

《算法与并行计算》

图书基本信息

书名：《算法与并行计算》

13位ISBN编号：9787302290094

10位ISBN编号：7302290091

出版时间：2012-11

出版社：清华大学出版社

作者：Fayez Gebali

页数：248

译者：都志辉

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

《算法与并行计算》

内容概要

算法与并行计算，ISBN：9787302290094，作者：（美）格巴里 著，都志辉 等译

《算法与并行计算》

作者简介

作者：（美国）格巴里（Fayez Gebali）译者：都志辉

书籍目录

第1章 引言 1

1.1 概述 1

1.2 自动并行编程 1

1.3 算法 3

1.3.1 算法的有向图 3

1.3.2 算法的邻接矩阵A 4

1.3.3 基于子任务的依赖关系对算法进行分类 5

1.3.4 串行算法 6

1.3.5 并行算法 6

1.3.6 SPA 6

1.3.7 NSPA 7

1.3.8 RIA 8

1.3.9 并行算法实现 8

1.4 设计并行计算系统 9

1.5 并行算法和并行体系结构 10

1.6 并行算法与并行体系结构相关 10

1.7 算法的实现：两个方面的问题 11

1.8 衡量并行计算的优势 11

1.8.1 加速比 11

1.8.2 通信开销 12

1.8.3 计算加速比和通信开销 12

1.9 针对多处理器系统的Amdahl法则 14

1.10 Gustafson-Barsis法则 15

1.11 并行计算的应用 16

1.11.1 气象建模 16

1.11.2 CT 17

1.11.3 计算流体力学 (CFD) 18

1.12 习题 18

第2章 增强单处理器的性能 21

2.1 概述 21

2.2 提高处理器的时钟频率 21

2.3 ALU的并行化 22

2.4 使用分级存储器体系 24

2.4.1 内存-高速缓存之间的操作 25

2.4.2 高速缓存的设计 26

2.4.3 分层高速缓存 26

2.4.4 将内存块映射到高速缓存行 26

2.4.5 关联映射 27

2.4.6 组相关映射 28

2.4.7 缓存容量对缓存命中率的影响 28

2.5 流水线作业 28

估算流水线作业的速度 29

2.6 超长指令字 (VLIW) 处理器 32

2.7 指令级并行 (ILP) 和超标量处理器 33

2.7.1 真实数据依赖：写后读 (RAW) 34

2.7.2 程序的依赖关系 35

2.7.3 资源冲突 35

2.7.4 输出依赖性：写后写 (WAW) 35

- 2.7.5 反依赖：读后写 (WAR) 36
- 2.8 多线程处理器 36
- 2.9 习题 37
- 第3章 并行计算机 39
 - 3.1 概述 39
 - 3.2 并行计算 39
 - 3.3 共享内存的多处理器 (统一内存访问UMA) 40
 - 3.4 分布式内存多处理器 (非统一内存访问NUMA) 41
 - 3.5 SIMD处理器 41
 - 3.6 脉动式处理器 42
 - 3.7 集群计算 44
 - 3.8 网格计算 (云计算) 44
 - 3.9 多核系统 44
 - 3.10 流多处理器 46
 - 3.11 并行处理器之间的通信 48
 - 3.11.1 通信类型 48
 - 3.11.2 消息传递 (MP) 通信机制 49
 - 3.12 并行体系结构总结 50
 - 3.13 习题 50
- 第4章 共享内存多处理器 52
 - 4.1 概述 52
 - 4.2 高速缓存一致性和内存一致性 53
 - 4.2.1 目录协议 56
 - 4.2.2 Snoopy协议 57
 - 4.3 同步和互斥 57
 - 4.3.1 同步：锁机制 58
 - 4.3.2 同步：互斥量 59
 - 4.3.3 同步：栅栏 60
 - 4.3.4 同步原语的对比 61
 - 4.4 习题 62
- 第5章 互连网络 63
 - 5.1 概述 63
 - 5.2 逻辑拓扑结构中互连网络的分类 63
 - 5.2.1 总线型 63
 - 5.2.2 星型 64
 - 5.2.3 环型 64
 - 5.2.4 网型 64
 - 5.2.5 交叉开关网络 65
 - 5.2.6 交叉开关网络的连接及仲裁 66
 - 5.2.7 多级互连网络 66
 - 5.2.8 榕树 (Banyan) 网络 66
 - 5.2.9 树型网络 67
 - 5.2.10 随机拓扑网络 68
 - 5.3 互连网络交换架构 68
 - 5.3.1 输入队列交换器 69
 - 5.3.2 输出队列交换器 70
 - 5.3.3 共享缓冲区交换器 71
 - 5.3.4 多输入队列交换器 73
 - 5.3.5 多输出队列交换器 73

- 5.3.6 多输入输出队列交换器 74
- 5.3.7 VRQ交换器 75
- 5.4 习题 76
- 第6章 并发平台 78
 - 6.1 概述 78
 - 6.2 并发平台 78
 - 6.3 Cilk++ 78
 - 6.3.1 Cilk++并行循环：cilk_for 79
 - 6.3.2 数据竞争和程序不确定性 80
 - 6.3.3 将串行代码并行化的Cilk++组件 82
 - 6.3.4 使用Cilk++实现矩阵乘法 82
 - 6.4 OpenMP 84
 - 6.4.1 OpenMP编译指导语句 85
 - 6.4.2 编译指导语句子句 86
 - 6.4.3 OpenMP负载分配 87
 - 6.4.4 循环指导语句：for 87
 - 6.4.5 循环指导语句：sections 89
 - 6.4.6 运行时库函数 90
 - 6.4.7 环境变量 90
 - 6.4.8 OpenMP同步 90
 - 6.5 统一计算设备架构 (CUDA) 91
 - 6.5.1 定义CUDA中的线程、块和网格 93
 - 6.5.2 将函数交付内核执行 94
 - 6.5.3 主机与CUDA设备间的通信 95
 - 6.5.4 CUDA线程的同步与通信 95
 - 6.5.5 内核和网格 95
 - 6.5.6 块 97
 - 6.5.7 线程 97
 - 6.5.8 CUDA C语言扩展 97
- 第7章 针对并行算法的特别技术 98
 - 7.1 概述 98
 - 7.2 定义算法变量 99
 - 7.3 独立循环调度 99
 - 7.4 依赖循环 100
 - 7.5 针对简单依赖循环的循环分发方法 100
 - 7.6 循环展开 101
 - 7.7 问题划分 101
 - 7.8 分而治之(递归划分)策略 102
 - 7.9 流水线 104
 - 7.10 习题 106
- 第8章 非串行-并行算法 107
 - 8.1 概述 107
 - 8.2 并行化用DAG表示的NSPA算法 108
 - 8.3 分析NSPA的形式化方法 109
 - 矩阵的幂的意义: 矩阵的连通性 110
 - 8.4 辨别算法中的环 112
 - 8.5 提取串行及并行算法的性能参数 113
 - 8.6 相关定理 114
 - 8.7 串行和并行算法在并行计算机上的性能 116

- 8.8 习题 116
- 第9章 z-变换分析 118
 - 9.1 概述 118
 - 9.2 z-变换的定义 118
 - 9.3 一维有限脉冲响应滤波器算法 119
 - 9.4 z-变换的软件硬件实现 119
 - 9.5 设计1: 用霍纳法则实现广播输入管道输出 120
 - 9.6 设计2: 管道输入广播输出 121
 - 9.7 设计3: 管道输入管道输出 122
 - 9.8 习题 123
- 第10章 依赖关系图分析 124
 - 10.1 概述 124
 - 10.2 一维有限冲击响应滤波算法 124
 - 10.3 算法的依赖关系图 124
 - 10.4 计算算法的依赖关系图 125
 - 定义D中的变量 125
 - 10.5 一维有限冲击响应滤波的调度函数 127
 - 10.5.1 将依赖关系图转换为有向无环图或串行-并行算法 127
 - 10.5.2 广播变量 128
 - 10.5.3 流水变量 128
 - 10.5.4 确定调度函数 129
 - 10.5.5 线性线程/任务调度的限制 130
 - 10.5.6 非线性调度操作 131
 - 10.6 结点投影操作 131
 - 10.7 非线性投影操作 132
 - 使用并发平台 133
 - 10.8 有向无环图分析的软件和硬件实现 133
 - 10.8.1 设计方案1: 投影方向 $d_1 = \backslash t$ 133
 - 10.8.2 设计方案2: 投影方向 $d_2 = \backslash t$ 134
 - 10.9 习题 135
- 第11章 计算几何分析 136
 - 11.1 概述 136
 - 11.2 矩阵乘算法 136
 - 11.3 3D依赖图和计算域D 136
 - 3D域边界 137
 - 11.4 D的面和顶点 138
 - 11.5 算法变量的依赖矩阵 138
 - 11.6 依赖矩阵的零空间: 广播子域B 139
 - A的零空间 139
 - 11.7 设计空间的探索: 选择广播变量还是流水线变量 141
 - 11.7.1 馈送/提取广播变量的点 141
 - 11.7.2 变量流水线 143
 - 11.8 数据调度 143
 - 调度函数对数据时序的影响 146
 - 11.9 使用线性投影算子进行投影操作 147
 - 11.9.1 投影矩阵P 147
 - 11.9.2 投影方向 148
 - 11.9.3 投影方向d的选择 148
 - 11.9.4 当投影方法d给定时, 找出矩阵P 149

- 11.10 投影操作对数据的影响 150
 - 11.10.1 输出数据 150
 - 11.10.2 输入数据M2 151
 - 11.10.3 输入数据M3 151
- 11.11 最终的多线程/多处理器体系结构 151
- 11.12 本章总结 152
- 11.13 习题 152
- 第12章 实例: 一维IIR数字滤波器 154
 - 12.1 概述 154
 - 12.2 一维IIR数字滤波器算法 154
 - 12.3 IIR滤波器的依赖图 154
 - 12.3.1 二维依赖图 154
 - 12.3.2 一维滤波器的调度函数 155
 - 12.3.3 投影方向和投影矩阵的选择 157
 - 12.3.4 设计1: 投影方向 157
 - 12.3.5 设计2: 投影方向 157
 - 12.4 一维IIR数字滤波器算法的z域分析 159
 - 12.4.1 设计3: 广播输入和流水线输出 159
 - 12.4.2 流水线输入和广播输出 159
 - 12.4.3 设计4: 流水线输入和输出 159
 - 12.5 习题 161
- 第13章 案例分析: 二维与三维数字滤波器 162
 - 13.1 概述 162
 - 13.2 行和帧环绕问题 162
 - 13.3 二维递归滤波器 163
 - 13.3.1 二维IIR设计1: 广播XY输入、流水输出 163
 - 13.3.2 二维IIR设计2: 流水XY输入、广播输出 164
 - 13.4 三维数字滤波器 165
 - 13.4.1 三维IIR设计1: 广播XY输入、流水输出 166
 - 13.4.2 三维IIR设计2: 流水化X和Y输入、广播输出 166
- 第14章 实例分析: 多重速率的采样器和插值器 168
 - 14.1 概述 168
 - 14.2 采样器的架构 168
 - 14.3 采样器的依赖关系图 169
 - 14.4 采样器时序 170
 - 14.5 在 $s_1=\backslash$ 的情况下, 采样器的有向无环图 171
 - 14.6 在 $s_2=\backslash$ 的情况下, 插值器的有向无环图 172
 - 14.7 在 $s_3=\backslash$ 的情况下, 插值器的有向无环图 174
 - 14.8 多相采样器的实现 174
 - 14.9 插值器的架构 175
 - 14.10 插值器的依赖关系图 176
 - 14.11 插值器的调度 177
 - 14.12 在 $s_1=\backslash$ 的情况下, 插值器的有向无环图 178
 - 14.13 在 $s_2=\backslash$ 的情况下, 插值器的有向无环图 179
 - 14.14 在 $s_3=\backslash$ 的情况下, 插值器的有向无环图 180
 - 14.15 多相插值器的实现 181
- 第15章 案例学习: 模式匹配 182
 - 15.1 概述 182
 - 15.2 将算法表达为正则迭代算法(RIA) 182

- 15.3 得到算法依赖图 183
- 15.4 数据调度 183
- 15.5 DAG结点的投影 184
- 15.6 设计方案1: 当 $s=\backslash t$ 时的设计空间 184
 - 15.6.1 设计方案1.a: 设 $s=\backslash t$, $da=\backslash t$ 185
 - 15.6.2 设计方案1.b: 设 $s=\backslash t$, $db=\backslash t$ 186
 - 15.6.3 设计方案1.c: 设 $s=\backslash t$, $dc=\backslash t$ 186
- 15.7 设计方案2: 当 $s=\backslash t$ 时的设计空间搜索 187
 - 15.7.1 设计方案2.a: 设 $s=\backslash t$, $da=\backslash t$ 187
 - 15.7.2 设计方案2.b: 设 $s=\backslash t$, $db=\backslash t$ 187
 - 15.7.3 设计方案2.c: 设 $s=\backslash t$, $dc=\backslash t$ 188
- 15.8 设计方案3: 当 $s=\backslash t$ 时的设计空间搜索 188
 - 设计方案3.a: 设 $s=\backslash t$, $da=\backslash t$ 188
- 第16章 案例学习: 用于视频压缩的运动估计 189
 - 16.1 概述 189
 - 16.2 FBMA 189
 - 16.3 数据缓冲要求 190
 - 16.4 FBMA的形式化 191
 - 16.5 运动估计的分层形式化 191
 - 16.5.1 第3层(最左层) 192
 - 16.5.2 第2层 192
 - 16.5.3 第1层 192
 - 16.5.4 第0层(最右层) 192
 - 16.6 层次化结构块的硬件设计 193
 - 16.6.1 第3层的硬件设计 193
 - 16.6.2 第2层的硬件设计 196
 - 16.6.3 第1层的硬件设计 197
 - 16.6.4 第0层的硬件设计 197
- 第17章 范例分析: $2m$ 阶伽罗瓦域乘法 198
 - 17.1 概述 198
 - 17.2 $2m$ 阶伽罗瓦域乘法算法 198
 - 17.3 将域乘法表示为RIA 200
 - 17.4 域乘法的依赖图 200
 - 17.5 数据调度 201
 - 17.6 DAG结点投影 203
 - 17.7 设计1: 使用 $d1=\backslash t$ 204
 - 17.8 设计2: 使用 $d2=\backslash t$ 204
 - 17.9 设计3: 使用 $d3=\backslash t$ 205
 - 17.10 有限域乘法器的应用 206
- 第18章 范例分析: $2m$ 阶伽罗瓦域的多项式除法 207
 - 18.1 概述 207
 - 18.2 多项式除法算法 207
 - 18.3 LFSR依赖图 208
 - 18.4 数据调度 209
 - 18.5 DAG结点投影 210
 - 18.6 设计1: $s1=\backslash t$ 时的设计空间 211
 - 18.7 设计2: $s2=\backslash t$ 时的设计空间 212
 - 18.8 设计3: $s3=\backslash t$ 时的设计空间 214
 - 18.9 3种设计方案的比较 215

- 第19章 快速傅里叶变换 217
 - 19.1 概述 217
 - 19.2 时分FFT 218
 - 19.3 流水线基2时分FFT处理器 221
 - 19.4 频分FFT 221
 - 19.5 流水线基2频分FFT处理器 224
- 第20章 求解线性方程组 225
 - 20.1 概述 225
 - 20.2 特别矩阵结构 225
 - 20.2.1 平面旋转(吉文斯)矩阵 226
 - 20.2.2 带状矩阵 226
 - 20.2.3 对角矩阵 227
 - 20.2.4 上三角矩阵 227
 - 20.2.5 下三角矩阵 227
 - 20.2.6 三对角矩阵 227
 - 20.2.7 上Hessenberg矩阵 227
 - 20.2.8 下Hessenberg矩阵 228
 - 20.3 前向替代(直接技术) 228
 - 20.3.1 前向替代依赖图 228
 - 20.3.2 前向替代规划方程和有向无环图(DAG) 229
 - 20.3.3 前向替代投影函数 230
 - 20.4 回代 230
 - 20.5 矩阵三角化算法 230
 - 20.5.1 Givens旋转算法 232
 - 20.5.2 矩阵三角化调度函数 233
 - 20.5.3 矩阵三角化投影方向 234
 - 20.6 连续超额松弛(SOR)(迭代法) 234
 - 20.6.1 SOR算法 235
 - 20.6.2 SOR算法调度算法 235
 - 20.6.3 SOR算法的投影方向 236
 - 20.7 习题 237
- 第21章 使用有限差分法求解偏微分方程 238
 - 21.1 概述 238
 - 21.2 1-D系统的FDM 239
 - 21.2.1 1-D FDM的调度函数 240
 - 21.2.2 投影方向 242
- 参考文献 243

章节摘录

版权页：插图：一对多（组播）一对多操作涉及一个发送端处理器和多个接收端处理器。图3.8（b）给出了一对多的通信模式。图中只显示了一个源到多个接收处理器的通信，但通常情况下，所有的处理器可以在同一时间执行一对多通信。接收处理器的数量取决于该算法的细节，以及如何完成任务到处理器的映射。此操作通常是在每次迭代中进行，因此必须做到高效率。大部分时间里，假设相邻的处理器之间的时钟同步已经完成，源寄存器和目的寄存器之间进行一个简单的数据交换。在其他情况下，双向方式（即数据一确认）甚至四向握手（即请求一确认一数据一确认）也是必要的。一对全部（广播）广播业务涉及在系统中发送相同的数据给所有处理器。图3.8（c）显示了处理器之间的广播通信模式。这种模式在提供数据给所有处理器时非常有用。它也可能意味着一个处理器作为发送端，其他处理器接收数据。我们将在脉动阵列和SIMD机上看到这种通信。收集收集操作包括从几个或全部处理器收集数据。图3.8（d）显示了处理器之间的收集通信模式。假设我们有P个处理器，所需收集数据的时间可以被计算为其中为传输一接收一处理一个数据项所需的时间。规约规约操作与收集操作类似，除了一些操作是针对收集到的数据。图3.8（d）显示了处理器之间的规约通信模式。一个规约操作的例子是当所有的处理器产生的所有数据相加以产生一个最终值。当需要规约的数据很多时，这项任务可能需要很长一段时间。假设我们有P个产生待加数据的处理器，总的时间预计为其中，是处理器处理一对已收到数据单元所需的时间。层次化地执行规约操作可能是值得的。在这种情况下，规约延迟时间为3.11.2消息传递（MP）通信机制 MP主要用于分布式内存机器。两个处理器之间传递消息过程涉及使用send（）和recv（）库函数。程序员使用send（destination, message）库函数来确定目的处理器或进程的ID以及要发送的数据。程序员还必须使用recv（source, message type）库函数指定源处理器或进程的ID和接收数据的类型。为了让两个处理器使用MP通信，需要两个操作：（1）在它们之间建立通信链路。链路建立依赖于互连网络的特性。我们可以考虑链路的物理性质（硬件）或它的逻辑性质（地址、单向或双向、能力、信息大小等）。（2）通过send（）和recv（）库函数交换消息。MPI是一个为了改善MP的使用和可移植性而开发的标准。MP同步确保处理器之间的正确通信。同步必须由程序员认真处理，因为执行send（）和RECV（）库函数是在操作系统或运行在处理器上的系统的控制下的。同步策略有两种类型：同步或阻塞，发送方在它执行了send（）库函数之后暂停执行，直到消息被接收。此外，接收方在执行RECV（）库函数后暂停，直到消息可用。异步或非阻塞，发送方执行send（）库函数后继续执行。此外，接收方执行RECV（）库函数后也继续执行。MPI标准支持单播和广播通信方式。

《算法与并行计算》

编辑推荐

《世界著名计算机教材精选:算法与并行计算》适合计算机工程、电气工程和计算机科学行业的研究人员和研究生。

《算法与并行计算》

精彩短评

1、敷衍了事拼凑的一本书，作者不认真，译着不认真，离“世界计算机经典图书”相差十万八千里，出版社太不负责任，这样平凡普通的一本书也加这么一个大帽，让我对这个系列的图书都有点畏忌了。我还买了另一本parallel programming，那本书作者的知识、内容组织、译者水平，都是超一流好，非常非常好！

《算法与并行计算》

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