1. 第一章
   1. 什么叫 数据并行和数据并行

任务必行是指将待解决问题所需要执行的各个任务分配到各个核上执行。

数据并行是指将待解决问题所需要处理的数据分配给各个核，每个核在分配到的数据集上执行大致相似的操作

* 1. 负载均衡 load balance

每个核分配大致相同数目的数据来计算

* 1. 并行、分布式、并发

并发：一个程序的多个任务在同一个时段内可以同时执行

分布式：一个程序需要与其他程序协作来解决某个问题

并行：一个程序通过多个任务紧密协作来解决某个问题

* 1. MPI、OpenMP和Pthreads的区别

MPI和Pthreads是C语言的扩展库，可以在C程序中使用扩展的类型定义、函数和宏；OpenMP包含了一个扩展库以及对C编译器的部分修改。

Pthreads和OpenMP是为共享内存系统的编程而设计的，它们提供访问共享内存的机制；MPI是为分布式内存系统的编程而设计的，它提供发送消息的机制。

OpenMP对C语言有更高层次的扩展。

Pthreads提供了一些在OpenMP中不可用的协调构造。OpenMP更容易将程序并行化，而Pthreads提供的一些构造使得它并行化其他一些程序的能力更强。

1. 第二章
   1. 冯诺依曼架构 哈佛结构

哈佛结构是一种将程序指令储存和数据储存分开的存储器结构。

冯·诺伊曼结构是一种将程序指令存储器和数据存储器合并在一起的电脑设计概念结构。

* 1. 进程和线程

线程间的切换比进程间的切换更快。线程相对于进程而言是“轻量级”的。线程包含在进程中，所以线程可以使用相同的可执行代码，共享相同的内存和相同的I/O设备。线程各自需要一个私有的程序计数器和函数调用栈

* 1. 一个进程为什么要有fork和join
  2. 为什么要用缓存 缓存有哪些替换策略

缓存的访问时间比其他存储区域的访问时间短。基于局部性原理，程序接下来可能会用到的指令和数据与最近访问过的指令和数据在物理上是邻近存放的。在访问完一个内存区域后，程序会在不久的将来（时间局部性）访问邻近的区域（空间局部性）

* 1. 什么叫 Cache命中和Cache缺失

当向Cache查询信息时，如果Cache中有信息，则称为Cache命中；如果信息不存在，则称为Cache缺失

* 1. 缓存的三个种类 直接映射 全相连 n路组相连

直接映射：每个高速缓存行在Cache中仅有唯一的位置

全相连：每个高速缓存行能够放置在Cache中的任意位置

n路组相连：每个高速缓存行都能放置到Cache中n个不同区域位置中的一个

* 1. LRU MRU

LRU：最近最少使用替换策略

MRU：最近最常使用替换策略

* 1. 什么叫虚拟内存

虚拟内存是计算机系统内存管理的一种技术。它使得应用程序认为它拥有连续的可用的内存（一个连续完整的地址空间），而实际上，它通常是被分隔成多个物理内存碎片，还有部分暂时存储在外部磁盘存储器上，在需要时进行数据交换。

在主存中存放当前执行程序所需要用到的部分，来利用时间和空间局部性；那些暂时用不到的部分存储在辅存的快中，称为交换空间

* 1. 为什么要有虚拟内存

电脑中所运行的程序均需经由内存执行，若执行的程序占用内存很大或很多，则会导致内存消耗殆尽。

* 1. 什么叫页表

页表是一种特殊的数据结构，存放着虚拟地址和物理地址的对应关系。

* 1. 什么叫页面失效（page fault）

想要访问的页不在内存中，即页表中该页没有合法的物理地址，该页只存储在磁盘上，那么这次访问称为页面失效。

* 1. 什么叫TLB

处理器有一种专门用于地址转换的缓存，来解决“使用页表会增加程序总体的运行时间”问题，它叫做转译后备缓冲区（Translation-Lookaside Buffer， TLB），即快表。

* 1. 什么叫SMT
  2. 硬件多线程

当前执行的任务被阻塞时，系统能够继续其他有用的工作。

* 1. Flynn分类 SISD SIMD MIMD 之间的区别

按照同时管理的指令流数目和数据流数目来对系统分类。

SISD：冯诺依曼系统，一次执行一条指令，一次存取一个数据项。SIMD：对多个数据执行相同的指令从而实现在多个数据流上的操作，向量处理器和GPU；MIMD：支持同时多个指令流在多个数据流上操作，通常是异步的，分为两种类型—共享内存系统和分布式内存系统。

* 1. 什么叫 共享内存系统和分布式内存系统

共享内存系统，一组自治的处理器通过互连网络与内存系统相互连接，每个处理器能够访问每个内存区域。处理器通过访问共享的数据结构来隐式地通信。

分布式内存系统，每个处理器有自己私有的内存空间，处理器-内存对之间通过互连网络相互通信。处理器之间通过发送消息或者使用特殊的函数来访问其他处理器的内存，这是一种显式的通信。

* 1. 网络连接
  2. 什么叫等分宽度 怎么求

衡量“同时通信的链路数目”或者“连接性”的一个标准。

① 将网络的节点尽可能地均分为两部分，在最坏情况下这两部分能发生的通信数就是等分宽度。

② 去除最少的链路数从而将节点尽可能地均分成两等分，去除的链路数就是等分宽度。

* 1. 什么叫 延迟和带宽

都是衡量互连网络性能的指标。

延迟是指从发送源开始传送数据到目的地开始接收数据之间的时间。

带宽是指目的地在开始接收数据后接收数据的速度。

* 1. 什么叫缓存一致性 监听缓存一致性协议和基于目录的缓存一致性协议

当一个处理器改变了Cache中某变量的副本时，其他处理器应该知道该变量已更新。P29

* 1. 什么叫伪共享 它和共享的区别

一个cache line可以被多个不同的线程所使用。如果有其他线程修改了cache line中某变量的值，线程1和线程2将会强制重新加载该cache line。因为一致性的保持是以cache line为基础的，而不是以单个独立的元素。这种明显没有必要的共享数据的方式被称作“False sharing”.

False sharing可以避免，但是true sharing不可避免

* 1. 什么叫 动态线程和静态线程

动态线程在有工作请求到达的时候会派生出一个工作线程来执行该请求。

静态线程则是直接派生出所有的线程，并在所有的线程合并到主线程后，主线程需要做一些清理工作。

* 1. 什么叫 非确定性

给定的输入能产生不同的输出。

* 1. 什么叫 线程安全性

如果一个代码块能够被多个线程同时执行而不引起问题，那么它是线程安全的。

* 1. 设计多线程和多进程的时候 输入和输出要考虑什么

P38

* 1. 什么叫 加速比和效率 物理含义是什么

P38

* 1. 什么叫 阿姆达尔定律

除非一个串行程序的执行几乎全部都并行化，否则，不论有多少可以利用的核，通过并行化所产生的加速比都会是受限的。

* 1. 什么叫 可扩展性 强可扩展性和弱可扩展性

增加一个程序的进程/线程数，如果在输入规模也以相应增长率增加的情况下，该程序的效率值一直都是E，那么这个程序就是可扩展的。

强可扩展，在增加进程/线程的个数时，可以维持固定的效率，却不增加问题的规模。弱可扩展，在增加进程/线程个数的同时，只有以相同倍率增加问题的规模才能使效率值保持不变。

1. 第三章
   1. 什么叫 通信子 为什么要有通信子

一组可以互相发送消息的进程的集合。

* 1. 什么叫 集合通信

涉及通信子中所有进程的通信函数称为集合通信。

* 1. MPI\_Send和MPI\_Recv有什么样的优点和缺点

3.1.11

* 1. MPI\_Send中type的作用

类型不能作为参数传递给函数

* 1. MPI\_Send和MPI\_Recv需要什么参数配对才能配对成功

recv\_comm = send\_comm, recv\_tag = send\_tag, dest = r, src = q

* 1. 会用MPI写程序 基本的程序要会写
  2. MPI\_Reduce的作用 MPI\_Allreduce MPI\_Bcast的作用 MPI\_Scatter MPI\_Gather
  3. 什么叫 MPI派生数据类型 为什么要用派生数据类型

可以表示内存中数据项的任意集合。

通信比本地计算开销大很多。

* 1. 3.6 3.11 还有3.8也做一做

1. 第四章
   1. 为什么要有join操作
   2. 什么叫临界区 为什么要有临界区

一个更新共享资源的代码段，一次只允许一个线程执行该代码段。

防止数据竞争。

* 1. 什么叫共享变量 优点和缺点是什么
  2. 哪些方法实现临界区

忙等待、互斥锁、信号量

* 1. 互斥锁和信号量的区别

① 互斥量用于线程的互斥，信号量用于线程的同步。这是互斥量和信号量的根本区别，也就是互斥和同步之间的区别。

② 互斥量值只能为0/1，信号量值可以为非负整数。也就是说，一个互斥量只能用于一个资源的互斥访问，它不能实现多个资源的多线程互斥问题。信号量可以实现多个同类资源的多线程互斥和同步。当信号量为二元信号量时，也可以完成一个资源的互斥访问。

③ 互斥量的加锁和解锁必须由同一线程分别对应使用，信号量可以由一个线程释放，另一个线程得到。

* 1. 信号量的操作 wait post

P116

* 1. barrier 忙等实现和信号量实现 条件变量
  2. 读写锁 为什么 优点 原理

P125

1. 第五章
   1. openmp和pthread相比 优缺点是什么

Pthreads提供了一些在OpenMP中不可用的协调构造。OpenMP更容易将程序并行化，而Pthreads提供的一些构造使得它并行化其他一些程序的能力更强。

* 1. 什么是pragma

一个预处理器指令，它指示使每个编译程序在保留C和C++语言的整体兼容性时提供不同机器和操作系统特定的功能。

* 1. 在openmp里面 变量的可视范围

5.3

* 1. openmp为什么要有归约子句 归约子句的基本写法 怎么实现的 可以避免什么问题

相同的归约操作符重复地应用到操作数序列来得到一个结果的计算。

5.4 P148~149

* 1. 循环调度的类型

5.7

* 1. openmp中实现临界区 critical、atomic、lock

5.8.10