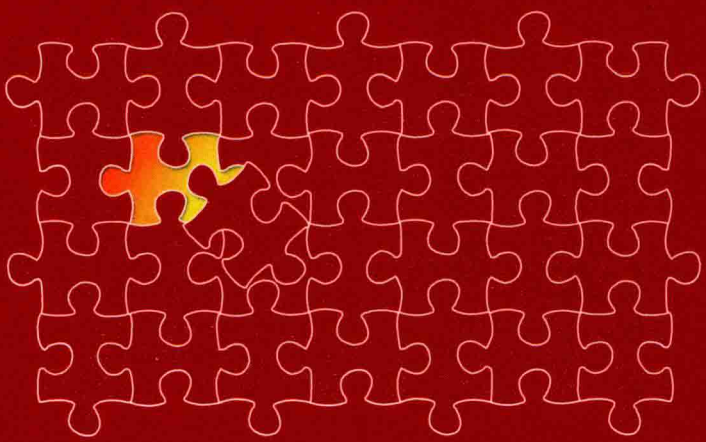
装备科技译著出版基金

ELSEVIER
爱思唯尔

容错系统

Fault-Tolerant Systems

【美】Israel Koren　C.Mani Krishna　著
杨志　唐宏　王海龙　张杰　译



國防工業出版社
National Defense Industry Press

装备科技译著出版基金

容 错 系 统

Fault – Tolerant Systems

[美] Israel Koren　C. Mani Krishna　著
杨 志　唐 宏　王海龙　张 杰　译

国防工业出版社
·北京·

著作权合同登记　图字:军-2011-013号

图书在版编目(CIP)数据

容错系统/(美)科伦(Koren,I.),(美)克里希纳(Krishna,C. M.)著;杨志等译. —北京:国防工业出版社,2015.5

书名原文:Fault-Tolerant Systems

ISBN 978-7-118-10075-4

Ⅰ.①容… Ⅱ.①科… ②克… ③杨… Ⅲ.①容错系统 Ⅳ.①TP302.8

中国版本图书馆CIP数据核字(2015)第108800号

Fault-Tolerant Systems

Israel koren and C. Mani Krishna

ISBN:978-0-12-088525-5

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路23号　邮政编码100048)

北京京华虎彩印刷有限公司印刷

新华书店经售

*

开本 710×1000　1/16　印张 19¼　字数 358千字

2015年5月第1版第1次印刷　印数 1—1000册　定价 86.00元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777　　发行邮购:(010)88540776

发行传真:(010)88540755　　发行业务:(010)88540717

序

在医疗卫生、商业贸易、交通运输、公共事业和国家安全等一些关键领域中，应用系统必须具备高度的安全性。在这些领域中广泛使用的计算机系统和其他电子系统要具有高可靠性。这些系统的高可靠性是通过容错设计来获得的。尽管设计容错系统的研究最初来自于对计算机高可靠性的需求，超大规模集成电路和系统的制造商也需要使用容错设计方法用以提高产品率。

这是由于随着超大规模集成电路特征尺寸的减小和在制造过程中所使用的平版印刷技术的缺陷，制造的器件变得不稳定。另外，器件的小尺寸使得它们容易受到辐射影响导致运行时错误。因此，即使在像消费电子这样的非关键应用系统中，也可能需要使用容错技术。

本书包含了软硬件容错设计，如何使用容错技术提高成品率、进行网络系统的分析和设计等广泛的内容，以及如何保护用于安全目的的加密子系统方法的资料。书中的资料可以帮助电子与计算机工程和计算机科学领域的广大的学生和从业者学习如何设计可靠的计算系统和如何分析容错计算系统。

Sudhakar M. Reddy
电子与计算机工程特聘教授
爱荷华大学基金会
爱荷华市，爱荷华州

赞　　誉

最近在智能卡业，故障攻击成为一个严重的问题。《容错系统》向读者清晰地阐述了这些攻击以及如何阻止这些攻击的保护策略，因此是该领域从业者和研究人员的必读物。

——David Naccache，巴黎高等师范学校

某一个领域的原理，无论是高尔夫球还是容错，都是立足这个领域需要掌握的预备知识。Krishna 和 Koren 合著的这本书向读者阐述了容错的基本原理。本书特别的及时，因为容错计算部分的设计，如处理器和磁盘，对主流计算业变得更加重要。

——Shubu Mukherjee，Intel 公司 FACT－AMI 集团总监

Koren 和 Krishna 教授最近编写的这本书具有双重目的，它首先描述了软件和硬件层面的不同冗余类型的基本容错工具，然后介绍了最新研究方向。它回顾了基本可靠的建模方法、组合块和马尔科夫链技术，并在关于统计仿真方法的章节中提供了实践评估指引和容错指南。书中所有的章节都有清晰的阐述，包括说明案例，其中有大量的参考文献列表，因此学生可以在几乎所有的主题中深入钻研。包含容错的几个实践和商务计算系统也有详细的阐述。此外，在密码系统和超大规模集成电路设计纠错两个章节中介绍了容错研究领域面临的最新挑战。

——Robert Redinbo，加州大学戴维斯分校

容错计算领域在过去十年里取得了长足的发展，然而该领域还没有一本综合这些进展的适合学生的启蒙读物。这是最近十年我了解的第一本关于容错计算软件和硬件方面的书，内容详实，并且把研究内容编写成了适合课堂使用的教科书。

——Kewal Saluja，麦迪逊市威斯康星大学

前　言

本书的目的是为广泛的容错计算研究领域提供一个全面详实的介绍。作为教科书，本书适用于高年级大学本科生和一年级研究生，也可为本行业的工程师提供参考。由于无法在一本书中涵盖全部已开发或当前使用的容错技术和实践经验，本书主要介绍该领域的基本知识以及丰富的背景资料，以帮助读者更好地查取容错研究领域迅速增长的文献资料。如果读者想了解更多详细内容，可以查阅本书每章末尾处列出的参考文献。为了能更好地理解本书的内容，读者应具备硬件及架构设计、软件开发原理以及概率论有关的基本知识。

本书共分 10 章，几乎每章有一个相关的参考文献以及一套练习题。练习题的答案可以通过网上获取，也可以由本书的授课教师向出版商联系获取。另外，也可以为教师提供相关的 Powerpoint 幻灯片。

本书第 1 章是基础性知识概述。紧接着的第 2 ~ 7 章构成本书的核心部分，这通常也是在其他容错系统的介绍中应包含的内容。

第 2 章介绍硬件容错，它是历史最悠久的学科（实际上，把硬件冗余应用于容错领域的思想正是史上最著名的计算机科学先驱冯·诺伊曼提出来的）。本章也介绍了可靠性测量分析中所使用的一些概率论工具。

第 3 章介绍信息冗余，主要专注于误差检测及校验码。这些校验码技术跟硬件容错一样，也具有比较长的历史，主要是为了解决信息传输中的错误而提出的。目前相同或类似的技术也在现代存储电路等其他应用中广泛使用。本章只是对最重要的编码技术的概览，而没有综合全部编码技术，因为那样会涉及太多的内容。接着分析存储器中信息冗余管理问题，最后介绍了基于算法的容错技术。

第 4 章内容为容错网络。随着处理器越来越便宜，分布式系统的使用也更为常见，本章介绍了一些关键的网络拓扑结构，并对如何量化和增强其容错能力作了分析。

第 5 章描述了软件容错技术。人们普遍认为，软件是造成当今计算机系统里大多数故障的原因。作为一个研究领域，软件容错没有硬件或信息冗余的容错技术成熟，这类问题也更难处理。软件很可能是人类创造的最复杂的结构体，其容错更是一项艰巨的任务。本章介绍诸如恢复块和 N 版本编程之类的技术，并讨论了验收试验以及为软件故障处理建立模型的分析方法。

第 6 章介绍了利用校验点的时间冗余的使用。大量的硬件故障是暂时的，换句话说，这些故障经过一段时间会消失。对这类故障明显的处理方式是回退执行过程，并重新执行程序。通过校验点技术，可以限制重新执行程序的范围。

第 7 章内容包含几个案例研究，是本书核心部分的完美收尾。本章描述了几个容错系统的实际案例，阐述了上述章节中提到的各种技术的使用情况。

本书其余 3 章介绍了比较专业的课题。

第 8 章介绍了超大规模集成电路缺陷容错。随着芯片尺寸的增加以及形体尺寸的减小，能够容忍超大规模集成电路制造缺陷同时又不影响其功能正变得日益重要。本章讨论了所使用的关键方法，以及基本的数学模型。

第 9 章专门介绍了密码设备。由于商业上对计算机的使用日益增加，包括智能卡和网上购物，刺激了加密技术在日常应用领域的使用。因此，向密码设备植入差错然后观察相关的输出，是一种攻击安全系统并获取其密钥的有效方法。本章介绍了差错检测法在应对此类安全攻击中的使用原理。

第 10 章是本书的结束篇，介绍了仿真以及实验技术。模拟一个容错系统并测量其可靠性，通常需要非常大的计算量。本章概述了基本的仿真技术，以及可以加快仿真速度的方法。另外本章也提供了可以用于分析仿真输出的基本统计工具，并给出了实验差错植入技术的概述。

辅助专用网站 www. ecs. umass. edu/ece/koren/FaultTolerantSyst - ems/ 上含有关于本书的其他学习资源，例如，教学幻灯片和不可避免的错误列表，以及更重要的，关于大量教学工具及仿真程序的广泛链接，这些可能对本书读者具有非常大的帮助。艾思唯尔(Elsevier)也提供了一个教师网站，为把本书作为教科书的教师提供习题答案。有关网址可以在网站 http://textbooks. elsevier. com 上查到。

致　　谢

在本书的出版过程中，我们得到了很多人的帮助。在此把感谢致予 Zahava Koren，她详细地通读了手稿并提供了很多鞭辟入里的意见，尽管这些问题得到了修正，对于书中尚存的错谬之处，责任均由作者承担。还有很多人审阅了手稿，并给出了很多有价值的反馈意见。这些给予意见的人中有些选择匿名，所以我们不能一一感谢他们。其他署名的人分别为：惠普研究院的 Wendy Bartlett，乔治亚理工学院的 Doug Blough，波士顿大学的 Mark Karpovski，俄勒冈州立大学的 Cetin Kaya Koc，英特尔公司的 Shubu Mukherjee，巴黎高等师范学校的 David Naccache，俄克拉荷马州立大学的 Nohpill Park，普渡大学的 Irith Pomeranz，玫瑰人技术研究所的 Mihaela Radu 加州大学戴维斯分校的 Robert Redinbo，位于麦迪逊的威斯康星大学的 Kewal Salujam，应用安全研究集团的 Jean Pierre Seifert，爱荷华州立大学的 Arun Somani 和卡耐基梅隆大学的 Charles Weinstock。

另外还要感谢摩根考夫曼的全体员工，感谢他们为本书所做的努力与贡献。特别感谢 Denise Penrose 和 Kim Honjo，从本书的技术内容到它的外观设计上，他们花费了很多精力及时间与我们研究和商讨。

目　录

第1章 预备知识

计算机曾经是政府机构和一些大公司专用的计算机器，且价格昂贵。然而在过去的50年间，计算机却从专用的计算机器变成了渗透到人们生活各个方面的日常设施。今天，在人们看得到的地方如桌面台式电脑、笔记本电脑、掌上电脑等，以及人们看不到的各个角落如汽车的关键部件、家庭应用、医疗器械、飞行器、工业制造以及发电供电系统等，都充斥着计算机的身影。计算机系统支撑着世界上大部分金融系统，离开计算机，货币交易、证券（债券）交易、货币市场都将变得不可想象。人类社会越来越希望将计算机应用到那些事关生命财产的重要领域，而这种需求背后的推动因素在很大程度上归功于计算机所带来的更多可能性。然而，当人们越来越依赖于计算机去完成这些至关重要的任务时，人们已经直接或间接地将自己的生命财产全部压在计算机上面了，期望计算机可以毫无差错地完成任务。

计算机（硬件以及在硬件上运行的软件）可能是人类迄今为止所发明的最复杂的系统。随着新技术的不断应用，设计师们试图提高晶体管的集成度，计算机硬件的复杂程度还在不断地增加。相比之下，计算机软件的复杂程度是有过之而无不及，随之带来的问题是发生故障的可能性也在增加。客观地讲，任何硬件或软件（哪怕是很小的一部分）都不是完美无缺的，即使是往返于太空的航天飞机，尽管其软件采用人类所掌握的最先进的技术开发和测试，但是也被证明其中的部分程序存在设计缺陷，这为灾难事故埋下了重大隐患。

为应对这类严峻的挑战，科学家们和工程师们也曾设计出复杂的系统，这些系统可以采用各种工具和技术来减少故障发生的次数。然而，仅仅依靠这些措施并不够，人们需要构建这样一种系统，它认可缺陷的实际存在，并采用特定的技术来容许缺陷的存在，同时也能够提供可接受的服务。基于以上观点，本书讨论与容错相关的内容。

1.1 故障分类

日常用语中，差错/缺陷（Fault）、故障（Failure）和错误（Error）可以相互混用，然而，在容错计算术语上，这些措辞有着截然不同的含义。缺陷（或故障）可以是硬件的毛病或瑕疵，也可以是软件或程序编写中的失误。相比之下，错误则

是缺陷或故障的表现形式。

举个例子来说，有一个加法器，假设它的一路输出始终保持为1，也就是说，不管输入的操作数如何变化，这一路的输出始终为1，这就是一个缺陷，但还不能称之为错误。只有当该加法器被实际应用，且这一路输出本应为0而不是1的时候，这样的缺陷才会导致错误。类似地，软件编程中的缺陷和程序执行中的错误也可以用来解释缺陷和错误的区别。假设有一个计算 $\sin(x)$ 的子程序，然而由于程序编写的失误，该子程序计算的是 $\sin(x)$ 的绝对值而不是 $\sin(x)$。只有当这个子程序被调用且正确的输出为负数时，程序的缺陷才会导致错误的发生。

无论是差错或是错误，它们都会在系统中传播。例如，某个芯片的电源和地发生短路，这也可能会导致附近的芯片失灵；由于某个单元的输出是另一个单元的输入，错误也会传播。反观前面提到的例子，无论是有缺陷的加法器还是计算 $\sin(x)$ 的子程序，它们的错误结果都会被反馈到后级计算中，因此导致了错误的传播。

为了避免此类错误的蔓延，设计师们将“隔离区”纳入到系统中。“隔离区”就像是一条鸿沟，它降低了差错或错误从一个区域传播到另一个区域的概率。举个例子来说，可以设计这样一个“差错隔离区”，隔离区内部电压的最大波动不会影响其他区域，每一个区域都有自己的独立电源。换而言之，设计师们尽可能地将不同的区域进行电气隔离。在本书的后续内容里，将会通过使用冗余单元或程序并在输出端进行表决，“错误隔离区”也可以应用于系统设计中。

硬件差错的分类方式有多种，基于不同的角度，有不同的分类方法。按照差错持续的时间，硬件差错可以分为永久差错、暂时差错和间歇差错。永久差错表示一个零部件将永久报废，烧坏了的白炽灯便是永久差错的一个例子。暂时差错是指某个零部件在一定时间内发生故障，但是一段时间后功能又完全恢复正常。例如电话通话中的随机噪声干扰就是暂时差错。再比如存储单元，其存储的内容因电磁干扰而被错误地改变，存储单元本身是没有损坏的，只是其存储的内容暂时是错误的，只要重新改写存储单元的内容就可消除错误，这也是一种暂时差错。间歇差错是指差错不会完全消除，它时而出现，时而消失。当差错消失的时候，零部件功能正常；当差错出现时，零部件会发生故障。电路连接中的掉线便是间歇差错的一个例子。

按照差错所导致的后果，还可将硬件差错分为良性的和恶性的。如果一个差错直接导致某个单元功能失灵，那么称该差错为良性的，这种差错是最容易处理的。难以处理的是那些潜藏的、产生看似合理实则为错误结果的差错；同样难以处理的还有那些导致零部件表现出诡异行为的错误，它向不同的接收端发送不同的结果。比如某飞行器的高度仪向一个接收端发送1000英尺[①]的高度数

① 1英尺 = 0.3048m。——译者注

据，而向另一个接收端却发送8000英尺的高度数据。这样的差错称为恶性差错（或称拜占庭式差错）。

1.2 冗余类型

容错所做的所有工作就是开发和管理冗余。冗余本质上就是指为达成一定的目标配备多个备用资源，而不是仅仅配备“最小系统”。当故障发生时，可以利用冗余来屏蔽故障或者干脆带着故障工作，从而将系统功能维持在所需要的级别上。

这里主要研究四种冗余类型：硬件冗余、软件冗余、信息冗余和时间冗余。硬件错误通常可以用硬件冗余、信息冗余或者时间冗余来处理，而软件错误通常用软件冗余来处理。

硬件冗余是通过在设计中添加额外的硬件来实现的，通过这种方式可以检测出或者代替有故障的零部件。例如，可以选择双处理器或三处理器方案，而不是单处理器方案，每个处理器完成同样的任务。当采用双处理器方案时，可以检测出单处理器所产生的故障；当采用三处理器方案时，可以通过大数判决来覆盖单个故障处理器的输出。这样的方式称为静态硬件冗余，它的目标就是在第一时间屏蔽故障。硬件冗余的另一种方式叫做动态冗余，当正在工作的零部件发生故障时，备用的零部件就会立即处于工作状态。当然也可以将静态冗余和动态冗余相结合，这种方式称为混合冗余。

硬件冗余可以是简单的硬件复制，也可以是一种复杂的结构，当正在工作的单元发生故障时，备用的单元就立即切换到工作状态。这种结构形式的硬件冗余所带来的经济开销也是巨大的，通常仅仅用在一些关键系统中，在这里开销往往是值得的、合理的。尤其是在处理恶性故障时，大量的硬件冗余则更是必需的。

最著名的信息冗余方式就是检错与纠错编码了。在这种编码中，通过在原始数据比特流中添加额外的比特位（称为校验位），就可以检测甚至纠正数据流中的错误位。而今，检错和纠错编码在存储单元和各种存储器件中广泛应用，从而保护这些器件不被良性错误侵扰。需要说明的是，和所有其他形式的信息冗余类似，检错编码中的冗余数据（即校验位）需要专门的硬件进行处理。

检错码和纠错码在数据通信中也有着广泛的应用，由于噪声的存在，信道常会出现暂时故障。信道可以是相隔很远的两端处理器之间的通信链路（如因特网），也可以是连接局域网处理器之间的局部链路。如果数据通信中的冗余编码仅能检错而不能纠错，那么必要的时候可以将信息进行重传，这称为时间冗余。

除了因为噪声而导致的短暂数据通信错误以外，局域网或广域网还可能会