

<<多核应用编程实战>>

图书基本信息

书名：<<多核应用编程实战>>

13位ISBN编号：9787115317506

10位ISBN编号：711531750X

出版时间：2013-6

出版时间：人民邮电出版社

作者：[美] Darryl Gove

译者：郭晴霞

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：<http://www.tushu007.com>

<<多核应用编程实战>>

内容概要

《多核应用编程实战》是一本全面实用的多核应用编程指南，旨在介绍如何编写功能正确、性能优越且适合扩展为在多个CPU核心的系统运行的应用程序。

《多核应用编程实战》面向多种操作系统和处理器类型引用程序示例，内容涵盖类UNIX操作系统(Linux、Oracle Solaris、OS X)和Windows系统上多核应用的编写方法、多核的硬件实现对应用程序的性能影响、编写并行应用程序时要避免的潜在问题，以及如何编写可扩展至大量并行线程的应用程序。

《多核应用编程实战》适合所有C程序员学习参考。

<<多核应用编程实战>>

作者简介

Darryl Gove

Oracle Solaris Studio编译团队的高级首席软件工程师，主要负责应用程序和基准方面的分析、并行和优化。

他具有英国南安普顿大学运筹学硕士和博士学位，另著有Solaris Application Programming、The Developer ' s Edge。

此外，他经常在个人博客www.darrylgove.com上发表关于性能、优化、多线程和编码的文章。

<<多核应用编程实战>>

书籍目录

第1章 硬件、进程和线程	1
1.1 计算机的内部结构	1
1.2 多核处理器的缘起	3
1.2.1 在单芯片上支持多线程	4
1.2.2 通过处理器核心流水线作业提高指令发出率	8
1.2.3 使用缓存保存最近使用的数据	10
1.2.4 用虚拟内存存储数据	12
1.2.5 从虚拟地址转换到物理地址	13
1.3 多处理器系统的特征	14
1.4 源代码到汇编语言的转换	16
1.4.1 32位与64位代码的性能	18
1.4.2 确保内存操作的正确顺序	19
1.4.3 进程和线程的差异	21
1.5 小结	23
第2章 高性能编码	24
2.1 定义性能	24
2.2 了解算法复杂度	25
2.2.1 算法复杂度的示例	26
2.2.2 算法复杂度的重要性	28
2.2.3 谨慎运用算法复杂度	30
2.3 结构如何影响性能	30
2.3.1 在源代码和生成结构上权衡性能和便利性	30
2.3.2 利用库结构化应用程序	33
2.3.3 数据结构对性能的影响	42
2.4 编译器的作用	47
2.4.1 两种编译器优化	48
2.4.2 选择合适的编译器选项	50
2.4.3 如何用跨文件优化提高性能	51
2.4.4 使用配置文件反馈	53
2.4.5 潜在的指针别名会如何抑制编译器优化	55
2.5 通过分析确定占用时间的地方	58
2.6 怎样避免手动优化	64
2.7 从设计角度看性能	64
2.8 小结	65
第3章 识别并行机会	66
3.1 使用多进程提高系统工作效率	66
3.2 多用户使用一个系统	67
3.3 通过整合提高机器工作效率	68
3.3.1 用容器隔离共享一个系统的应用程序	69
3.3.2 使用虚拟机监控程序托管多个操作系统	69
3.4 采用并行机制提高单个任务的性能	71
3.4.1 理解并行应用程序	72
3.4.2 并行如何影响算法的选择	72
3.4.3 Amdahl定律	73
3.4.4 确定最大实际线程数	75

<<多核应用编程实战>>

3.4.5	同步成本怎样降低扩展性	76
3.5	并行模式	78
3.5.1	使用SIMD指令的数据并行	78
3.5.2	通过进程或线程实现并行化	79
3.5.3	多个独立任务	79
3.5.4	多个松散耦合的任务	80
3.5.5	相同任务的多个副本	81
3.5.6	单个任务拆分到多个线程	82
3.5.7	使用流水线任务完成某个事项	82
3.5.8	将工作分配给客户端和服务端	83
3.5.9	将责任划分给生产者和消费者	84
3.5.10	结合多种并行化策略	85
3.6	依赖关系对并行运行代码能力的影响	85
3.6.1	反依赖和输出依赖	86
3.6.2	通过推测打破依赖	88
3.6.3	关键路径	91
3.7	发现并行机会	92
3.8	小结	93
第4章	同步和数据共享	94
4.1	数据争用	94
4.1.1	使用工具检测数据争用	95
4.1.2	避免数据争用	98
4.2	同步原语	98
4.2.1	互斥量和临界区	98
4.2.2	自旋锁	99
4.2.3	信号量	100
4.2.4	读写锁	100
4.2.5	屏障	101
4.2.6	原子操作和无锁代码	102
4.3	死锁和活锁	103
4.4	线程和进程间的通信	104
4.4.1	内存、共享内存和内存映射文件	104
4.4.2	条件变量	105
4.4.3	信号和事件	107
4.4.4	消息队列	108
4.4.5	命名管道	108
4.4.6	通过网络栈进行通信	109
4.4.7	线程之间共享数据的其他方法	110
4.5	存储线程私有数据	110
4.6	小结	112
第5章	使用POSIX线程	113
5.1	创建线程	113
5.1.1	线程终止	114
5.1.2	用子线程接收和传递数据	115
5.1.3	分离线程	116
5.1.4	设置pthread的属性	117
5.2	编译多线程代码	119

<<多核应用编程实战>>

5.3	进程终止	121
5.4	线程之间共享数据	122
5.4.1	使用互斥锁保护访问	122
5.4.2	互斥锁属性	124
5.4.3	使用自旋锁	125
5.4.4	读写锁	127
5.4.5	屏障	129
5.4.6	信号量	130
5.4.7	条件变量	136
5.5	变量和内存	140
5.6	多进程编程	143
5.6.1	在进程之间共享内存	144
5.6.2	在进程之间共享信号量	147
5.6.3	消息队列	147
5.6.4	管道和命名管道	150
5.6.5	使用信号与进程通信	151
5.7	套接字	156
5.8	可重入代码和编译器标志	158
5.9	小结	160
第6章	Windows线程	161
6.1	创建Windows本机线程	161
6.1.1	终止线程	165
6.1.2	创建和重新启动挂起的线程	167
6.1.3	使用内核资源的句柄	168
6.2	同步和资源共享的方式	168
6.2.1	线程间需要同步的一个例子	169
6.2.2	保护对临界区代码的访问	170
6.2.3	用互斥量保护代码段	172
6.2.4	轻量级读写锁	173
6.2.5	信号量	175
6.2.6	条件变量	177
6.2.7	向其他线程或进程发出事件完成的信号	178
6.3	Windows中的宽字符串处理	179
6.4	创建进程	180
6.4.1	在进程之间共享内存	182
6.4.2	在子进程中继承句柄	185
6.4.3	互斥量命名及其在进程间的共享	186
6.4.4	用管道通信	187
6.4.5	用套接字进行通信	190
6.5	变量的原子更新	193
6.6	分配线程本地存储	195
6.7	设置线程的优先级	197
6.8	小结	198
第7章	自动并行化和OpenMP	199
7.1	使用自动并行化产生并行代码	199
7.1.1	识别和并行约简	203
7.1.2	对包含调用的代码进行自动并行化	204

<<多核应用编程实战>>

7.1.3	协助编译器实现代码的自动并行化	206
7.2	使用OpenMP生成并行应用程序	208
7.2.1	使用OpenMP并行化循环	209
7.2.2	OpenMP应用程序的运行时行为	210
7.2.3	OpenMP并行区域中的变量作用域	210
7.2.4	使用OpenMP并行化约简	212
7.2.5	在并行区域外访问私有数据	212
7.2.6	使用调度改进工作分配	214
7.2.7	用并行段完成独立工作	217
7.2.8	嵌套并行	218
7.2.9	使用OpenMP动态定义并行任务	219
7.2.10	保持数据对线程私有	223
7.2.11	控制OpenMP运行时环境	225
7.2.12	等待工作完成	227
7.2.13	限制执行代码区域的线程	229
7.3	确保并行区域的代码按顺序执行	232
7.4	折叠循环改进工作负荷均衡	233
7.5	强制实现内存一致性	234
7.6	并行化示例	235
7.7	小结	239
第8章	手工编码的同步和共享	240
8.1	原子操作	240
8.1.1	用比较和交换指令构成更复杂的原子操作	242
8.1.2	强制实现内存排序以确保正确操作	245
8.1.3	编译器对内存排序指令的支持	247
8.1.4	编译器对操作的重新排序	247
8.1.5	易失变量	251
8.2	操作系统提供的原子操作	251
8.3	无锁算法	254
8.3.1	Dekker算法	254
8.3.2	带循环缓存的生产者/消费者	256
8.3.3	扩展到多个消费者或生产者	259
8.3.4	将生产者/消费者扩展到多个线程	260
8.3.5	更改生产者/消费者代码为使用原子操作	266
8.3.6	ABA问题	268
8.4	小结	271
第9章	基于多核处理器的扩展	272
9.1	对应用程序扩展的限制	272
9.1.1	串行代码对性能的限制	272
9.1.2	超线性扩展	275
9.1.3	工作负荷不均衡	276
9.1.4	热锁	277
9.1.5	库代码扩展	282
9.1.6	工作量不足	284
9.1.7	算法限制	286
9.2	扩展的硬件限制	288
9.2.1	核心之间的带宽共享	288

<<多核应用编程实战>>

9.2.2	伪共享	290
9.2.3	缓存冲突和容量	293
9.2.4	流水线资源匮乏	297
9.3	操作系统对扩展性的限制	301
9.3.1	过度订阅	301
9.3.2	使用处理器绑定改善内存局部性	303
9.3.3	优先级反转	310
9.4	多核处理器和扩展	310
9.5	小结	311
第10章	其他并行技术	312
10.1	基于GPU的运算	312
10.2	语言扩展	314
10.2.1	线程构建模块	314
10.2.2	Cilk++	317
10.2.3	Grand Central Dispatch	320
10.2.4	为未来C和C++标准提议的可能功能	321
10.2.5	微软的C++/CLI	324
10.3	其他语言	325
10.4	集群技术	327
10.4.1	MPI	328
10.4.2	以MapReduce作为扩展策略	331
10.4.3	网格	332
10.5	事务性内存	332
10.6	向量化	333
10.7	小结	334
第11章	结束语	335
11.1	编写并行应用程序	335
11.1.1	识别任务	335
11.1.2	估算性能提升	336
11.1.3	确定依赖关系	336
11.1.4	数据争用和互斥锁扩展限制	336
11.1.5	锁的粒度	337
11.2	多核处理器上的并行代码	337
11.3	并行化的未来	339
参考文献		340
索引		342

<<多核应用编程实战>>

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:<http://www.tushu007.com>