Broadview®
www.broadview.com.cn

大话 Java性能优化

周明耀／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

大话 Java性能优化

周明耀／著

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

内 容 简 介

本书主要提供 Java 性能调优方面的参考建议及经验交流。作者力求做到知识的综合传播，而不是仅仅只针对 Java 虚拟机调优进行讲解，另外力求每一章节都有实际的案例支撑。具体包括：性能优化策略、程序编写及硬件服务器的基础知识、Java API 优化建议、算法类程序的优化建议、并行计算优化建议、Java 程序性能监控及检测、JVM 原理知识、其他相关优化知识等。

通读本书后，读者可以深入了解 Java 性能调优的许多主题及相关的综合性知识。读者也可以把本书作为参考，对于感兴趣的主题，直接跳到相应章节寻找答案。

总的来说，性能调优在很大程度上是一门艺术，解决的 Java 性能问题越多，技艺才会越精湛。我们不仅要关心 JVM 的持续演进，也要积极地去了解底层的硬件平台和操作系统的进步。

图书在版编目（CIP）数据

大话 Java 性能优化 / 周明耀著. —北京：电子工业出版社，2016.4
ISBN 978-7-121-28481-6

Ⅰ. ①大… Ⅱ. ①周… Ⅲ. ①JAVA 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2016）第 063438 号

责任编辑：董　英
印　　刷：北京京科印刷有限公司
装　　订：三河市皇庄路通装订厂
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：787×1092　1/16　　印张：35.25　　字数：993 千字
版　　次：2016 年 4 月第 1 版
印　　次：2016 年 4 月第 1 次印刷
印　　数：3000 册　　定价：89.00 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到 dbqq@phei.com.cn。

服务热线：（010）88258888。

序

最大的思想紊乱是相信人们想要相信的事情。

——路易斯·巴斯德（Louis Pasteuer）

Michael 周是个具有丰富程序经历的架构师和项目管理者，他从国内作坊式的软件开发公司起步，经历了著名的咨询公司凯捷的欧洲工作洗礼，后来于美国花旗软件担任高级软件技术总监，平时常常思考和总结 21 世纪以来我国软件开发者，特别是 Java 开发工程师的困惑。

我们通常情况下，一开始可以有条不紊地进行软件需求定义和分析，随着上线时间的不断追近，面对客户的咄咄逼人的需求修改和即刻变更需求上线压力，程序员作为弱势群体，往往会考虑时间优先原则，很难守住按部就班的开发计划和开发方式，从而导致出现了软件质量的大幅度下降。**软件一定存在修改的余地，**但是程序员们通常不相信自己的系统存在诸多问题，尤其是感觉自己已经做得相当完美。系统调优在软件的后续改进和重构中占有很大的地位，能够弥补前述的不足，本书以通俗的语言和引人入胜的故事，重点讲述软件性能调优的方法论和具体实现路径，读者可以根据自己的实际情况进行参照比对，就像进了兵器库挑选合适自己的顺手武器。

程序凑合着上线是一回事，而能够优美地运行在压力下往往很不容易。本书对于所有有志于进行软件高级管理的人员而言，具有非常重要的意义。

海适云承 CEO 兼首席架构师　沈英桓（Sam Shen）

前　言

7 岁那年，当我合上《上下五千年》一套三册书籍时，我对自己说，我想当个作家。这一晃 27 年了，等待了 27 年，我的第一本书《大话 Java 性能优化》即将面世了。我是多么的忐忑、惊喜，就像第一次面对我的女儿“小顽子”，给她取这个小名，希望她顽强到底，因为我相信，你若顽强到底，一切皆有可能。

从 15 岁拥有自己第一台电脑算起，已经有接近 20 年的计算机学习时间，加上 11 年的工作经历，我对于工作，对于工程师这个职业，有一些自己的感悟。我认为，职业素养非常重要。

1929 年，在汪精卫的支持下，余云岫等人提出了全面废除中医、禁止中医的提案，并很快获得初审通过。在这样的局面下，全国各地中医师多次到南京请愿，虽有孙科等人的支持，但反响不大。相持阶段，无独有偶，汪精卫的岳母身患痢疾，西医师医治无效，京城四大名医之一的施今墨先生毅然赴汪府。施今墨凭脉，每言必中，使汪精卫的岳母心服口服，频频点头称是。处方时施今墨说：“安心服药，一诊可愈，不必复诊。”病危至此，一诊可愈？众人皆疑。据此处方仅服数剂，果如施今墨所言。汪精卫不得不服中医，最终撤回提案。施老先生医德高尚，死后遗体都捐献出来供科学研究，绝不是阿谀奉承之人，他赴汪府，完全是因为对中医生这个职业的尊重，为了让人知道中医的深奥。

戒口

佛教五戒之一的不妄语，要求我们不欺骗他人、不在不清楚实际情况的时候胡乱说话，放到职场，也可以加上信息安全的要求。

《越绝书》载文种述九术时说：“故曰九者勿患，戒口勿传，以取天下不难，况于吴乎？”文种希望勾践秘而不宣，以免人多口杂，泄露机密。每个人都有自己的岗位、职责，我们要做的是做好自己的事情，不对不属于自己工作范围内的事情评价、传播，不在背后说同事的坏话。作为一名技术人员，如果不能做到戒口、静心、专心，那我觉得你应该尽早转行，你不适合，也绝不会成为一名技术大拿。

气场

一位职业的工作者，他身上有一种称为气场的东西存在。人的气场是看不见的，但这种力量是巨大的，就像万有引力一样，我们每个人身上的这种气场无时无刻不在影响你的人生。这种气场的行程与你的观念、信仰、环境、朋友、呼吸、事物、欲望、静息与睡眠相关。一个人的气质

很好，外表精神、有修养、有道德，这个人的气场就好，就会吸引好的事，吸引好的运气。每个人都会遇到各种各样的苦难，但是我坚信，你若顽强到底，一切皆有可能。

教养

看不见的教养很难。在乌合之众中谁能保持优雅和教养？在群体无意识中谁能保持清醒和判断？更难的是那些“慎独”的教养。日本有一种文化，叫作“不给别人添麻烦”的文化，我们每个人在做事之前都应该考虑是否自己的行为会给别人造成麻烦。教养不是道德规范，也不是小学生行为准则，其实也并不跟文化程度、社会发展、经济水平挂钩，它更是一种体谅，体谅别人的不容易，体谅别人的处境和习惯。对于教养，我个人的理解是，谦逊是一种教养，自尊更是。

心态

尼克·胡哲说过，人们经常埋怨什么也做不来，但如果我们只记挂着想拥有或欠缺的东西，而不去珍惜所拥有的，那根本改变不了问题！真正改变命运的，并不是我们的机遇，而是我们的态度。

一个人的心态很是重要，心量小的人，芝麻大小的事情也能在心里翻江倒海。心量大的人，即使在危机面前也能镇静自若。同样一件事情，掀起的波澜大小却因人而异。有一句话很好，用于技术人员我觉得尤其合适，“想要成为一棵大树，就不要去和草争”。

一个人的成就，不得以金钱衡量，而是一生中，你善待过多少人，有多少人怀念你。成功并非单指事业，无论是爱好或职业上的成功都只是成就。成功应该是多元化的，如人的一生包含了很多追求一样，而非单一指向。然后，无论你多有成就，真正的成功，就是陪伴家人。所有的情感都是需要陪伴的，这些陪伴成为一个个美好的回忆，这些都是整个家庭最宝贵、最重要的财富，这些远远超越物质的重要性。在中国，因为价值观相对比较单一，社会显得很浮躁、很物质，所以大多以物质的追求为主，越多越好，内心也想过美好的生活。但当你的心完全趋向金钱的时候，很多美好的东西就会自动屏蔽了，不会出现在生活中。别让忙碌空白了回忆。

此外，作为一名技术人员，我觉得，职业生涯中可能很多次需要面对工作的变换、角色的变化，有很多知识需要学习，所以，我们应该把“归零”当成一种生活的新常态。

劝学

我觉得有一句话总结得特别好，“能干工作、干好工作是职场生存的基本保障”。

荀子是儒家八派中一派的创始人，其思想学说以儒家为本，兼采道、法、名、墨诸家之长。荀子在他的著作《劝学》一文中这样写道，“君子曰：学不可以已。青，取之于蓝，而青于蓝；冰，水为之，而寒于水。”这段文字大体表达了学习是不可以停止的，君子广泛学习并且每天反省自己，就会明白道理，行为上也不会有什么过错。

全球成功的科技型企业，无论是微软的比尔·盖茨，还是苹果的乔布斯，Facebook 的扎克伯

格，无一不是技术专家，创新型企业必须由这样的企业家带队，懂技术，就会站在前沿。对于大型科技企业而言，光懂技术不够，还要懂市场。

诸葛亮在给他的儿子写的著名的《诫子书》中指出，宁静才能够修养身心，静思反省。不能够静下来，则不可以有效地计划未来，而且学习的首要条件，就是有宁静的环境。审慎理财，量入为出，不但可以摆脱负债的困扰，更可以过着简朴的生活，不会成为物质的奴隶。要计划人生，不要事事讲求名利，才能够了解自己的志向，要静下来，才能够细心计划将来。学习需要专注，平静心境才能事半功倍。学习的过程中，决心和毅力非常重要，因为缺乏了意志力，就会半途而废。拖延就不能够快速地掌握要点。时光飞逝，意志力也会随着时间消磨。

归属感

每个足球队有 11 位球员在球场上比赛，估计最不引人注目的应该是守门员了吧，他要忍受着大多数时间的无聊，还要保持着警惕。当危机发生时，很有可能还要一个人战斗，需要勇敢地面对对方前锋，唯一的目标是，绝对不让你攻破球门。我们很多时候可能也是如此，苦苦奋斗，当解决了某个问题，或是帮助公司拿到某个招标，我们都会感到自豪感、成就感，这就是归属感，对于技术领域的归属感。

最后，自我介绍一下，我叫周明耀，研究生学历，一名九三学社社员，12 年工作经验，IBM 开发者论坛专家作者。我是一名 IT 技术狂热爱好者，一名顽强到底的工程师。我推崇技术创新、思维创新，对于新技术非常热爱。

感谢我的家人，和谐的家庭帮助我完成了这本书，我的妻子，她美丽、细心、博学、偶尔不那么温柔，但是我很爱她。我的小顽子，她天生的性格很像我，希望她能够踏踏实实做人，保持创新精神，平平安安、健健康康地生活下去。感谢我妻子父母、我的父母，他们帮我照顾小孩，我才有时间编写此书。感谢浙江省特级教师、杭州高级化学老师郑克良老师，郑老师的一句“永远不要放弃”，推动着我多年的发展。感谢数学老师张老师在公开场合对我智商的褒奖，第一次收获这样的赞赏，对我这样性格的孩子是多么的重要，谢谢。感谢王芳同学，因为你的插画天赋，让这本书的内容更加丰富、可读，不要忽视了自己的才华，你很有天赋。

我相信这本书不是终点，它是麦克叔叔此生一系列技术书籍的开端，下一本书籍见。

目　录

第 1 章　性能调优策略概述

2011 年 1 月，新加坡飞往杭州的航班。飞行持续时间很长，大约 6 个小时，坐在四周的人很快熟悉了，互相攀谈起来。有一位小姑娘，十六七岁的模样，长得很漂亮，默默地坐在座位上。热心的阿姨和她攀谈，问起她的情况，她带着疲倦自我介绍起来，“我在新加坡念初三，那所学校一点都不好，我在成都是最好的初中毕业的，也考上了成都最好的高中，但是，我的父母，他们一定要我来新加坡复读初三，让我考新加坡的高中，我一点都不喜欢这里，这里的同学看不起我们这些大陆学生，经常上课找大陆来的老师麻烦，经常辱骂我们，我烦透了!!!”对，这不是自我介绍，这是一个人接近奔溃边缘的歇斯底里。也就是在当时，我做出了决定，我绝不会让我的女儿这样远离我，一个人在很年幼的时候就必须独立面对生活的困难，绝不。无论她的父母出于什么原因让她去国外念书，我所看到的，是让一个不适合承受压力的人承担了巨大的压力，这就是本书的编写原因。在这本书里我想要和大家讨论的话题是基于 Java 语言的性能优化，我们不能随意地给出性能优化方案，就像随意指派由那位小姑娘来完成全家的未来方向一样。我们必须经过严密的研究、测试及验证，明确造成性能瓶颈真正的原因后才能开始着手，盲目地行动只会造成不必要的损失。当然，如果系统架构设计得很好，就可以在很大程度上避免类似事情发生，这不是本书的主要讨论范围。

本章主要介绍和解决以下问题，这些也是全书的基础：

- 为什么需要调优，这是您阅读本书的依据，只为需要调优而调优。
- 了解程序性能的各项指标，包括物理机器性能、程序性能。
- 性能调优分类方法，包括调优方向、调优方法、调优层次。

1.1　为什么需要调优

注意，这一节会提到许多技术名词，本着让 Java 初学者看懂本书的目的，笔者尽量第一时间做出注释，如有遗漏读者可以阅读后续章节，均有详细介绍。

经历了多年的发展，Java 已由一门单纯的计算机编程语言，逐渐演变为一套强大的技术体系平台。根据不同的技术规范，Java 设计者们将 Java 划分为 3 种结构独立但却又彼此依赖的技术体系分支，分别是 Java SE、Java EE 和 Java ME[1]，其中 Java EE 被广泛使用在企业级领域，除了包括 Java API 组件外，还扩充有 Web 组件、事务组件、分布式组件、EJB 组件、消息组件等，并持续发展到现在。综合 Java EE 的这些技术，开发人员可以构建出一个具备高性能、结构严谨的企业级应用，并且 Java EE 也是用于构建 SOA 架构的首选平台。

Java 的持续发展要感谢 Google，正是 Google 将 Java 作为 Android 操作系统的应用层编程语言，使得 Java 可以在 PC 时代、移动互联网时代都得到迅猛发展，可以用于手持移动设备、嵌入式设备、个人电脑、高性能的集群服务器或大型机。

随着互联网业务的不断拓展、繁荣，越来越多的系统架构开始参照互联网企业的系统架构方式。无论是互联网、物联网，还是传统行业的软件设计，笔者认为，任何技术都离不开对业务需求的支撑[2]，所以开始展开研究程序性能问题之前，我们需要先了解系统业务逻辑。

铁道部的 12306[3]网站一直被全国人民所诟病，它确实存在一些问题，但是这些看似简单的问题，其背后牵扯着复杂的系统架构设计。这些设计最终是为业务需求服务的，即 12306 的职责是为所有旅客的需求服务的，而程序员设计的程序又是为 12306 服务的，所有的用户体验归到最终就是服务意识。我们来看一下 12306 的业务，12306 需要支持海量并发查询，即海量用户同时查时间、查车次、查座位、查铺位。此外，对应的下单过程也就会伴随着海量并发的数据库操作。据说，淘宝在双十一期间也只有几百万用户[4]，而春运期间抢购火车票是全国人民的统一活动，瞬时访问数量有千万级别甚至是亿级别的。据说 12306 的高峰访问是 10 亿 PV[5]，这些访问主要集中在早 8 点到 10 点，每秒 PV 在高峰时上千万[6]。

再来看看其他的业务系统。奥运会期间的奥运票务系统采用抽奖的方式，这样的业务设计让系统不存在先来先得抢购需求，由于是事后抽奖，因此事前只负责收集信息，所以不需要保证数据的一致性，这也就没有高强度并发锁[7]的需求，很容易通过水平扩展方式克服性能瓶颈。B2C 网站一般实时性要求不高，比如下单，用户提交订单后，订单并不是马上被处理的，而是等待一定时间后，用户才会收到订单是否确认的通知，这样就确保了数据不需要立即被处理，没有了数据高并发同步的需求。也就是说，在高并发要求下的数据一致性是通常情况下的性能瓶颈点，也是通常意义上的技术难点之一。

前面提起过，高并发情况下的数据高度实时一致性需求是很难实现的。对于一个网站来说，并发浏览网页造成的高负载较容易处理，高并发的查询负载也可以处理，但是实时下单是最难处

---

1 Java SE（Java Platform，Standard Edition）；Java EE（Java Platform，Enterprise Edition）；Java ME（Java Platform，Micro Edition）。

2 例如金融系统，不能单纯按照程序员的思维设计系统和数据库结构，而是应该更加紧密地与业务结合。

3 12306：中国铁路客户服务中心（12306 网）是铁路服务客户的重要窗口，将集成全路客货运输信息，为社会和铁路客户提供客货运输业务和公共信息查询服务。

4 来源 2013 年的网络数据，作者非阿里人士，也没有淘宝账户，所以数据不准确请读者见谅。

5 PV：即页面浏览量，通常是衡量一个网络新闻频道或网站甚至一条网络新闻的主要指标。

6 摘自 2013 年的官方数据。

7 提高系统并发吞吐能力的方式，第 5 章会详细介绍。

理的，因为下单需要访问当前的库存量，对于 12306 网站来说，库存量就是指火车票的库存，由于这是一个全国联网系统，所以可以预见库存量保持数据一致性的难度。据说苹果 CEO 库克[8]正是因为处理好了库存问题才得以继任乔帮主[9]的宝座。目前来看，很多 B2C[10]网站的下单都是通过异步方式来实现的，这样的做法可以避免数据高度一致性要求。

淘宝模式相较于传统 B2C 网站有一个优势，即它不需要查询库存。B2C 网站拥有自己的仓库，每次下单前，都需要查找距离客户最近的仓库是否有库存，这样的计算量累计后会很大。比如，你在上海买一本书，如果上海附近的仓库没货，我们需要先计算哪个仓库既离上海最近又有这本书。淘宝网站由于本身商业模式的原因，它不需要去实时检查库存，反而对于性能扩展较为容易。

的确我们可以通过 Nginx[11]来搞定每秒 10 万的静态请求，只要有足够的网络带宽、磁盘 I/O，服务器的并发计算能力够强，可以很容易地处理 10 万的并发连接。但是如果我们引入了大量的业务逻辑，那就不是单纯的访问问题了，该解决方案也就成了浮云。

除了业务需求、程序运行方式之外，程序设计本身需要考虑基础编程技术、系统架构、网络技术、操作系统、硬件服务器等诸多因素。计算机专家在问题求解时非常重视表达式简洁性的价值。UNIX 的先驱者 Ken Thompson[12]曾经说过非常著名的一句话："丢弃 1000 行代码的那一天是我最有成效的一天之一。"这对于任何一个需要持续支持和维护的软件项目来说，都是一个当之无愧的目标。早期的 Lisp[13]贡献者 Paul Graham[14]甚至将语言的简洁性等同为语言的能力。这种对能力的认识让我们把编写紧凑、简洁的代码作为许多现代软件项目选择语言的首要标准。

任何程序都可以通过重构代码方式去除多余的代码或无用的占位符，例如空格，删除空格后会让代码变得更加简短。不过某些语言天生就善于表达，也就特别适合于简短程序的编写。APL 语言的设计理念是利用特殊的图形符号让程序员用很少量的代码就可以编写功能强大的程序。这类程序如果实现得当，可以很好地映射成标准的数学表达式。简洁的语言在快速创建小脚本时非常高效，特别是在目的不会被简洁所掩盖的简洁明确的问题域中。

相比于其他程序设计语言，Java 语言的冗长已经名声在外，主要原因是由于程序开发社区中所形成的惯例。在完成任务时，很多情况下要更大程度地考虑描述性和控制能力。例如，长期来看，长变量名会让大型代码库的可读性和可维护性更强。描述性的类名通常会映射为文件名，在向已有系统中增加新功能时会显得很清晰。如果能够一直坚持下去，描述性名称可以极大简化用于表明应用中某一特定的功能的文本搜索。实践证明，这些定义方式让 Java 在大型复杂代码库的

---

[8] 蒂姆·库克（Tim Cook），1960 年 11 月 1 日出生于美国阿拉巴马州，1982 年毕业于奥本大学工业工程专业。1988 年获得杜克大学企业管理硕士学位。曾在 IBM 供职 12 年，负责 PC 部门在北美和拉美的制造和分销运作。1998 年年初，库克进入苹果，任副总裁，主管苹果的电脑制造业务。2011 年接替乔布斯担任苹果公司 CEO。

[9] 即乔布斯，1955 年 02 月 24 日—2011 年 10 月 05 日。

[10] B2C：Business To Customer。

[11] 一个高性能的 HTTP 和反向代理服务器，也是一个 IMAP/POP3/SMTP 服务器。

[12] 肯·汤普森（Kenneth Lane Thompson，1943 年 2 月 4 日），一般称之为 Ken Thompson，为美国计算机科学学者，与丹尼斯·里奇同为 1983 年图灵奖得主。

[13] LISP 是一种通用高级计算机程序语言，长期以来垄断人工智能领域的应用。Lisp 作为因应人工智能而设计的语言，是第一个函数式程序设计语言，有别于 C、Fortran 等命令式程序设计语言和 Java、C#等面向对象语言。

[14] 保罗·格雷厄姆（Paul Graham），美国著名程序员、风险投资家、博客和技术作家。他以 Lisp 方面的工作而知名，也是最早的 Web 应用 Viaweb 的创办者之一，后来以近 5 千万美元的价格被雅虎收购，成为 Yahoo! Store。

大规模实现中取得了极大的成功。

相对于传统的 32 位虚拟机，64 位虚拟机所具备的最大优势就是可以访问大内存。32 位虚拟机最大可用内存空间被限定在了 4GB，并且 Java 堆区的大小配置存在最大限制，如果是在 Windows 平台下最大只能设置到 1.5GB，而在 Linux 平台下最大也只能设置到 2GB~3GB。也就是说，Java 堆区的内存大小设置还需要依赖于具体的操作系统平台。

既然 32 位虚拟机无法满足大内存消耗的应用场景，那么 64 位虚拟机的出现则是顺理成章的。64 位虚拟机之所以能够访问大内存，是因为其采用了 64 位的指针架构，这也是寻址访问大内存的关键要素。

在 JDK1.6 Update14 版本之前，64 位虚拟机的综合性能表现实际上是不如 32 位虚拟机的，这主要是因为 OOPS（Ordinary Object Pointers，普通对象指针）从 32 位膨胀到 64 位后，CPU Cache Line 中的可用 OOPS 变少，这样一来就会直接影响并降低 CPU 的缓存使用率，这就是 64 位虚拟机在性能上之所以落后于 32 位虚拟机的主要原因。其次，由于部署在 64 位虚拟机上的性能都需要用到大内存，尤其是互联网项目，经常需要使用多达几十乃至几百 GB 的内存，这对于传统的 32 位虚拟机将无法承载，只能依靠 64 位虚拟机去支撑。但是管理这么大的内存开销对于 GC（Garbage Collection）来说将会是一场非常严峻的考验，甚至很有可能会导致 GC 在执行内存回收期间消耗更长的时间，同时也意味着工作线程的等待时间将会延长。如今随着 64 位虚拟机的逐渐成熟，指针压缩将会通过对齐补白等操作将 64 位指针压缩为 32 位，以此改善 CPU 缓存使用率达到提升 64 位虚拟机运行性能的目的。

对于小型项目来说，简洁性则更受青睐，某些语言非常适于短脚本编写或者在命令提示符下的交互式探索编程。Java 作为通用性语言，则更适用于编写跨平台的工具。在这种情况下，“冗长 Java”的使用并不一定能够带来额外的价值。虽然在变量命名等方面，代码风格可以改变，不过从历史情况来看，在一些基本的层面上，与其他语言相比，完成同样的任务，Java 语言仍需更多的字符。为了应对这些限制，Java 语言一直在不断地更新，尝试包含一些通常称为“语法糖”的功能。用这些习语可以实现用更少的字符表示相同功能的目标，与其对应的更加冗长的配对物相比，这些习语更受程序开发社区的欢迎，也会被社区作为通用用法快速地采用。

现代 CPU 架构将多核、多硬件执行线程技术推向前台，这意味着我们可以利用更多的 CPU 资源做更多的工作。然而，要利用好这些额外的 CPU 资源，运行于其上的程序必须要能够支持并行工作这个需求。通俗点讲，这些程序需要按照多线程的方式构造或设计才能充分地利用额外的硬件线程。

最近这几年，服务器端网络使用的基础通信技术并没有取得太大的进步。服务器端大多数在绝不允许服务中断的关键任务环境中，新技术很难渗透，也很难植根于这样的环境。但正因如此，服务器端的多余部分才得以被剔除，逐渐地形成了非常精简单纯的风格。网络的基础技术可以说已经成型了，然而在网络上运行的网络设备和服务器的技术仍然踩着现在进行时的节奏在持续不断地爆发性发展，由此出现了虚拟技术和网络存储技术等基于网络的创新技术。如今，它们已经在系统中不可或缺。随着这些技术的发展，人们追求的网络形态和网络设计的方式也在时刻发生着变化，基础架构工程师和服务器工程师必须能灵活应对这些变化才行。对应地，软件设计程序员也需要有针对性地做出应对措施。