGi 的技术来设计集中化运维软件，第三部分介绍设计这款运维软件所涉及的技术和一些设计思想。

本书适合从事系统运维及运维开发的人员阅读。

图书在版编目（CIP）数据

自动化运维软件设计实战/吴文豪著. —北京：电子工业出版社，2015.7

ISBN 978-7-121-26468-9

Ⅰ. ①自… Ⅱ. ①吴… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2015）第 142285 号

责任编辑：陈晓猛
印　　刷：北京京师印务有限公司
装　　订：北京京师印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×980　1/16　印张：18.25　字数：408.8 千字
版　　次：2015 年 7 月第 1 版
印　　次：2015 年 7 月第 1 次印刷
印　　数：3000 册　定价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前 言

不知不觉也与各种开发语言做了挺久的小伙伴，依稀记得当初很好奇用 C 语言这种在黑框框里面跑的程序怎么才能写出一个界面，后来在兴趣的驱使下不断地接触各种各样的技术，让我感慨时间过得还是挺快的。

我曾经接触过一个与自动化运维相关的项目，整个项目对我来说挑战是非常大的，而且开发过程也非常坎坷。当做到运维部分的时候，出现了非常大的挑战，我们不得不面对技术水平参差不齐的维护团队，各种各样的操作系统，限制条件非常多的网络环境，当然，还有项目进度的步步紧逼。对于运维的功能，Ansible、Puppet、SaltStack 都是非常不错的选择，但是偏偏在我们所要面对的环境下用起来实在太困难。在和同事讨论之后，偶然发现 OSGi 这种技术能够解决我们的问题，所以也就自己重新造了一个轮子来解决我们所面临的问题。

项目进行得差不多的时候，我觉得 Apache Karaf 与 Apache ActiveMQ 这两种技术整合起来所设计的运维框架也有它的一些优点，于是在 OSChina 上写了一篇博客与大家分享。非常巧的是，电子工业出版社的编辑通过我的博客联系上了我，希望我可以写一本关于采用 Apache Karaf 与 Apache ActiveMQ 整合所设计的运维软件的书籍，这让我感到非常荣幸。

我把当初设计这套软件的思路以及一些需要注意的要点写在了这本书中，希望我所分享的内容能够对运维的小伙伴们有所帮助。

本书面向的读者

本书面向的读者是从事系统运维的开发人员，希望能够给读者在设计运维软件的时候提供一种不同的思路。书中的内容以思路分享居多，因为笔者认为如今的互联网非常发达，某个功能如何实现，我们搜索一下就会找到很多方案，而思路的分享更能引起读者与笔者在思想上的碰撞，在碰撞中让读者发现一些其他的方法。

内容介绍

本书共 12 章。

第 1 章与读者一起探讨什么是自动化运维。

第 2 到第 4 章简单介绍目前比较热门的集中化运维软件 Ansible、Puppet 和 SaltStack。

第 5 章介绍为什么在有这么多集中化运维软件的情况下我们还需要重复造一个轮子。

第 6 章和第 7 章介绍重复制作轮子所需要的一些技术——Apache Karaf 和 Apache ActiveMQ。

第 8 章到第 10 章介绍如何使用 Apache Karaf 和 Apache ActiveMQ 制作出一个可插拔式的集中化运维框架。

第 11 章介绍如何与 Zabbix 进行整合。

第 12 章与读者分享了一个小故事，希望通过这个小故事让读者能够更加了解这款运维软件所要解决的问题。

致谢

- 感谢我的同事陈自欣，我非常佩服他在技术知识面上的广度，采用 OSGi 的技术来开发运维软件的思路就是他提出来的。
- 感谢我的同事崔威，在我刚工作的时候教会了我许多软件开发的技术，让我在后续的技术发展道路上少走了许多弯路。
- 感谢我的家人，给了我这么多的时间让我可以专心地完成本书的写作。

吴文豪

2015 年 5 月

目 录

第1章　什么是自动化运维

1.1　硬件运维和软件运维

运维，指的是对一些已经搭建好的网络软 / 硬件进行维护。

在笔者实际从事运维开发这一工作之前，只要提到运维工程师，脑海里面就会出现这么一个场景：在一个有几百台设备发出“嗡嗡”声的机房里面，一个身穿白衬衫、带着眼镜的帅气小伙正在对一台电源出故障的设备进行故障排查（曾经简单地认为这就是运维的全部内容）。

从事运维开发工作之后，发现运维也是有细分领域的，有负责业务运维的、数据库运维的、主机运维的，等等。虽然运维的工作被细分成这么多种，但是粗略地看，运维主要分为硬件运维和软件运维两大块。无论是哪一种运维，它的最终目的都是为了让应用系统能够稳定地运行，更好地为企业提供服务。

1.1.1　小故事之一——电脑专家

小明对电脑的硬件非常感兴趣，家里的电脑都不知道被他拆了又装、装了又拆多少遍了，年纪轻轻的他就已经懂得了各种电脑问题的维修技巧。这不，在大妈圈里面，小明可是一个被传得神乎其神的电脑专家。按照大妈们的话来说，无论你的电脑出了哪方面的问题，只要找到小明，问题都能迎刃而解。

“咚咚咚”，小明听到三声敲门声之后打开了门，看到一脸焦急的小菜。还没等小明开

口，小菜就说到：“小明哥啊，赶紧帮忙救个火啊，我在学校负责维护的一套选课系统今天在选课的时候突然变得很慢了，整个系统慢到几乎是不可用的状态了，平时我也就只会装个杀毒软件杀个毒，每周重启一下电脑，但这次不管怎么弄就是没效果。我听我妈妈说你在电脑方面很厉害，你能不能帮我检查一下，给我提供点建议？”小明一听就傻眼了……

1.1.2 小故事之二——你居然不会修电脑

小菜是某大学计算机专业的学生，在班里可是出了名的计算机高手，不论是C语言、Java程序设计，还是数据库这些科目，小菜在班里的考试成绩都是排名第一。而且在学长的指导下，小菜在系统运维方面的能力也十分了得，学校的教务系统、班上的点名系统都是小菜在负责维护，只要他发现系统出现运行缓慢或者是无法打开的现象，不出半天他就能漂亮地把问题解决。

在班里计算机成绩这么出众，带来的结果必定只有一个，那就是有女同学会叫你帮忙修电脑。最近小菜就碰到这么个烦心事，有个师妹的电脑出问题了，不知道为什么就是开不了机，于是就找到了小菜，希望小菜能帮她把电脑修好。小菜去师妹宿舍之后捣鼓了半天，却没定位出电脑开不了机的问题是因为主板损坏。于是师妹忍不住在旁边小声嘀咕：“还说是什么计算机高手，连修个电脑都不会”。

1.1.3 硬件运维与软件运维

硬件运维可以理解为对基础设施的运维，包括机房的设备和电脑设备。比如对机房的备用电源、空调，主机上的硬盘、内存这些物理设备的维护，都可以看作硬件运维。

软件运维包括系统运维和应用运维。系统运维指的是对操作系统、数据库、中间件等的监控和维护，这些系统是介于设备和应用之间的。应用运维指的是对业务系统的运维，例如企业内部的OA、ERP等业务系统。

前面我们讲到小明作为一个对硬件知识了解得比较深入的中国好邻居，却解决不了小菜的燃眉之急。第二个小故事里面的小菜虽然对软件维护方面的知识有很深了解，但是却解决不了师妹的电脑开不了机的硬件故障。运维的知识很广阔，硬件层面的运维与软件层面的运维所需要了解的知识点差别还是比较大的。

所以本书后续所描述的自动化运维中的“运维”，指的都是软件运维，也就是系统层面的运维和应用层面的运维。

1.2 软件运维的主要问题

1.2.1 设备数量多

在虚拟化发展起来之前，企业的IT建设普遍是把单个应用部署在一套甚至是多套基础设施上进行建设的。直白点讲，就是一个应用会部署在一套服务器上。随着公司内部的IT系统一天一天地增大，运维工程师需要运维的主机数量也随之慢慢地增加。刚开始的时候，运维工程师可能只需要运维十多台设备，一段时间过后，他们需要运维的设备数可能就会翻倍了。给十台主机打补丁、升级软件，估计还受得了，但是给五十台主机打补丁、升级软件，这种重复性的劳动估计就没多少人能受得了了。

有小伙伴可能会问，为什么不把多个应用系统部署在一台主机上呢？这样一方面可以减少由于主机数量的增加而给运维带来更大的负担，另一方面也可以为企业节省成本，何乐而不为呢？我们可以从以下几个角度来看待这个问题。

首先，为了避免被某个软件厂商垄断了企业的系统，一般情况下企业的系统都会交给不同的厂商进行开发。既然是不同的厂商，当多个应用系统被部署到了一起的时候，很容易就会出现这样一个场景：A公司和B公司的系统在同一台主机上，A公司于本周五上了一个新功能，周末的时候出现了主机的I/O负载很高，导致该主机上的业务系统都出现了反应迟钝的现象。于是B公司的开发人员就开骂了，指责A公司的新功能导致主机负载高，这个得扣他们的款。A公司的开发人员也不认输，认为我们的功能绝对没有问题，是你们的系统周末出账用了太多的I/O才导致系统的速度变缓慢的。于是一场扯皮大战就这样开始了。

然后，从保证业务系统的可用性来看，不同的应用系统之间需要在系统层进行隔离。不然一旦操作系统出现问题，部署在操作系统上的所有应用系统都会出现故障，这肯定是达不到企业在业务上的要求的。而且，多套业务系统部署在一台主机上，也会为性能优化、故障排查等后续处理带来许多干扰。

1.2.2 系统异构性大

给运维工程师带来困扰的第二个问题就是系统的异构性。

由于应用系统是由不同开发商开发的，不同开发商内部的技术栈会有差异，所以很容

易会出现 A 开发商开发出的应用系统需要跑在 Redhat 上，Web 服务器需要用 Tomcat，数据库需要用 MySQL，而 B 开发商开发出的应用系统需要用 Windows Server 来承载，Web 服务器需要用 IIS，数据库需要用 SQLServer 的情况。

这个时候运维工程师就傻眼了，这怎么运维？系统出故障了之后还得想想究竟是用 SSH 去维护还是用远程桌面去维护。公司规定每月要有一次常规性的主机重启，究竟哪个 IP 的设备需要用 SSH 去重启，哪个 IP 的设备需要用远程桌面去重启？

1.2.3 虚拟化的成熟带来更大的困难

可能有那么一些运维工程师通过自己多年积累下的脚本度过了一些难关，本以为可以歇一歇了，没想到近年来，随着虚拟化的日渐成熟，又一个挑战来了。

以前企业的设备数量虽然会增长，但是毕竟需要走过一个漫长的企业内部流程才能完成设备的采购和上线。但是随着虚拟化的成熟，企业的 IT 建设再也不需要像以前一样，上一个新的业务系统就经历一个漫长的采购流程了，也不需要再费心思去找个机房放主机了。IT 管理人员只需要申请一台虚拟机，然后再在 CMDB 里面填一下这台主机的信息就可以了。

> CMDB 存储管理企业 IT 架构中设备的各种配置信息，它与所有服务支持和服务交付流程都紧密关联，支持这些流程的运转，发挥配置信息的价值，同时依赖于相关流程以保证数据的准确性。

在这个虚拟化如此“任性”的年代，IT 建设的成本在不断降低，IT 建设的速度也在不断提升，需要运维的设备数量从原来的几百台增加到几千台甚至上万台，而且很有可能这些设备也仅仅是这家企业的一个部门的设备数而已，这给运维工程师带来了更大的挑战。

1.3 运维常用工具

当设备数量很多、设备存在很大的异构性时，我们能不能有一个更加轻松愉快的方式来管理这些主机呢？从竖井式 IT 建设到层次性 IT 建设过程中对运维工程师来说，最大的问题是设备数量的爆炸性增长，原有靠堆人力或者积累脚本的做法已经显得不太可行了。这个时候，我们就需要思考这样一个问题，能不能有一个既可以减少人力成本，又可以可

靠地批量完成集中化运维任务的方法呢？

在设备的架构比较一致的情况下，实现集中化运维并不是什么难事。对于 Windows 的设备，假如我们需要对它们进行集中化的重启，就可以采用 PowerShell 的方式来对设备进行集中化的操作。例如，我们需要对主机进行重启的时候，就可以通过 PowerShell 去调用 WMI 的 API 来完成这个功能，而像一些 msi 程序的发布和安装，就可以通过 AD 域控的方式去完成，总体来说还是挺方便的。

在 Linux 和 UNIX 操作系统下，我们可以通过 SSH+Expect 的方式或者是双机互信后采用 SSH 的方式完成集中化的运维，难度也不大。

但是，由于不同业务系统对可用性和鲁棒性的要求不一样，通常会出现既有 Windows 主机，也会有 Linux 和 UNIX 主机的场景。而这种操作系统的多样性，也给集中化的运维带来不少的麻烦。无论是 AD 域控+WMI 还是 SSH 的方式，都只是解决了一类主机运维的问题。试想一下，现在需要对运维的所有主机进行常规的集中化重启操作，我们还得先去看一下究竟哪些设备是 Windows 操作系统，哪些设备是 Linux 操作系统，然后再去做相应的操作。这种费时费力的操作方式并不是我们想要的，我们更希望有一个统一的入口，我们可以通过这个入口所提供的对外封装好的一些接口去对异构的设备进行集中化运维，如图 1.1 所示。

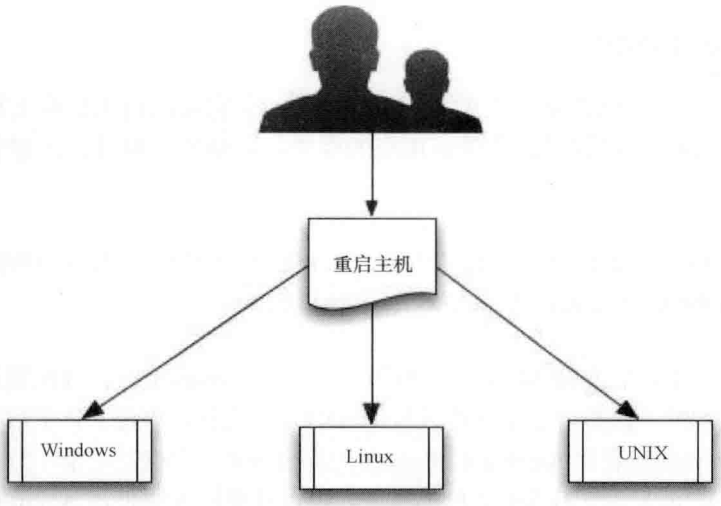


图 1.1　对异构设备进行集中化运维