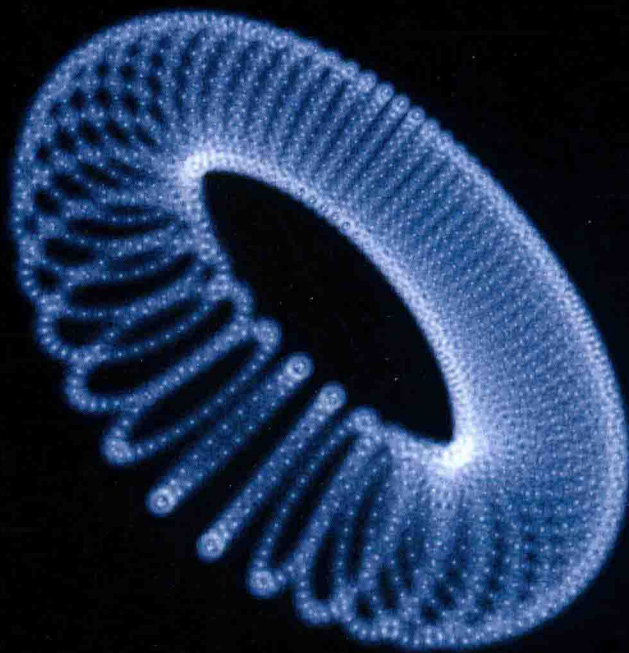


Intel资深架构师的深度分享,“方法+工具+技巧”成就数据中心的问题“终结者”
Linux态势感知创新方法,用数学取代直觉,将系统故障排除的艺术转化为产品



Problem-solving in High Performance Computing
A Situational Awareness Approach with Linux

高性能计算的问题 解决之道

Linux态势感知方法、实用工具及实践技巧

[美] 伊戈尔·卢布希斯 (Igor Ljubuncic) 著

张文力 译



机械工业出版社
China Machine Press



第 1 章 绪论

1.1 高性能计算的发展背景与意义
1.2 高性能计算的定义与范畴
1.3 高性能计算的分类与特点



1.4 高性能计算的挑战与机遇
1.5 本章小结

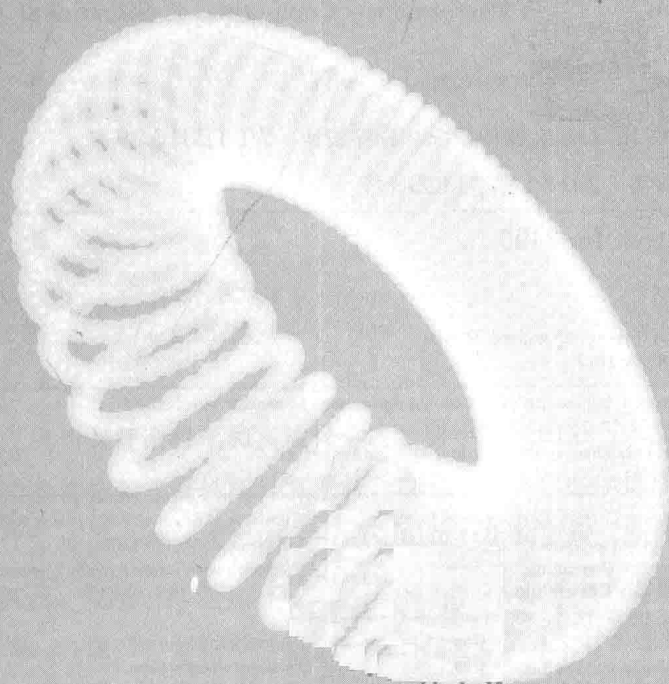
高性能计算的问题 解决之道

吕建明 著

清华大学出版社

清华大学出版社





Problem-solving in High Performance Computing
A Situational Awareness Approach with Linux

高性能计算的问题 解决之道

Linux态势感知方法、实用工具及实践技巧

[美] 伊戈尔·卢布希斯 (Igor Ljubuncic) 著

张文力 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

高性能计算的问题解决之道: Linux 态势感知方法、实用工具及实践技巧 / (美) 伊戈尔·卢布希斯 (Igor Ljubuncic) 著; 张文力译. —北京: 机械工业出版社, 2018.1

(高性能计算技术丛书)

书名原文: Problem-solving in High Performance Computing: A Situational Awareness Approach with Linux

ISBN 978-7-111-58978-5

I. 高… II. ①伊… ②张… III. Linux 操作系统 - 程序设计 IV. TP316.89

中国版本图书馆 CIP 数据核字 (2018) 第 011390 号

本书版权登记号: 图字 01-2016-3496

ELSEVIER

Elsevier (Singapore) Pte Ltd.

3 Killiney Road, #08-01 Winsland House I, Singapore 239519

Tel: (65) 6349-0200; Fax: (65) 6733-1817

Problem-solving in High Performance Computing: A Situational Awareness Approach with Linux

Igor Ljubuncic

Copyright © 2015 Igor Ljubuncic. Published by Elsevier Inc. All rights reserved.

ISBN-13: 978-0-12-801019-8

This translation of Problem-solving in High Performance Computing: A Situational Awareness Approach with Linux by Igor Ljubuncic was undertaken by China Machine Press and is published by arrangement with Elsevier (Singapore) Pte Ltd.

Problem-solving in High Performance Computing: A Situational Awareness Approach with Linux by Igor Ljubuncic 由机械工业出版社进行翻译, 并根据机械工业出版社与爱思唯尔 (新加坡) 私人有限公司的协议约定出版。

《高性能计算的问题解决之道: Linux 态势感知方法、实用工具及实践技巧》(张文力 译)

ISBN: 978-7-111-58978-5

Copyright © 2018 by Elsevier (Singapore) Pte Ltd.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from Elsevier (Singapore) Pte Ltd. Details on how to seek permission, further information about the Elsevier's permissions policies and arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by Elsevier (Singapore) Pte Ltd. and China Machine Press (other than as may be noted herein).

注意

本译本由 Elsevier (Singapore) Pte Ltd. 和机械工业出版社完成。相关从业及研究人员必须凭借其自身经验和知识对文中描述的信息数据、方法策略、搭配组合、实验操作进行评估和使用。由于医学科学发展迅速, 临床诊断和给药剂量尤其需要经过独立验证。在法律允许的最大范围内, 爱思唯尔、译文的原作者、原文编辑及原文内容提供者均不对译文或因产品责任、疏忽或其他操作造成的人身及/或财产伤害及/或损失承担责任, 亦不对由于使用文中提到的方法、产品、说明或思想而导致的人身及/或财产伤害及/或损失承担责任。

Printed in China by China Machine Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong SAR, Macau SAR and Taiwan. Unauthorized export of this edition is a violation of the contract.

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

高性能计算的问题解决之道 Linux 态势感知方法、实用工具及实践技巧

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 曲 熠

印 刷: 北京市荣盛彩色印刷有限公司

开 本: 186mm×240mm 1/16

书 号: ISBN 978-7-111-58978-5

责任校对: 李秋荣

版 次: 2018 年 3 月第 1 版第 1 次印刷

印 张: 16.75

定 价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

高性能计算已成为继理论科学和实验科学之后科学研究的第三大支柱，越来越多的科学发现和工程设计成果都依赖于高性能计算系统。高性能计算已经全面应用到国家安全、政府决策、科学研究、工业设计等诸多层面，典型应用包括生物工程、新药研制、石油物探、运载器设计（航空航天、舰船、汽车）、材料工程、核爆模拟、尖端武器制造、密码研究和各类大规模信息处理等。各国正在积极研制的 E 级（每秒百亿亿次）计算系统有望在 2020 ~ 2022 年出现。近几年来，以电子商务、微信为代表的互联网服务使得万众受益，大规模系统是这些服务的核心基础设施，而系统平均无故障时间（Mean Time To Failure, MTTF）则会随节点数增加而显著降低。互联网服务系统往往包含数以千计的磁盘和服务器，在这种情况下，即便不计成本地使用 MTTF 达到 4 年的设备，在一个拥有 1000 个节点的集群中，也会每周发生近 5 次故障。若适当考虑成本因素而采用 MTTF 为两年的设备，每周的故障次数将达到近 10 次。故障停机对互联网服务企业直接意味着巨额损失，大规模 IT 系统需要更好的设计和管理来处理更频繁的故障。

工欲善其事，必先利其器。基于 Linux 态势感知方法来解决高性能计算的问题，是本书的一个创造性尝试。正如作者所说，在解决大系统问题时，没有通盘战略，你可能会编写一些脚本或者花很长时间盯着日志在屏幕上滚动，你可能会绘制图表来显示数据趋势，你可能会向同事请教他们领域的问题，你可能会参与或领导特别工作组试图解决危急问题或恢复供电，但在最后，就像拼图一样，没有一个统一的方法能解决所有的问题。在诸如大型数据中心、云平台架构和高性能的环境中，大型部署管理是一项非常微妙的任务，它需要大量的经验、努力和对技术的理解来创造一个成功而高效的工作流程。所谓态势感知的问题解决方法，是借鉴自科学领域的一种方法，试图用数学来代替人的直觉，使用统计工程和实验设计以对抗混乱，系统地、循序渐进地工作，努力找到一种统一的方法用于解决同类问题。作者分享

了多年来为基于英特尔的系统开发解决方案的经验，即如何在涉及数千台服务器、数十万个核、分布式数据中心和 PB 级共享数据的环境下排除故障，他的经验涵盖全栈，从识别问题、理解问题到再现问题，再到使用系统化的自上而下的方法来解决。自始至终，本书提供了独特的实践中的例子，强调了高性能计算机系统的规模和操作复杂性。

本书共 11 章。前 6 章主要介绍在解决高性能计算问题方面遇到的一些纯技术性问题，后 5 章侧重于扩展技能和理解，介绍一种完整的整体态势感知方法，深入数学模型和最佳实践，以及监控和配置管理，这使我们更有信心应对新的挑战。

第 1 章介绍了一种可以说是在数据中心应该如何解决问题的近乎方法论的观点。

第 2 章讨论解决问题和研究的基本方法，专门介绍在研究中可能会遇到的一些常见的陷阱。

第 3 章围绕着解决问题使用的一些常见应用程序和工具，比如 ps、top、vmstat、iostat 等，建立起报告的症状与软硬件资源的实际行为之间的联系。

第 4 章讲述进程空间和内核，为了使用工具来揭示任务结构和操作系统本身，我们需要对底层构建模块有更深入的了解。

第 5 章专注于深入分析和刻画（profile）应用程序，以及分析内核本身，通过使用不同的工具为问题创建一个更完整的多维画面，并提出相关的解决方案。

第 6 章进一步讨论内核崩溃分析和内核调试器的使用。包含研究过程中最复杂的部分，从 Kdump 实用程序的设置开始，然后是深入的内核崩溃分析，最后是 kdb 工具的简要概述。

第 7 章关于问题的解决形成了一个具有多层工具和实践的完整原则。首先以实用的方式组织数据，然后与供应商和行业进行内部和外部接洽。随着问题的解决，一个复杂的周期（包括测试、验证和分阶段实施）将在几个关键组件上帮助你做出正确的决定。

第 8 章详细阐述了建立稳健有效的监测环境的考虑因素、方法和工具。解决问题的下一个步骤是确保我们能在数据中发现规律，将它们关联到产生数据的源代码，并试图在第一时间阻止问题出现。

第 9 章介绍环境变化和控制环境的后续过程。我们的关键工具是版本控制和配置管理软件。

第 10 章通过几个实际的例子说明如何在合适的时候做出必要的调整，调整环境 with 需求相匹配，并提供必要的操作灵活性，从而避免之后不得不工作在“救火”模式。

第 11 章总结全书，通过解决问题、在研究中使用系统化的方法，尝试避免工作过程中的经典错误和陷阱。

对于每一章内容，作者都力争以最小的篇幅介绍更多的知识，从工具和工具间对比到基本使用、基础实例、高级实例，层层深入，系统全面。作者运用形象的表达，生动地讲解原本枯燥的知识，能够让读者保持持久的学习热情。比如第 6 章的内核 Kdump 调试部分，内核崩溃所产生的转储（dump）文件可以对调试工作产生难以想象的巨大影响，但随着作者的介绍，你可以进行故意崩溃，可以进行回溯，可以通过异常信号快速定位 bug，让读者深刻领略了 Linux 内核的魅力。纸上得来终觉浅，绝知此事要躬行。在复杂的环境中，比如在一个大规模的履行关键使命的数据中心，要想真正掌握解决问题的方法和技巧，还需要不断实践本书方法并认真地研究系统问题。

本书由中国科学院计算技术研究所张文力博士翻译完成。首先，我要感谢机械工业出版社曲熠编辑的邀请，感谢她的信任、支持和理解。其次，感谢实验室主任陈明宇研究员的支持，感谢课题组的宋辉和梁冬协助完成部分章节的初稿，还要感谢宋纪涛的校译与润色。在此书翻译过程中，我带领团队承担着国家和中国科学院重要研究课题和项目，以及华为公司的数据中心合作项目等，由于任务繁重，翻译工作主要在休息时间完成，感谢我的家人、幼子在此过程中给予的配合、理解和大力支持。

关于此书的翻译工作，译者已尽最大的努力完善，如有不尽之处敬请批评指正。总之，这是一次宝贵的体验！

张文力

2017 年 11 月

前言 Preface

我花了大部分的 Linux 职业生涯在数着成千上万的服务器，就像一个音乐家盯着乐谱能看到在和声中隐藏的波形。过了一段时间，我开始了解数据中心的工作模式和行为。它们就像活着、会呼吸一般，有自己的跌宕起伏、周期和不同寻常。它们远不止是简单的叠加，当你把人作为元素添加到方程中，更是变得不可预知。

在诸如大型数据中心、云平台架构和高性能的环境中，大型部署管理是一项非常微妙的任务。它需要大量的经验、努力和对技术的理解来创建一个成功而高效的工作流程。未来的愿景和经营战略也是需要的。但在相当多的时候，其中某一个重要部分是缺失的。

在解决问题时，没有通盘战略。

本书是我的一个创造性尝试。那些年，在我设计解决方案和产品致力于使我掌控的数据中心变得更好、更强且更高效的时候，也暴露了我在解决问题上的根本差距。人们很难完全理解这意味着什么。是的，它涉及工具和非法侵入系统。你可能会编写一些脚本，或者花很长时间盯着日志在屏幕上滚动。你可能会绘制图表来显示数据趋势。你可能会向同事请教他们领域的问题。你可能会参与或领导特别工作组试图解决危急问题和恢复供电。但在最后，就像拼图一样，没有一个统一的方法能解决所有的问题。

态势感知的问题解决方法借鉴自科学领域，它试图用数学来代替人的直觉。我们将使用统计工程和实验设计以对抗混乱。我们会慢慢地、系统地、一步一步地工作，努力找到一种统一的方法用于解决同类问题。我们关注打破数据神话，摆脱一些波及数据中心的偏见和传统。然后，我们将把系统故障排除的艺术转化为产品。这可能听起来很残酷，艺术将按重量出售，但当你深入阅读本书的时候，其中的必要性就变得显而易见。你的那些不自在，在原本无论接触监控、变化控制和管理、自动化以及其他哪怕是最好的实践都会有的不自在，统

统将会慢慢融入现代数据中心。

最后同样重要的是，请不要忘记我们尽一切努力去研究和解决问题的初衷——乐趣和好奇心，这正是让我们成为工程师和科学家的真正原因，正是让我们热爱数据中心技术编织的混乱、忙碌而又疯狂的世界的原因。

快来和我们一路同行吧。

Igor Ljubuncic

致 谢 *Acknowledgements*

写这本书的时候，我会偶尔离开办公桌，四处与人交谈。他们的意见和建议帮助塑造了这本书，使它拥有更漂亮的形式。因此，我要感谢 Patrick Hauke 确保这个项目最终完成，感谢 David Clark 编辑书稿并调整句子段落，感谢 Avikam Rozenfeld 提供有用的技术反馈和理念，感谢 Tom Litterer 在正确的方向上做正确的引导，最后同样重要的是，感谢其他那些在英特尔的聪明而又努力工作的人。

女士们，先生们，请允许我脱帽致敬。

Igor Ljubuncic

Contents 目 录

译者序	参考文献	5
前 言		
致 谢		
引言 数据中心与高端计算	第1章 你有问题吗	6
数据中心一览	问题的识别	6
现代数据中心布局	如果森林里有一棵树倒下, 没人	
欢迎来到 BORG, 抵抗是徒劳的	能听到	6
那就是力量	一步一步地识别	7
企业与 Linux	总是优先使用简单工具	7
经典的办公室	过多的知识导致误区	9
Linux 计算环境	问题的定义	9
Linux 云	当前发生或可能的问题	10
10000 个 1 不等于 10000	停机规模和严重性与业务需求	11
问题的非线性扩展	已知与未知	13
大数定律	问题的再现	13
同质化	你能隔离问题吗	13
企业的当务之急	偶发性问题需要特殊处理	14
7 天 24 小时全天候开放	计划如何控制混乱	14
任务危急	放手是最难的事	15
停机时间等于金钱	因与果	16
千里之堤溃于蚁穴	不要流连于症状	16
	先有鸡还是先有蛋	16

严格控制环境改变，直到你理解了问题的本质	18	喊狼来了的监控	29
如果改变，确保你知道预期结果是什么	18	读取系统信息和日志	30
结论	19	使用 ps 和 top	30
参考文献	19	系统日志	36
第2章 开始研究	20	进程记账	39
隔离问题	20	检查命令执行的模式	40
推动从生产到测试	20	与问题表现相关	41
重新运行获得结果所需的最小集	21	避免快速的结论	41
忽略偏见信息，避免假设	21	统计作为辅助	41
与健康系统和已知参考的比较	22	vmstat	41
那不是程序错误，而是一个特性	22	iostat	43
将预期结果与正常系统作比较	23	系统活动报告 (SAR)	47
性能和行为的参考是必需的	24	结论	49
对变化的线性与非线性响应	24	参考文献	50
一次一个变量	25	第4章 深入探讨系统	51
线性复杂度的问题	25	使用 /proc	51
非线性问题	25	层次	52
响应可能会延迟或掩盖	25	每个进程的变量	52
Y 到 X 而不是 X 到 Y	26	内核数据	56
组件搜索	26	进程空间	60
结论	26	检查内核可调参数	63
第3章 基础研究	27	sys 子系统	63
刻画系统状态	27	内存管理	64
环境监控	27	文件系统管理	65
机器可访问性、响应性和正常运行时间	27	网络管理	65
本地和远程登录以及管理控制台	29	SunRPC	66
		内核	67
		sysctl	69
		结论	70
		参考文献	70

第5章 变身极客——跟踪和

调试应用 72

使用 strace 和 ltrace 72

strace 72

ltrace 86

结合两种工具获得最佳结果 87

其他工具 90

使用 perf 93

介绍 93

为什么不用 OProfile 94

前提 94

基本使用 94

基础实例 99

高级实例 103

小结 107

使用 gdb 107

介绍 107

前提 108

简单实例 108

不那么简单的实例 113

其他有用的命令 122

结论 124

参考文献 124

第6章 极客进阶——应用和内核核心、

内核调试器 126

收集应用核心 126

如何转储应用核心 127

收集内核核心 (Kdump) 132

Kdump 服务概览 133

Kdump 配置 137

测试配置 143

Kdump 网络转储功能 145

Kdump 使用 146

小结 147

崩溃分析 (crash) 147

前提 147

运行 crash 148

内核崩溃核心分析 156

超级极客技能: C 代码分析 167

小例子 168

中级例子 172

困难的例子 176

内核崩溃的 bug 报告 182

崩溃分析结果 184

内核 bug 与硬件错误 185

小结 186

内核调试器 186

内核编译 186

进入调试器 187

基本命令 187

小结 189

结论 189

参考文献 189

第7章 问题的解决方案 191

如何处理收集到的数据 191

文档 191

数据的杂乱 193

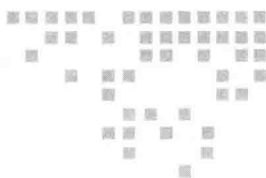
最佳实践 195

搜索引擎、邮件列表以及供应商

支持 202

找到根本原因	203	为什么需要版本控制	222
消除问题	204	Subversion、Git 及相关软件	223
实现和跟踪	206	简单回滚	223
结论	207	配置管理	223
参考文献	208	变懒：自动化	224
第8章 监控和预防	209	大环境下的熵	225
监控什么数据	209	掌控混乱	225
过多的数据比没有数据更糟	209	配置管理软件	225
Y 到 X 将定义你所需要的监控	210	环境中引入变化的正确方法	226
不要害怕改变	210	一次一个变化	226
如何监控和分析趋势	211	不要急着冲到截止时间	227
设置你关心的监控	211	理解影响	228
监控不等于报告	212	没报告问题意味着什么都没有	228
不监控随机指标	212	连锁反应	228
定义数学趋势	213	结论	229
如何应对趋势	215	参考文献	229
当它来得太晚	216	第10章 微调系统性能	230
内务管理	217	日志大小与日志轮转	230
预防是解决问题的关键	217	系统日志会一直增长	230
配置审核	218	慢点，倒带	232
为什么审核有用	218	确定消息的内容	232
控制环境的变化	219	文件系统调优	234
安全方面	219	Ext3/4 文件系统	234
系统数据收集工具	219	XFS 文件系统	235
定制工具	220	sysfs 文件系统	236
商业支持	220	层次	236
结论	221	block 子系统	237
参考文献	221	FS 子系统	237
第9章 让你的环境更安全、更健壮	222	内核子系统	237
版本控制	222	模块子系统	238

结合 proc 和 sys	238	使用的工具	248
内存管理实例	238	使用的工具概述	248
CPU 调度实例	238	所选工具的优缺点	249
网络优化实例	241	从简单到复杂	249
结论	243	不要过多涉猎：知识是你的	
参考文献	243	敌人	249
第11章 整合所有的部分	245	逐步进阶的方法	249
自上而下的方法	245	不要害怕退步	250
保持简单：从简单开始	245	有时你只需要缓解问题	250
首先理解环境	245	操作约束	250
它就像洋葱：分层而且会使你		钱，钱，钱	250
流泪	246	你的客户永远无法忍受停机	250
“自我消失”的问题会重现，且		你将不得不妥协	251
扩大影响	246	聪明的做法	251
没有捷径：努力是金科玉律	246	分享就是关怀	251
使用的方法学	247	请教别人：他们以前碰到过	
文档	247	类似问题	252
一种清晰的方法	247	工作安全根本没有安全可言	252
Y 到 X 应该是座右铭	247	授人以鱼或者授人以渔	253
统计工程不受重视	248	只有你知道什么对你而言是	
数学是强大的，但没人用	248	最好的	253
		结论	253



引言

数据中心与高端计算

数据中心一览

如果你正想给数据中心定个调，找一种通俗的说法，那么不妨称它们为现代化电厂。它们是老旧的、乌黑的煤炭工厂的等价物，这些工厂曾经帮助 19 世纪中期具有开创精神的年轻实业家获得比当地村庄的商人更丰厚的收益。工厂和工人是那个时代的无名英雄，他们在幕后默默无闻地辛勤劳作，却是 19 世纪风靡世界的工业革命中真正的中坚力量。

快进 150 年，一场类似的革命正在发生。伴随着所有相关的困难、嗡嗡声和真正的技术挑战，世界正从模拟的状态朝着数字化方向转换。其中，正是数据中心承载着互联网的动力室、搜索的心脏地带、成就大数据之所谓“大”的这些重要角色。

现代数据中心布局

实际上，如果要探讨数据中心的设计细节和所有基础部件，我们将需要半打书把这一切写下来。此外，由于这只是一个引子，一道开胃菜，因此我们将只简单地接触这一领域。在本质上，它归结为三个主要部分：网络、计算和存储。无限长的线缆、数千的硬盘、怒然全速运行的中央处理器（CPU），它们共同服务着每秒上亿万的请求。但单靠这三大支柱尚不足以支撑一个数据中心，还有更多的组成部分。

如果你想做一个类比，可以想象一下航空母舰。第一个闪过我脑海的是，汤姆·克鲁斯驾着他的 F-14 起飞，背景播放着肯尼·洛金斯的《危险区》。而太容易忽略的事实是，起飞操作的背后有成千上万的航空乘务员、技工、技师、电工和其他专家。我们几乎太容易忘记，是一层楼一层楼的基础设施和车间，以及在它最核心的 IT 中心，共同精心策划了这全部的场景。

数据中心有点类似于巡逻海洋的 10 万吨级的庞然大物。它们有自己的组成部分，但每部分都需要沟通和共同协作。这就是为什么当你谈论数据中心概念的时候，冷却和功率密度之类的概念与可能使用的处理器和磁盘类型同样关键。远程管理、设施安全、灾难恢复、备份——所有这些都几乎不在列表上，但当系统规模扩展得越大时，这些就显得越发地重要。

欢迎来到 BORG，抵抗是徒劳的

在过去的几年中，我们看到一个趋势，包括计算部件在内的陈旧配置正在走向标准化。像其他技术一样，数据中心已经达到了一个点，它再也不能独善其身，这个世界不能容忍一百个不同版本的数据中心。类似于其他技术的融合，例如网络协议、浏览器的标准以及从某种程度而言的媒体标准，作为一个整体的数据中心也正在走向标准化。例如，开放数据中心联盟（Open Data Center Alliance, ODCA）是一个成立于 2010 年的协会，致力于推动可以共同使用的解决方案和服务——标准——在整个行业的采纳。

在这样的现实下，流连于自定义的车间就有如逆水游泳。迟早，不是你就是河流准得放弃。但仅仅拥有一个数据中心是远远不够的。这也是本书存在的一部分原因——在庞大的、独一无二的高性能配置中，针对数据中心解决问题并创造解决方案，这是未来不可避免的。

那就是力量

在深入任何战术问题之前，我们需要讨论策略。在一台家用电脑上和在数据中心做同类工作远没有可比性。尽管技术是几乎相同的，但你以前总结的所有注意事项——以及你的本能——都是完全错误的。

高性能计算的开始和结束是按比例扩展的，能力以可持续的方式稳步增长，却不会造成成本以指数方式成倍增加。这一直是一个具有挑战性的任务，很多时候，企业的业务一旦无限制膨胀，就不得不牺牲增长。迫使放缓速度的因素往往是不起眼的东西——电源、物理空间，经常不是那些直接的或可见的考虑因素。

企业与 Linux

我们所面临的另一个挑战是转变，从经典业务的传统模式到来势汹汹、快节奏且不断变化的云的转变。再一次，这不是一个技术问题。它是关于那些已经在 IT 行业摸爬滚打多年的人，他们正经历着眼前这突如其来的变化。

经典的办公室

一直以来，让上班族使用软件、与同事和合作伙伴进行沟通、发送电子邮件和聊天已