

《发布！软件的设计与部署》

图书基本信息

书名：《发布！软件的设计与部署》

13位ISBN编号：9787115380457

出版时间：2015-2-1

作者：[美] Michael T.Nygard

页数：228

译者：涂鸣

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《发布！软件的设计与部署》

内容概要

本书分为四个部分，每部分内容都由一个研究案例引出。第一部分介绍了如何保证系统的生存，即维护系统正常运行。第二部分介绍了如何衡量系统的容量，以及如何随时间来优化系统的容量。在第三部分讲述了架构师在为数据中心构建软件时应该思考的一般设计问题。第四部分讨论了系统的运行寿命，将其作为整个信息生态环境的一部分。书中代码采用的Java语言，熟悉编程的人都能轻易阅读。作者从Java和Unix的视角看问题，因而本书的焦点比较中立，着眼于跨所有平台的技术和概念。在每种模式和反模式的末尾，都有一段简短精要的总结。

本书适合面向企业级软件的架构师、设计师和开发人员阅读参考。

《发布！软件的设计与部署》

作者简介

Michael T. Nygard

从业二十余年的资深程序员、架构师，被誉为在线业务的“流动解决问题专家”。先后为美国政府、军队、银行、金融、农业和零售等多个行业交付过运营系统，这种实际运营的经历改变了他对软件架构和开发的看法，也让他对在相当不友好的环境下构建高性能、高可靠性的软件有了独特的见解。他写过多篇文章和社论，是软件架构经典著作《架构之美》作者之一。

书籍目录

第1章 引言	1
1.1 瞄准正确的目标	1
1.2 使用决断力	2
1.3 生活的质量	3
1.4 挑战的范围	3
1.5 随手一松就是一百万	3
1.6 务实的架构	4
第一部分 稳定性	
第2章 案例研究：航空系统宕机的异常	8
2.1 事故	9
2.2 结果	12
2.3 事后调查	12
2.4 确凿的证据	15
2.5 一点预防	17
第3章 稳定性概述	19
3.1 定义稳定性	20
3.2 故障模式	22
3.3 裂痕扩散	22
3.4 故障链	23
3.5 模式与反模式	24
第4章 稳定性反模式	26
4.1 集成点	27
4.2 连锁反应	37
4.3 连锁故障	40
4.4 用户	42
4.5 阻塞的线程	50
4.6 自我否定攻击	55
4.7 尺度效应	57
4.8 不平衡的容量	60
4.9 慢响应	63
4.10 SLA倒置	64
4.11 无边界结果集	67
第5章 稳定性模式	70
5.1 使用超时	70
5.2 断路器	73
5.3 隔板	75
5.4 稳定状态	78
5.5 快速失效	83
5.6 握手	85
5.7 测试装置	87
5.8 去耦合中间件	90
第6章 稳定性总结	93
第二部分 容量	
第7章 案例研究：被客户压迫	96
7.1 发布倒计时	96
7.2 瞄准QA	97
7.3 负载测试	99

7.4	被大量会话所杀	101
7.5	测试的鸿沟	102
7.6	后果	103
第8章	容量概述	105
8.1	定义容量	105
8.2	约束	106
8.3	关联	107
8.4	可扩展性	107
8.5	容量的神话	108
8.6	总结	114
第9章	容量反模式	115
9.1	资源池竞争	115
9.2	泛滥的JSP碎片	118
9.3	AJAX过度之伤	119
9.4	驻留过久的会话	121
9.5	HTML中浪费的空间	122
9.6	刷新按钮	125
9.7	手工的SQL语句	126
9.8	数据库富营养化	128
9.9	集成点延迟	130
9.10	Cookie怪兽	131
9.11	总结	133
第10章	容量模式	134
10.1	连接池	134
10.2	谨慎使用缓存	136
10.3	预计算容量	137
10.4	调整垃圾回收器	140
10.5	总结	142
第三部分	一般设计问题	
第11章	网络连接	144
11.1	多宿主服务器	144
11.2	路由	146
11.3	虚拟IP地址	146
第12章	安全	149
12.1	最少特权原则	149
12.2	配置的密码	150
第13章	可用性	151
13.1	收集可用性需求	151
13.2	记录可用性需求	152
13.3	负载均衡	153
13.4	集群	157
第14章	管理	159
14.1	“测试和产品匹配吗？”	159
14.2	配置文件	161
14.3	启动和关闭	163
14.4	管理接口	164
第15章	设计总结	165
第四部分	运营	
第16章	案例研究：惊人的宇宙	168

《发布！软件的设计与部署》

16.1	旺季	168	
16.2	婴儿的第一个圣诞	169	
16.3	切脉	169	
16.4	感恩节	170	
16.5	黑色星期五	170	
16.6	重要的信号	172	
16.7	诊断测试	172	
16.8	专家打来电话	173	
16.9	比较解救方案	174	
16.10	条件是否会响应处理	175	
16.11	收尾	176	
第17章 透明度		177	
17.1	视角	178	
17.2	透明度设计	184	
17.3	使用各种技术	184	
17.4	日志	185	
17.5	监控系统	190	
17.6	法律上及事实上的标准	194	
17.7	操作数据库	201	
17.8	支持流程	205	
17.9	总结	208	
第18章 适应		209	
18.1	与时俱进	209	
18.2	适应性的软件设计	210	
18.3	适应性的企业架构	215	
18.4	发布应无害	220	
18.5	总结	224	
参考书目		226	

《发布！软件的设计与部署》

精彩短评

- 1、很简单
- 2、翻译太烂。
- 3、非常非常棒，进阶必备
- 4、本来应该是入门程序员必读书，介绍了线上系统可用性的各个方面，但是烂到极致的翻译把这本书变成了鸡肋。译者显然既不精通翻译，也不精通it，否则也不会把出现“如果一点点好，那么很多一定会更好”，或者“资源隔板”之类匪夷所思的词句了。
- 5、翻译不行
- 6、像在看故事书。不要被标题骗了，和发布与部署关系不大。
- 7、帮助你理解这个世界的复杂性
- 8、这本书介绍了在设计软件过程中的一些不正确的反模式，然后解释什么才是比较不错的模式，包括集成点、架构分层、连接池、缓存、数据库等各个方面。作者结合自己的经历介绍了一些案例，看了之后还是有所感触，以前自己也遇到过作者提到的问题。在软件设计结束后发布时，作者又介绍了发布时常常遇到的一些诸如网络、集群、负载均衡、配置等问题，以及运营有关的问题。总的来说这本书还是写的比较浅显易懂的，讲的也全面，挺不错。
- 9、终于看到中文版了
- 10、从稳定性和容量等比较有价值的角度，提出了建议和方法。值得一读，中文翻译得勉强
- 11、讲了线上故障，服务化，容量规划等提升线上系统可用性的模式和反模式，只是翻译的太烂了。
。。
- 12、可能是书太老了，读的时候感觉看到目录就能大概知道具体可能要讲什么，所以个人感觉难符它的盛名，不过对关注点的考虑倒是真的比较全面，要自己弄出这样一个完整的目录还是需要很多经验积累的。
- 13、译的晚了，而且译得一般。很多思想还是很值得借鉴的。架构师入门。
- 14、系统发布投产开始，才是真正诞生的一天，而不是开发和设计开始。不断适应变化，但是变化本身都是有成本的，开发测试发布都是如此，所以敏捷其实反而是最强调质量的，只有自动化测试，持续集成和自动化部署，重构，全部整套的团队，技术，方法，工具充分，才有可能敏捷

《发布！软件的设计与部署》

章节试读

1、《发布！软件的设计与部署》的笔记-第97页

每个网站项目都是企业集成项目。

2、《发布！软件的设计与部署》的笔记-第20页

高度稳定的设计通常和不稳定的设计的实现成本相同。

3、《发布！软件的设计与部署》的笔记-第98页

Conway法则

任何一个产品设计都会不可避免地反映出设计它的团队组织的沟通结构。

4、《发布！软件的设计与部署》的笔记-第123页

空格每年会耗费15000美元。

5、《发布！软件的设计与部署》的笔记-第71页

使用有带timeout参数的函数。

6、《发布！软件的设计与部署》的笔记-第36页

利用断路器和解耦中间件防范集成点故障。

7、《发布！软件的设计与部署》的笔记-第136页

在Java中，应该用SoftReference对象建立缓存来保持缓存单元。如果内存不足，垃圾回收器将保证回收任何只能通过软引用才能访问的对象。结果就是，使用软引用的缓存将帮助而不是阻碍垃圾回收器回收内存。

8、《发布！软件的设计与部署》的笔记-第66页

调用第三方，服务水平只会下降

9、《发布！软件的设计与部署》的笔记-第40页

当某层的故障引发了调用层的故障，就会出现连锁故障。

10、《发布！软件的设计与部署》的笔记-第52页

不要信任对领域对象的同步方法。

11、《发布！软件的设计与部署》的笔记-尺度效应

《发布！软件的设计与部署》

生物学上，平方立方定律告诉我们为什么没见过大象一般大的蜘蛛。昆虫的重量和它的体积成比例，因此是 $O(N^3)$ 。腿的长度和表面积成比例，因此是 $O(N^2)$ 。如果你让小动物十倍大，这会让强度重量比变为 $1/10$ ，那么腿就不能支撑它了。

12、《发布！软件的设计与部署》的笔记-第32页

PLA服务水平协议：提供一个可衡量、量化的服务的合同义务，违反PLA会收到经济处罚。

13、《发布！软件的设计与部署》的笔记-第136页

应用程序这一级所用的缓存的最大内存应该是可配置的。如果不限限制占用的最大内存，缓存将最终吃掉系统所有的可能内存。这时，为了处理请求，垃圾回收器将花越来越多的时间来试图恢复足够的内存。缓存消耗了其他任务所需的内存，将最终导致系统运行极其缓慢。

14、《发布！软件的设计与部署》的笔记-第93页

故障不可避免，避免反模式，也许不会阻止坏情况的发生，但是它会帮你减少损失。

15、《发布！软件的设计与部署》的笔记-第13页

重大事故之后的管理认知和管理事故成本本身同样重要。

16、《发布！软件的设计与部署》的笔记-第70页

网络永远是不可靠的。

17、《发布！软件的设计与部署》的笔记-第54页

阻塞线程是大多数故障的原因。

18、《发布！软件的设计与部署》的笔记-第4页

不要为了避免一次开发成本，而以经常性运营成本为代价

19、《发布！软件的设计与部署》的笔记-第4页

系统设计与架构上的决策也是财务决策，制定这些决策都必须着眼于实现成本和宕机成本。这种技术和财务相融合的观点，是本书反复探究的一个最重要的主题。不要为了避免一次性开发成本，而以经常性运营成本为代价。

20、《发布！软件的设计与部署》的笔记-第2页

我们需要设计出一个个独立的软件系统，以及相互依赖的多个系统所构成的整个生态系统，以此来低成本和高质量的IT运营。

21、《发布！软件的设计与部署》的笔记-第84页

《发布！软件的设计与部署》

在事务开始时检查资源的可用性。

22、《发布！软件的设计与部署》的笔记-观察的关键

首先和最重要的是，观察者应该观察趋势又观察异常值。 - 每周，审查上周的问题单据。

- 每月，查看所有问题。考虑问题类型的分布。随着严重问题被改正，总体趋势应该是严重程度递减的。

...

- 如果有太多的单据和服务台呼叫需要彻底审视，就寻找顶层分类，并随机抽取单据来找出让你头疼的事情。

- 至少每个月，检查数据容量和查询统计。

...记住，兴趣的焦点是会随着时间转移的。在早期，问题主要是反映式的。单据检察、事故事后报告以及最近趋势将是最有价值的。随着根本原因被改正，新代码发布，流量模式改变，重点将会从反应式过渡到预测式分析。... 一旦某个度量信息停止产生有用的信息，就停止检查它。

23、《发布！软件的设计与部署》的笔记-第81页

不要把日志留在生产环境，要把他们复制到一个中间地带进行分析。

24、《发布！软件的设计与部署》的笔记-第21页

进行寿命测试，这是找到长期存在的错误的唯一方式。

25、《发布！软件的设计与部署》的笔记-第79页

数据清除功能应该加入到第一版中。

26、《发布！软件的设计与部署》的笔记-第111页

存储是驱动器、互连、分配、冗余和用来备份高端服务的整个管理系统。

存储是项服务，而不是设备。

《发布！软件的设计与部署》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：www.tushu000.com