

软件优化技术

——IA-32平台的高性能手册（第2版）

The Software Optimization Cookbook
High-Performance Recipes for IA-32 Platforms(Second Edition)

[美] Richard Gerber, Kevin B. Smith, [荷] Aart J.C. Bik, [加] Xinmin Tian 著
王涛 单久龙 孙广中 等译

```
[C:/cmplr/PS/BOOK/COOK] icl -QxP -xmm0 -xmm1 -xmm2 sum.cpp
sum.cpp(5) : (col. 3) remark: LOOP WAS VECTORIZED.
pxor   xmm0, xmm0      ; setup accumulator
pxor   xmm1, xmm1      ; setup zero vector
xor     eax, eax        ; setup loop index
$B1$2:
movq    xmm2, QWORD PTR a[edx] ; load  4 shorts
punpcklwd xmm2, xmm1      ; zero-ext 4 ints
padd    xmm0, xmm2        ; add  4 ints
add     eax, 8            ;
cmp     eax, 512          ;
jb      $B1$2             ; looping logic
```



软件优化技术

——IA-32平台的高性能手册（第2版）

The Software Optimization Cookbook

High-Performance Recipes for IA-32 Platforms(Second Edition)

[美] Richard Gerber, Kevin B. Smith, [荷] Aart J.C. Bik, [加] Xinmin Tian 著
王涛 单久龙 孙广中 等译

```
(C:\emp\rip5\BOOK\COOK) id -c -x86 -csum.cpp(5) : (col. 3) remark: LOOP WAS VECTORIZED.  
pxor    xmm0, xmm0        ; setup accumulator  
pxor    xmm1, xmm1        ; setup zero vector  
xor     eax, eax          ; setup loop index  
$B1$2:  
movq    xmm2, QWORD PTR a[eax] ; load 4 shorts  
punpcklwd xmm2, xmm1      ; zero-ext 4 ints  
padd    xmm0, xmm2        ; add 4 ints  
add     eax, 8  
cmp     eax, 512  
jbe     $B1$2              ; looping logic
```

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

《软件优化技术——IA-32 平台的高性能手册》是 Intel 出版社最畅销的书籍之一。它填补了软件优化的初级课本与解决专门 Intel®架构的高级手册之间的空白。随着 Intel 扩展内存 64 位技术 (Intel EM64T)、多核处理技术、超线程技术、OpenMP[†]和多媒体扩展等新技术的流行, Intel 出版社邀请另外三位 Intel 专家与原作者一起扩展、更新形成了本书的第 2 版。

《软件优化技术——IA-32 平台的高性能手册》第 2 版提供了如何优化软件,使软件在 Intel 平台上获得巨大性能提升的最新技术。通过简单的讲解和示例,四位 Intel 专家向我们展示了如何从以下一些方面改进软件性能,包括算法、内存访问、分支预测、自动矢量化、SIMD 指令、多线程和浮点计算。想了解最新技术以提高应用性能和想提高编码技巧的开发者,均可以从本书中受益。

The Software Optimization Cookbook, Second Edition

Original English Language edition published by Intel Corporation, Intel Press Business Unit, JF3-330 2111 NE 25th Street, Hillsboro, OR 97124, Copyright©2004 by Intel Corporation, Intel Press Business Unit.

Simplified Chinese-language edition copyright ©2007 by Publishing House of Electronics Industry. All Rights Reserved.

Printed in China.

本书中文简体版专有出版权由 Intel Corporation 授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2007-0314

图书在版编目 (CIP) 数据

软件优化技术: IA-32 平台的高性能手册: 第 2 版 / (美) 格贝尔 (Gerber, R.) 等著; 王涛等译. —北京: 电子工业出版社, 2007.4

书名原文: The Software Optimization Cookbook: High-Performance Recipes for IA-32 Platforms (Second Edition)
ISBN 978-7-121-04005-4

I. 软… II. ①格… ②王… III. 软件设计—手册 IV. TP311.5-62

中国版本图书馆 CIP 数据核字 (2007) 第 036757 号

责任编辑: 顾慧芳

印 刷: 北京民族印刷厂

装 订: 北京鼎盛东极装订有限公司

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: 787×980 1/16 印张: 22.75 字数: 415 千字

印 次: 2007 年 4 月第 1 次印刷

定 价: 59.00 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, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

译者序

随着计算机和网络技术的发展，应用软件变得越发复杂，并深入到社会的每个角落。然而，在不断丰富软件功能的同时，软件性能的优化也日益受到人们的重视。软件运行速度是软件性能的一个重要指标。虽然现在计算机的硬件性能越来越好，价格也日益降低，但运行速度过慢的软件依然无法满足用户对于实时处理的需求。对于软件开发人员来说，掌握软件优化技术是十分有用的。通过有效调整算法和数据结构，使用优化编译选项和并行等优化技术，也可以使软件的运行速度在已有的机器上提高几十倍乃至几千倍。

软件优化技术涉及计算机的各个方面、各个层次，如程序设计语言、数据结构、算法设计与分析、编译技术、操作系统、体系结构、微处理器技术、并行计算等方面。软件开发人员是否可以通过一本书的学习，在较短时间内掌握软件优化的技术？《软件优化技术》就是这样一本能够帮助软件开发人员大幅度提高软件运行速度的有用书籍。本书内容全面、注重实践和实用性强等特点使其成为了 Intel 出版社最畅销的书籍之一。根据英文版的读者反馈意见，本书填补了软件优化的初级课本与解决专门 Intel® 架构的高级手册之间的空白。

本书由 Intel 公司该领域的四位资深专家合著，共 21 章，从各个方面向我们展示了如何进行软件优化，包括算法、内存访问、分支预测、自动矢量化、SIMD 指令、多线程和浮点计算等。作者通过深入浅出的语言，大量的类比和示例，向读者介绍了 Intel 扩展内存 64 位技术 (Intel EM64T)、多核处理技术、超线程技术、OpenMP[†] 等最新技术；引导我们使用 Benchmark、软件工具、编译器开关和代码优化，在 Intel 处理器平台上充分利用硬件与软件资源来实现高速度的软件。对于这本内容丰富、注重实践的畅销书，我们有理由相信它的中文版一定会成为中国广大软件开发人员的良师益友。

本书英文版原名《The Software Optimization Cookbook》让人联想到，这是一

本指导读者按照一定步骤和方法进行软件优化的“菜谱”。本书作者将软件优化类比为高效的快餐馆，软件发挥最大性能的本质就是要使处理器的每个部件都繁忙不停地工作，例如：餐馆为了提高工作效率，经常预先准备好一些食物备用（预先计算），将经常需要的原料放在离工作台最近的位置上（缓存），多个厨师共同制作一种食品或完成一种食物的不同工序（多核，并行）。另外，书中还提供了不少有趣的西式菜谱。在了解最新技术以提高应用性能之余，喜欢烹饪的读者还能从中体会到烹饪的乐趣。

本书的前言、第 1 章到第 12 章由王涛翻译。第 12 章到第 21 章和索引由单久龙翻译。孙广中、王涛和单久龙负责修改和整理全稿。对于本书的出版我们首先要感谢电子工业出版社的郭立老师和 Intel 出版社的胡燕老师，是她们的努力促成了本书的顺利翻译并迅速出版，使中国读者能够通过本书了解 Intel 更多的先进技术，更好地进行软件优化。同时，我们也特别感谢英特尔公司的郭胜、丛杨、刘嘉骥、廖世伟、Xinmin Tian，英特尔中国研究中心的栗强、李文龙、陈玉荣、张益民和中国科学技术大学的黎鹤、苗乾坤，他们在本书的翻译和审校过程中给予了很多热心的支持和技术上的帮助。

在翻译过程中，我们力求忠实、准确地把握原著，同时保留原著的风格。但由于译者水平有限以及时间仓促，书中难免有错误和不准确之处，恳请广大读者批评指正。

王 涛 单久龙 孙广中

2007 年 2 月

于英特尔（中国）研究中心 北京

前 言

《软件优化技术》是 Intel 出版社最畅销的书籍之一。读者反馈表明，本书填补了软件优化的初级课本与解决专门 Intel®架构的高级手册之间的空白。随着 Intel 扩展内存 64 位技术 (Intel EM64T)、多核处理技术、超线程技术、OpenMP[†] 和多媒体扩展技术的流行，该书第 1 版的内容显得有些陈旧。因此，Intel 出版社邀请另外三位 Intel 专家与原作者一起扩展、更新形成了本书的第 2 版。

《软件优化技术》第 2 版提供了在 Intel 平台上部署高性能应用的最新技术。作者通过简单的解释和示例，向你展示了如何解决具有算法、内存访问、分支预测、自动矢量化、SIMD 指令、多线程和浮点计算等内容的性能相关问题。软件开发者可以从中学习如何利用 Intel EM64T、多核处理技术、超线程技术、OpenMP 和多媒体扩展技术进行软件优化。这本书引导你通过使用软件工具、编译器开关和代码优化，高效地改善基于 Intel 平台的应用性能。想要了解最新技术以提高应用性能和想提高编码技巧的软件开发者均可以从本书中受益。

致 谢

感谢对本书、Intel 编译器和性能分析工具作出贡献的每一个人。对下面这些人的努力要表示特别的感谢：Zia Ansari, Pete Baker, Mitch Bodart, Christopher Bord, Mark Buxton, Ryan Carlson, Yen-Kuang Chen, Joshua Chia, Martyn Corden, Robert Cox, William E. Damon III, Max J. Domeika, Milind Girkar, Koby Gottlieb, Grant Haab, Mohammad Haghighat, Jay Hoeflinger, John Holm, Alexander Isaev, Michael Julier, Wei Li, Christopher Lishka, Diana King, Knud Kirkegaard, David Kreitzer, Tim Mattson, Eric Moore, Dan Macri, Andrey Naraikin, Kannan Narayanan, Clark Nelson, John Ng, Paul Peterson, Michael Ross, Hideki Saito, Sanjiv Shah, Ernesto Su, Sara Sarmiento, William Savage, Dale Schouten, David Sehr, Ronak Singhal, Kevin J. Smith, Stacey Smith, Craig Stoller, Bob Valentine 和 Ronen Zohar.

特别感谢为本书提供反馈意见的审阅者：原 Intel 公司的 Tim Carver, GE Global Research 的 Walt Dixon, 澳大利亚 Scandpower Petroleum Technology 公司的 Lars Petter Endresent,, 联邦邮政服务的 James H. Hill, Jeppesen/Boeing 公司的 Brian Kennedy, Zeesoft 公司的 Jitendra Maheshwari, 康萨斯大学自然历史博物馆的 Kevin Ruland 和佛罗里达州立大学的 Robert van Engelen。

在写作过程中, Intel 出版社给予了大力帮助。我们要特别感谢编辑 David Spencer 和内容架构师 Stuart Goldstein, 他们在本书出版的全过程中给予了热心的关怀和专业的指导。

读者也能够给予帮助

为了出版本书, 作者在原书第 1 版的基础上更新了其中的大部分内容。特别是, Richard Gerber 更新整理了全书。Aart Bik 更新了第 2 章、第 9 章和第 12 章并增加了第 13 章自动矢量化。Kevin Smith 更新了第 3 到第 8 章、第 10 章、第 11 章和第 14 章。Xinmin Tian 更新了第 15 章并增加了关于 OpenMP 并行化的第 16 章、第 17 章和第 18 章。

尽管付出了很多努力, 我们仍然渴望不断更新本书的内容。我们喜欢听到读者的问题和要求, 以帮助大家正确理解本书的内容。在本书的网站: www.intel.com/intelpress/swcb2, 我们提供了联系方式、勘误表和其他相关的资料。

Richard Gerber	Kevin B. Smith
Aart J. C. Bik	Xinmin Tian

目 录

第一部分 性能分析工具和基本概念（开胃品）

第 1 章 引言	3
软件优化	4
软件优化陷阱	4
软件优化过程	6
要点	7
第 2 章 测试基准	9
Benchmark 的属性	10
可重复性（必须）	10
代表性（必须）	11
易用性（必须）	11
可验性（必须）	11
时间可测性（可选）	11
完全覆盖性（条件依赖）	12
精确性（条件依赖）	12
Benchmark 示例	12
要点	16
第 3 章 性能分析工具	18
计时机制	18
优化编译器	20
使用 Intel® C++ 和 Fortran 编译器	20
特定处理器优化	21
针对特定处理器的函数实现	22
其他编译器优化	23

软件分析的类型	24
性能监测	25
VTune™性能分析器	25
采样	26
调用图分析	27
Intel 编译器 Codecov 分析器	28
Microsoft Visual C++分析器	29
采样与插桩型分析器的比较	30
尝试、错误、判断力与耐心	31
要点	32
第 4 章 热点	34
什么因素导致热点和冷点?	36
不仅仅是时间	36
均匀执行与没有热点	37
要点	39
第 5 章 处理器体系结构	41
功能模块	42
要两个汉堡!	43
取指与译码	45
指令执行	47
退出	50
寄存器和内存	50
要点	52
第二部分 性能相关问题（主菜）	
第 6 章 算法	55
计算复杂度	55
指令选择	56

数据相关与指令并行	61
内存需求	63
并行算法	64
算法的通用性	65
检测算法问题	65
要点	69
第 7 章 分支	72
找到关键的预测错误的分支	74
第一步：找到预测错误的分支	74
第二步：找到耗时的热点	75
第三步：确定预测错误的分支的比率	76
最后的全面检查	77
分支的不同类型	78
使分支更容易被预测	80
使用 CMOV 移除分支	81
使用标志移除分支	83
使用 min/max 指令移除分支	84
通过额外工作去除分支	85
要点	86
第 8 章 内存	88
内存概览	89
主存和虚拟内存	89
处理器缓存	90
缓存细节	92
硬件预取	93
软件预取	94
不通过缓存写数据：非临时性写	95
影响内存性能的问题	96
强制缓存载入	96

容量缓存载入	97
冲突缓存载入	97
缓存效率	98
存储转发	99
数据对齐	100
编译器和数据对齐	101
软件预取	102
检测内存问题	103
找到页缺失	104
找到存储转发问题	106
找到 L1 缓存缺失	106
理解潜在的改进	108
修复内存问题	109
要点	115
第 9 章 循环	117
数据相关	118
循环分配和融合	120
循环剥离	122
循环展开与合并	123
循环交换	126
循环不变计算	128
循环不变分支	128
循环不变结果	129
要点	130
第 10 章 慢操作	132
慢指令	132
查找表	134
系统调用	137
系统空闲进程	140

要点	143
第 11 章 浮点	145
数值异常	145
清零和非格式化到零	148
精度	149
打包和标量模式	152
浮点到整数的转换, 取整	153
下界和上界函数	154
浮点操作技巧	154
浮点类型到整型的转换	155
平方根	155
倒数平方根	155
要点	156
第 12 章 SIMD 技术	158
SIMD 技术介绍	159
MMX™ 技术	159
SIMD 流扩展	160
使用 SIMD 技术	161
自动矢量化	162
C++ 类库	163
Intrinsics	164
嵌入汇编语言	165
四种方法的优缺点	166
SIMD 技术的考虑事项	167
决定何处使用 SIMD 技术	167
内存对齐	167
数据布局	169
选择合适的打包数据类型	171
SIMD 和 x87 FPU 计算的兼容性	173

要点	173
第 13 章 自动矢量化	176
矢量化相关的编译开关	176
常用的编译开关	176
编译开关使用示例	179
矢量化相关的编译提示	180
常用的编译提示	180
编译提示使用示例	185
矢量化指南	186
设计与实现的考虑	187
矢量化诊断的使用	189
别名与负面影响的最小化	192
编程风格	196
目标体系结构	197
要点	200
第 14 章 面向处理器的特殊优化	202
32 位 Intel® 体系结构	202
Pentium® M 处理器	204
一级指令缓存	205
指令译码	206
指令延迟	207
指令集	208
浮点控制寄存器	208
MXCSR 状态寄存器	209
一级数据缓存	209
内存预取	209
处理器事件	210
部分寄存器延迟	210
部分标志延迟	212

暂停指令.....	212
要点.....	213
第 15 章 多处理技术介绍.....	214
并行程序设计.....	215
线程管理.....	217
使用 OpenMP [†] 高层线程化.....	217
低层线程.....	220
线程化的目标.....	220
线程化的相关问题.....	222
Intel 编译器与线程工具.....	225
要点.....	226
第 16 章 OpenMP[†] 多线程.....	228
OpenMP [†] 的关键要素.....	228
多线程执行模型.....	232
OpenMP [†] 的存储模型.....	234
OpenMP [†] 的限制.....	238
编译 OpenMP [†] 程序.....	239
自动并行化.....	241
线程化指南.....	244
要点.....	246
第 17 章 任务队列和 OpenMP[†] 的高等应用专题.....	249
任务队列—Intel 对 OpenMP [†] 的扩展.....	249
任务队列执行模型.....	249
任务队列和任务结构.....	252
线程化 N 皇后问题：一个实例研究.....	255
线程级流水线并行.....	260
开发嵌套式并行性.....	263
多级并行.....	267

线程绑定研究	269
理解循环调度	271
要点	274

第三部分 设计和应用优化（甜点）

第 18 章 实例研究：线程化视频编码器	279
H.264 视频编码器的初始性能	279
H.264 视频编码器的并行化	280
任务和数据划分	281
切片级并行性	282
帧级并行	283
基于两个切片队列的实现	284
基于任务队列模型的实现	285
性能	288
速度提高与压缩效率间的折衷	288
在支持 HT 技术的多处理器上的性能	290
性能的理解	291
多线程的额外开销	294
进一步的性能调试	295
线程化总结	296
要点	297
第 19 章 针对性能的设计	298
数据移动	299
内存与并行性	299
设计过程中的性能实验	300
算法	301
要点	303

第 20 章 综合：基本的优化	306
采摘低处的果实	306
应用	307
相关资料	309
Benchmark	309
Benchmark 结果的理解	310
浮点到长整型转换的改进	310
算法的并行化	312
采用自动矢量化	313
基于 Intrinsics 的指令级并行	314
优化总结	315
要点	316
第 21 章 综合：最后的百分之十	318
最快的速度	318
更高的 SIMD 效率	320
最终的优化	323
优化总结	324
要点	325
参考文献	326
索引	332

第一部分

性能分析工具和基本概念 (开 胃 品)