AADL模型的测试方法研究

王 庚 周兴社 张 凡 董云卫
(西北工业大学计算机学院 西安 710072)

摘 要 近几年来,MDA开发方式的应用使得如何保证模型质量成为研究的热点。以基于模型的测试为研究对象,研究了对AADL模型进行模型测试的方法,并提出了结合马尔可夫链对AADL模型进行测试的框架以及实施方法。最后,通过示例进一步说明了该方法。

关键词 基于模型的测试,结构分析和设计语言,马尔可夫链

Research on Model-based Testing on AADL

WANG Geng ZHOU Xing-she ZHANG Fan DONG Yun-wei
(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract As MDA becomes a popular software development methodology, it is one of the hot issues to ensure the quality of software which is based on models. This article studied on model-based testing(MBT) focusing on AADL model, and an algorithm was proposed to carry out model-based testing on AADL with markov chain. An example was given at the end of the article to demonstrate the algorithm.

Keywords Model-based testing, AADL, Markov chain

1 引言

随着计算技术的发展,嵌入式系统在工业领域中扮演着越来越重要的角色。早期嵌入式软件的任务简单明确,同时软件的规模也比较小,因此在软件的设计和实现过程中,关注的重点是如何有效地满足控制软件的功能及非功能性需求,基本不考虑软件工程中控制和管理系统规模及复杂性的方法和技术。随着嵌入式软件系统复杂性不断扩大,早期的嵌入式软件开发方法不能够完全满足嵌入式软件设计需求,因此出现了MDA的软件开发方法。

MDA[1](model driven architecture 模型驱动架构)是由OMG(Object Management Group)定义的软件开发框架。在MDA的开发方法中,软件开发行为被提升到更高的级别——分析模型级,而针对特定计算平台的编码工作由机器自动完成,这样就将业务逻辑与实现技术成功分离,使软件模型成为开发过程中的核心。

UML是目前MDA中应用广泛的一种建模方法,能够对体系结构、对象与对象之间的关系、数据、构件结构及组装关系进行建模。但是UML缺乏对嵌入式软件实时性、可靠性等非功能属性的描述。AADL(Architecture Analysis and Design Language,结构分析和设计语言)是在国际汽车工程师协会(SAE)的指导下开发的,按照构件及构件之间的相互作用对应用系统的结构进行描述与分析,并且支持对嵌入式软件非功能属性的描述。AADL将构件抽象为软件构件、执行平台(硬件)和系统构件等3种类型[2]。

随着AADL在嵌入式软件设计中的应用越来越广泛,如何对AADL模型进行测试成为保证AADL模型质量的关键。本文主要研究了结合马尔可夫链模型对AADL模型进行测试的方法,设计了一种面向AADL模型进行测试的框架。本文第2节对基于模型的软件测试背景进行了说明;第3节提出了一种面向AADL模型的模型测试方法;第4节结合示例说明了上述方法;最后对全文进行了总结。

2 相关研究

模型驱动的开发模式(MDA)的发展促进了以模型为中心的软件测试方法和技术的发展,这种方法称为基于模型的测试。

基于模型的测试分为两类:一类是依赖模型产生测试用例,将产生的测试用例应用于被测系统中;另一类是将模型作为被测对象,实施基于模型的测试的目的就是发现模型中存在的错误,保证模型的质量。本文主要研究第二类基于模型的测试,以下所提的基于模型的测试所指的也是第二类。

基于模型的测试思想依赖于分析期望行为模型的执行路径,并以这些执行路径作为测试用例即:被测模型的输入、期望输出[3]。将此输入应用到被测模型中,得到该输入对应的实际输出,与期望输出比较之后发现模型中存在的错误。由于在软件开发的早期阶段就能实施测试行为并且将测试直接与软件需求相连,因此基于模型的测试对软件产品的行为具

到稿日期:2008-12-22 返修日期:2009-03-10 本文受863项目"构件化嵌入式软件测试方法及其工具研究"(项目号:2008AA01Z142)及航空基金项目(项目号:20081953012)资助。

王 庚 硕士生,主要研究方向为嵌入式系统与嵌入式系统测试,E-mail:gwang_03@hotmail.com;周兴社 教授,博士生导师;张 凡 博士生;董云卫 教授。

有良好的覆盖并且能够有效地降低测试成本。

在学术界和工业界，对基于模型的测试的研究还处于初期阶段。在学术界，主要的基于模型的测试工具有 MIT 的 MulSaw[6]项目，它有两种不同的测试用例产生工具，TestEra 支持 Alloy 建模语言，KORAT 支持 JAVA 建模语言(JML)。University of Twente 的 TorX[7]能够通过模型仿真，分析系统规格描述的行为和模型行为的差异对模型进行测试。这些工具对嵌入式领域中广泛应用的 AADL 模型不支持，这就需要针对 AADL 开展模型测试方法研究。在工业界，主要的基于模型的测试工具有 Conformiq Software Ltd. 公司的产品 CONFORMIQ TEST GENERATOR[4]，该工具以 UML 模型为被测对象，并且能够支持实时属性。Reactive Systems Inc 公司的产品 REACTIS[5]，该工具支持使用 SimuLink 和 StateFlow 建立的模型，不仅能够自动产生测试用例而且能够通过分支、状态、转移等多种方式判定测试用例对模型的覆盖率。

3 面向 AADL 模型的测试方法

AADL 模型通过构件对软件系统结构进行描述，在 AADL 模型中软件系统的状态通过构件的“模式”表现。为了对 AADL 模型描述的软件系统状态以及状态迁移进行测试，我们根据基于模型的测试的思想，结合 AADL 模型的特点，提出了将 AADL 模型中构件的“模式”及“模式”迁移映射为马尔可夫链模型(MC)中的状态及状态迁移，并将此 MC 作为被测模型，应用测试用例对其进行测试，从而达到测试 AADL 模型构件“模式”及“模式”迁移的目的。

3.1 测试总体思想

如图 1 所示，软件的需求规格是对软件期望行为的描述，根据需求规格建立用于描述软件期望行为的马尔可夫链模型，这个马尔可夫链模型称为期望马尔可夫链，其状态迁移概率根据软件早期的版本或者软件功能特点得出，属于先验概率。根据期望马尔可夫链的状态迁移概率随机生成 N 组输入序列，这 N 组输入序列组成了测试用例。

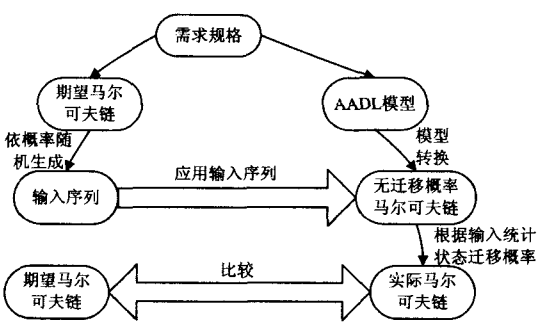


图 1 面向 AADL 的测试结构图

另一方面，软件设计人员根据需求规格设计了 AADL 模型。使用一定的约束规则及转换策略将 AADL 模型转换为马尔可夫链，此马尔可夫链称为无迁移概率马尔可夫链，此时状态迁移概率未知。

将第一步生成的 N 组测试用例输入至无迁移概率马尔可夫链中，每一组测试用例产生相应的状态迁移，N 组测试用例输入之后，根据数学统计方法计算每个状态迁移的概率。至此，得到具有状态迁移概率的实际马尔可夫链。由于期望马尔可夫链描述的是软件的期望行为，实际马尔可夫链描述的是软件的实际行为，因此通过比较期望马尔可夫链与实际马尔可夫链的差异，可以发现实际马尔可夫链中的错误。检测出错误之后，可由实际马尔可夫链回归至 AADL 模型进行错误分析。

3.2 建立期望马尔可夫链模型(软件的期望行为)

马尔可夫链是一种数学分析方法下的统计模型。一般地，一个马尔可夫链模型可以描述为五元组：$\langle S,\Sigma,S_0,F,Q\rangle$[8]，其中，$S$ 是系统的状态空间；一般情况下假定 S 是一个有限的可列的集合，其元素由 S_i,S_j,S_k 等形式表示。Σ 为转移标识集合，表示为三元组 $\Sigma=\langle S\times S,E,P\rangle$，$S\times S$ 表示状态到状态的转移，E 为引起系统内各状态转移的事件的集合，表示为 $[e_{ij}]n\times n$，其中 e_{ij} 表示引起状态 S_i 至 S_j 转移的事件，P 是系统的状态之间的转移概率矩阵，表示为 $[P_{ij}]n\times n$，其中 P_{ij} 表示状态 S_i 转移至状态 S_j 的概率，n 表示系统状态数目。

软件的需求规格作为判定软件是否合格的标准，必须对软件的行为尽可能详细地描述，并且应该避免出现二异性。根据需求规格建立的期望马尔可夫链应是对需求规格正确的反映。首先，由需求规格得到软件的状态集合以及各状态之间的转移关系。其次根据软件之前的版本或者原型确定状态转移概率矩阵 P。如果无早期版本，则需根据经验对转移概率进行估计。这样，就完成了期望马尔可夫链模型的建立。

3.3 建立实际马尔可夫链模型(软件的实际行为)

由软件开发人员设计的 AADL 模型对构件的状态以及状态之间的转移进行了描述。根据 AADL 模型建立实际马尔可夫链模型的方法如下。

(1)根据期望马尔可夫链 $M_e\langle S,\Sigma,S_0,F,Q\rangle$，其中各状态以及状态转移概率已知，依概率由初始状态到终止状态随机生成 N 组序列。记为 $\theta_i,i\in[1,N]$，θ_i 表示为马尔可夫链模型中 Σ 的子集，这 N 组序列组成了测试用例。如图 2(a)所示，该图是由需求规格建立的期望马尔可夫链模型，依概率生成总计 N 组输入序列，其中，$\{e_{0,1}\}$ 生成概率为 $p_{0,1}$，$\{e_{0,2}\}$ 生成概率为 $p_{0,2}$，$\{e_{0,3}\}$ 生成概率为 $p_{0,3}$。

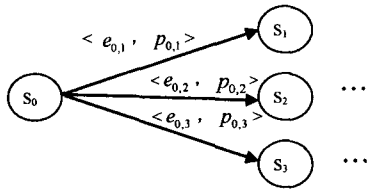
(2)AADL 模型的构件能够描述不同的“模式”以及“模式”之间的迁移，这些模式的本质是不同的状态以及状态间的迁移，因此将 AADL 构件中的模式以及模式迁移映射为马尔可夫链模型中的状态以及状态之间的迁移，这里状态的转移概率未知，如图 2(b)所示，该图为由 AADL 模型得到的无概率马尔可夫链模型。

(3)将测试用例 $\theta_i,i\in[1,N]$ 依次输入(2)得到的无概率马尔可夫链模型中，用 $[fre_{ij}]n\times n$ 表示由状态 S_i 转移到状态 S_j 的频次。此处分 3 种情况进行说明：

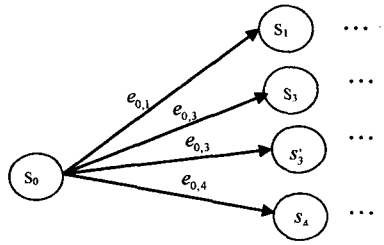
• 第一种情况，在期望马尔可夫链和无概率马尔可夫链中均存在相同的 $e_{i,*}$ 使得 $s_i\xrightarrow{e_{i,*}}$。$s_i\xrightarrow{e_{i,*}}$ 表示在事件 $e_{i,*}$ 的作用下，状态 s_i 可以转换为其它状态。此时若转换后的状态相同，如图 2(a)、图 2(b)所示 $s_0\xrightarrow{e_{0,1}}s_1$ 和 $s_0\xrightarrow{e_{0,3}}s_3$，每当在一个测试用例下发生此种状态转移，相应 fre_{ij} 的值加一。如图 2 所示，$fre_{0,1}$，$fre_{0,3}$ 各自加一，当前测试用例继续。若转换后的状态不同，如图 2(a)、图 2(b)所示 $s_0\xrightarrow{e_{0,3}}s_3'$，此时，当前测试用例终止。

• 第二种情况，在期望马尔可夫链中存在的 $e_{i,*}$ 事件使

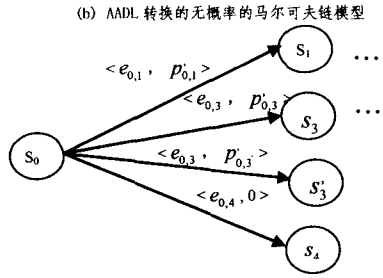
得 $s_i \xrightarrow{e_{i,*}}$，在无概率马尔可夫链中不存在。如图 2(a)所示的 $s_0 \xrightarrow{e_{0,2}} s_2$，在无概率马尔可夫链模型中并无此转换，此时对应的 $fre_{0,2}$ 记为 0，当前输入序列终止。



(a) 期望马尔可夫链模型



(b) AADL 转换的无概率的马尔可夫链模型



(c) 完整的实际马尔可夫链模型

图 2　基于 AADL 软件架构的马尔可夫链模型

• 第三种情况，在无概率马尔可夫链中存在的 $e_{i,*}$ 事件使得 $s_i \xrightarrow{e_{i,*}}$，在期望马尔可夫链中不存在。如图 2(b)所示的 $s_0 \xrightarrow{e_{0,4}} s_4$，由于在期望马尔可夫链中不存在使 s_0 转移为 s_4 的事件，故当测试用例输入至无概率马尔可夫链时，$s_0 \xrightarrow{e_{0,4}} s_4$ 这一转移发生的频次为 0。并且在期望马尔可夫链模型中无该转移，因此该转移频次不计入矩阵$[fre_{ij}]n\times n$ 中。

当 N 组测试用例依次输入至无概率马尔可夫链模型之后，得到状态转移的频次矩阵$[fre_{ij}]n\times n$，经过计算可得到实际马尔可夫链各状态转移概率矩阵$[P'_{ij}]n\times n$，至此生成完整的实际马尔可夫链模型，如图 2(c)所示。

3.4　发现模型错误

期望马尔可夫链是软件的理想执行行为模型，一般认为不包含错误；而实际马尔可夫链是软件的实际执行行为模型，是特定的测试条件下的执行，可能包含一些错误。比较这两个模型的差异是进行测试的有效方法。由于期望模型和实际模型各自的概率转移矩阵都已经得到，可以应用数学统计方法分析期望马尔可夫链模型与实际马尔可夫链模型的差异，从而发现实际模型中的错误。

这里比较期望马尔可夫链模型和实际马尔可夫链模型的三元组 $\Sigma=\langle S\times S,E,P\rangle$。使用数理统计的方法比较两个转移概率 $p_{i,j}$ 和 $p'_{i,j}$，一种计算的方法为[9] $D(U,T)=\sum_{ij}\pi_i p_{ij}\log_2\frac{p_{ij}}{p'_{ij}}$。其中，$U$ 为期望马尔可夫链模型，T 为实际马尔可夫链模型。π 是期望模型 U 的平稳分布，$p_{i,j}$ 为 U 中 s_i 转移到 s_j 的概率，$p'_{i,j}$ 为相应的 T 中的相关状态间的转移概率。理想状况下当且仅当 $p_{i,j}=p'_{i,j}$（对于所有的状态迁移），$D(U,T)=0$，此时说明 U 中的所有状态转移都在 T 中以相同的概率发生转移。在测试进行之前，根据测试经验为 $D(U,T)$ 设定一阈值 d，当 $D(U,T)>d$ 时，认为此次测试的结果不合格，当 $D(U,T)<d$ 时，认为实际模型实现了期望模型中描述的软件行为，在该组测试用例 θ_i，$i\in[1,N]$ 下，实际模型中没有发现错误。

4　范例-家居控制系统

Internet 与网络得到迅速普及应用，并向家庭领域不断扩展使嵌入式系统越来越普及，从而促进了普适计算的发展。

数字家庭是普适计算应用的一个典型。数字家庭系统是现代化家居管理嵌入式系统。智能窗帘控制系统是数字家庭系统的一个子系统。在这个系统里，一个百叶窗帘由两组电机控制，其中一组负责窗帘整体的拉动，另一组负责窗帘活页的张开与闭合。百叶窗窗帘的状态 S_{car} 有 3 种：打开(0)，关闭(1)，半开(0.5)。百叶窗叶片的状态 S_{leaf} 有两种：闭合(0)，张开(1)。百叶窗的状态是由窗帘的状态与叶片的状态联合的二元组$\langle S_{car},S_{leaf}\rangle$。

约束条件：百叶窗窗帘只有在叶片打开的时候才能够活动，并且窗帘不可能在叶片闭合的时候关闭。因此$\langle 1,0\rangle$是一个错误的状态。

根据需求规格建立期望马尔可夫链模型如图 3 所示。马尔可夫链模型的状态空间为 $S:\{S_0,S_1,S_2,S_3,S_4\}=\{\langle 0,0\rangle,\langle 0,1\rangle,\langle 0.5,1\rangle,\langle 0.5,0\rangle,\langle 1,1\rangle\}$。

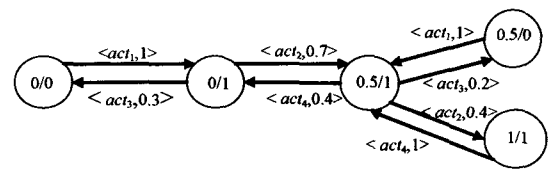


图 3　智能窗帘控制系统期望马尔可夫链模型

状态转移标识集合 $\Sigma=\langle S\times S,E,P\rangle$，事件集合 E 为：

act_1：打开百叶窗叶片

act_2：拉开半张百叶窗窗帘

act_3：闭合百叶窗叶片

act_4：关闭半张百叶窗窗帘

其中，状态转移概率 P 根据经验得到，属于先验概率。期望马尔可夫链模型的状态转移信息如表 1 所列。

表 1　期望马尔可夫链模型信息表

起始状态	事件编号	事件描述	目标状态	转移概率
S_0	$e_{0,1}$	act_1	S_1	1
S_1	$e_{1,2}$	act_2	S_2	0.7
S_1	$e_{1,0}$	act_3	S_0	0.3
S_2	$e_{2,3}$	act_3	S_3	0.2
S_2	$e_{2,4}$	act_2	S_4	0.4
S_2	$e_{2,1}$	act_4	S_1	0.4
S_3	$e_{3,2}$	act_1	S_2	1
S_4	$e_{4,2}$	act_4	S_2	1

设计人员根据需求规格建立的 AADL 模型如图 4 所示。图中实线平行四边形表示 AADL 中的构件“进程”，虚线平行四边形表示构件“线程”，虚线圆角矩形表示构件“线程组”。六边形表示进程中的模式。如图 4 所示，在进程中包含 7 个子构件，其中一个为主线程，其余的为线程组。图中每个线程

组通过事件端口控制进程状态之间的迁移。在图中将错误的状态和状态间的转移用灰色线表示以便区别。

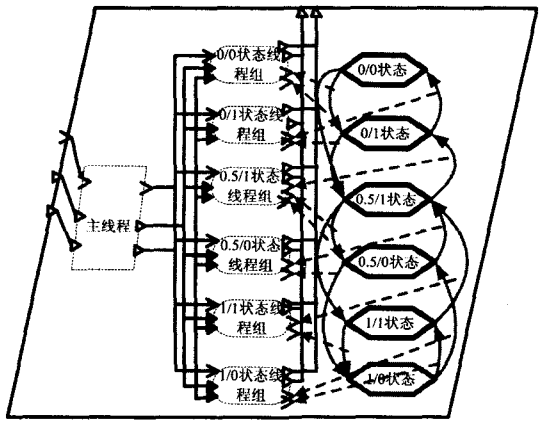


图 4 智能窗帘控制系统 AADL 模型

根据 AADL 模型转换的无概率马尔可夫链模型如图 5 所示。

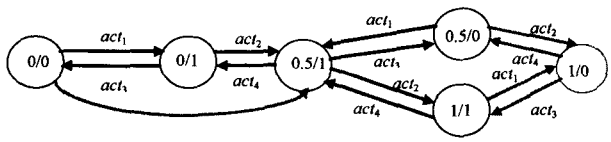


图 5 智能窗帘系统无概率马尔可夫链模型

根据图 3 的期望马尔可夫链，依概率产生 50 组输入序列，这些输入序列组成了测试用例，并结合以往测试经验设定该被测模型的 $D(U,T)$ 的阈值 $d=0.03$，根据上述算法计算得出实际马尔可夫链模型信息表如表 2 所列。

表 2 实际马尔可夫链模型信息表

起始状态	事件编号	事件描述	目标状态	转移概率
S_0	$e_{0,1}$	act_1	S_1	0.8
S_1	$e_{1,2}$	act_2	S_2	0.65
S_1	$e_{1,0}$	act_3	S_0	0.35
S_2	$e_{2,3}$	act_3	S_3	0.25
S_2	$e_{2,4}$	act_2	S_4	0.4
S_2	$e_{2,1}$	act_4	S_1	0.35
S_3	$e_{3,2}$	act_1	S_2	1
S_4	$e_{4,2}$	act_4	S_2	1

结合表 1，表 2 使用公式 $D(U,T)=\sum_{ij}\pi_i p_{ij}\log_2\frac{p_{ij}}{p'_{ij}}$，计算分析实际模型与期望模型之间的差异，得到 $D(U,T)=0.0652$。设定的阈值 $d=0.03$，$D(U,T)>d$，说明实际模型与期望模型的差异较大，实际马尔可夫链模型中存在错误，即由此映射的 AADL 模式转移有错误。

结束语 本文首先介绍了基于模型的测试的基本概念和基本原理，进而提出了一种结合马尔可夫链模型对 AADL 模型进行测试的方法，并给出示例对这种方法进行了说明。由于 AADL 模型没有对状态迁移概率进行描述，因此应用本文的测试方法，能够更加精确地从数学统计角度对测试结果进行分析。由于对基于模型的测试的研究尚处于早期阶段，因此对 AADL 模型进行测试还有很多问题需要解决，比如测试用例的生成及优化，测试的覆盖率问题，测试自动化方法的研究以及测试结果的分析评估等，这些都需要进行进一步的研究。

参 考 文 献

[1] OMG. Model Driven Architecture(MDA)[OL]. http://www.omg.org/mda/,2002

[2] Feiler P H,Gluch D P,Hudak J J. The Architecture Analysis & Design Language(AADL):An Introduction[M]. Carnegie Mellon University,2006

[3] Pretschner A. Model Based Testing[C]// ICSE'05. St. Louis, Missouri,USA,May 2005

[4] Conformiq Software Ltd. Conformiq Test Generator[OL]. http://www.conformiq.com/products.php

[5] Reactive Systems Inc. Reactis[OL]. http://www.reactive-systems.com/

[6] MIT Software Design Group. MulSaw[OL]. http://projects.csail.mit.edu/mulsaw/

[7] University of Twente, Formal Methods and Tools research group. TorX[OL]. http://fmt.cs.utwente.nl/tools/torx/introduction.html

[8] 陈祎. 基于 UML 模型的软件测试技术研究与实现[M]. 成都：电子科技大学，2006

[9] Whittaker J A, Thomason M G. Senior-Member-, IEEE A Markov Chain Model for Statistical Software Testing[J]. IEEE Transactions on Software Engineering,1994,20(10)

(上接第 126 页)

的磁盘空间利用率。两层之间的动态数据迁移是下一步研究的重点。

参 考 文 献

[1] Patterson D,Katz R. A case forredundant arrays of inexpensive disks (RAID). Chicago,1988:109-116

[2] Blaum M. EVENODD:An optimal scheme for tolerating double disk failures in RAID architectures[J]. IEEE Transactions on Computers,1995,44(2):245-254

[3] Blaum M. New array codes for multiple phased burst correction [J]. IEEE Transactions on Information Theory,1993,39(01):66-77

[4] Hafner J L. Hover erasure codes for disk arrays[R]. RJ10352 (A0507-015). IBM Research Division,July 2005

[5] Hafner J L. WEAVER codes:Highly fault tolerant tolerant erasure codes for storage systems[C]// Proceedings of the FAST-2005:4th Usenix Conference on File and Storage Technologies. December 2005

[6] Bloemer J M,Kalfane M,Karpinski R. An XOR-based erasure-resilient coding scheme[R]. ICSI TR-95-048

[7] Alvarez G A,Cristian F. Tolerating multiple failures in RAID architectures with optimal storage and uniform declustering. Colorado,1997:62-72

[8] Jin Fan. An investigation on new complex rotary codes. Brighton,UK,1985:1-8

[9] Feng Dan,Zhang Jiang-ling. Improved EVENODD code. Ulm, Germany,1997:261-262

[10] Dong Huan-qing,Li Zhan-huai. RAID-VCR:A New RAID Architecture for Tolerating Triple Disk Failures[D]. School of Computer Science,Northwestern Polytechnical University,Xi'an

[11] Wikes J,Golding R,Staelin C,et al. The HP AutoRAID Hierarchical Storage System. Hewlett-Packard Laboratories