并行计算

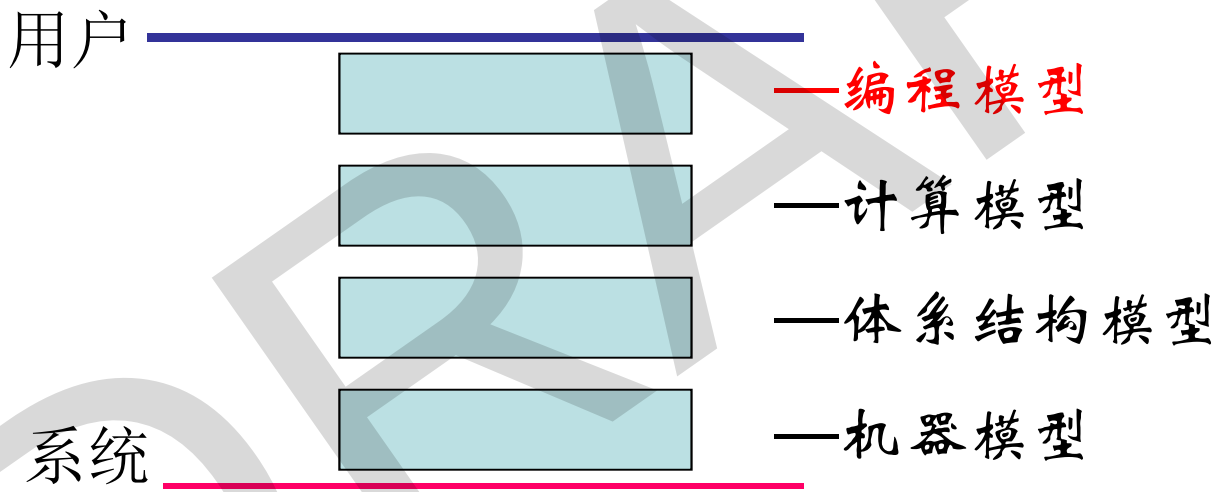
第四讲 并行编程模型与环境

何克晶
kejinghe@gmail.com
华南理工大学
计算机科学与工程学院

華南理工大學
South China University of Technology

并行编程模型

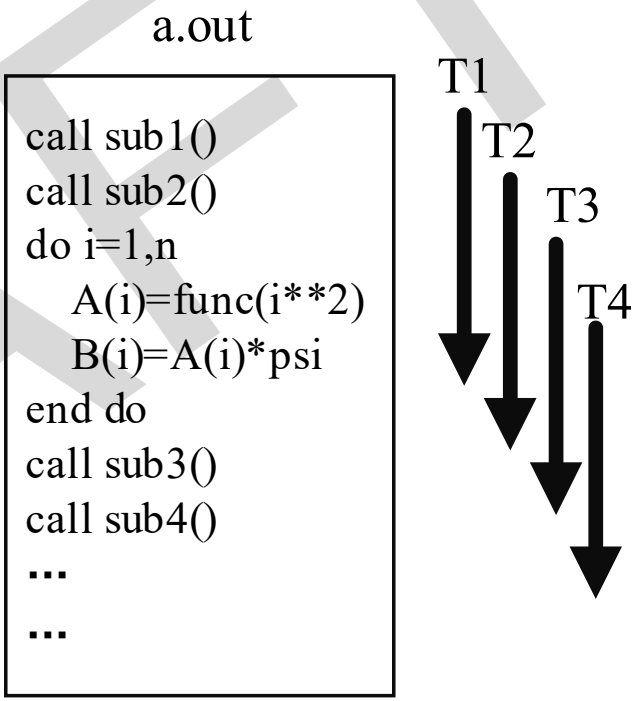
- 并行编程模型是并行计算机体系架构的一种抽象，它便于编程人员用程序对算法进行实现



目录

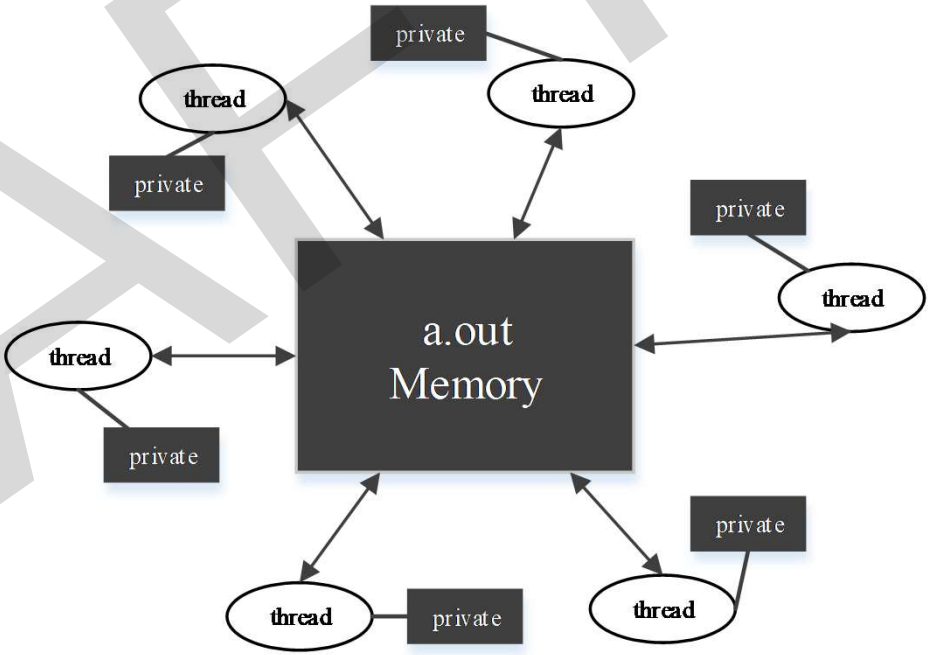
线程模型

- 进程是程序执行时的一个实例，它是程序已经执行到何种程度的数据结构的汇集。从内核的观点看，进程的目的在于担当分配系统资源（CPU、内存等）的基本单位。线程是进程的一个执行流，是CPU调度和分派的基本单位，它是比进程更小的能独立运行的基本单位



线程模型

- 共享：线程与同属一个进程的其他线程共享进程所拥有的全部资源
- 轻量级：和多进程并行编程相比，多线程并行编程是一种“轻量级”的多任务操作方式
- 启动一个线程花费的空间远远小于一个进程所花费的空间，而且线程间彼此切换所需的时间也远远小于进程间切换所需要的时间

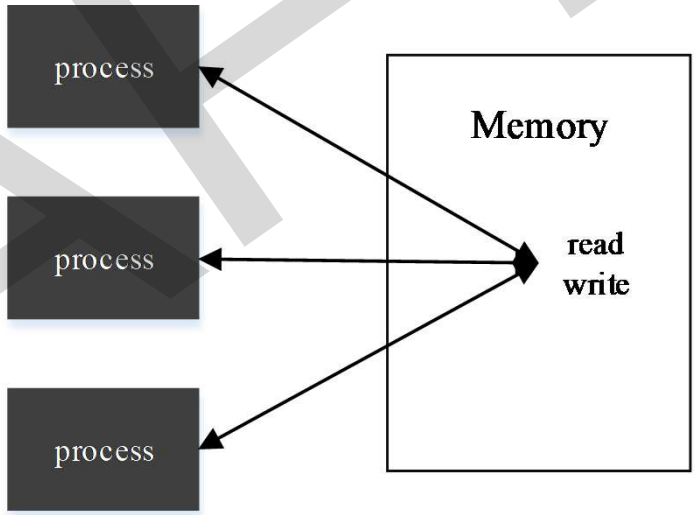


线程模型

- 多线程并行编程作为一种多任务、并发的工作方式，相比于多进程并行编程还具备以下三个优势：
  - （1）加速应用程序响应
  - （2）使多CPU系统更加高效
  - （3）改善程序结构
- 在软件和硬件上实现多线程
  - POSIX线程就是典型的在软件层面实现的多线程
  - Intel的超线程（Hyper-threading）技术则在硬件层面上实现多线程，超线程技术是通过当一个线程在停止或者等待I/O状态时切换到另外一个线程上实现
  - OpenMP也是基于线程的并行编程模型，OpenMP采用fork-join并行编程模型，程序开始于一个单独的主线程，主线程会一直串行的执行，直至进入并行域，主线程创建一系列并行的线程执行并行域的代码。当所有并行线程完成代码的执行后，它们或被同步或被中断，最后只剩下主线程执行

共享内存模型

- 所有处理器/任务都共享同一个内存空间，对共享资源的进行异步读写。各种机制（例如锁/信号量）用于控制对共享内存的访问、解决争用和防止竞争及死锁
- 在共享内存机器上，操作系统、编译器和硬件等共同支持共享内存模型。例如POSIX标准为使用共享内存提供了API，而UNIX提供了共享内存段（例如shmget、shmat、shmctl等）。在分布式内存机器上，共享的内存分布在网络的不同机器上，但是通过特定的硬件和软件实现了全局共享



共享变量通信模型实现

a.虚地址-物理地址的映射

*并行机具有全局物理地址空间--UMA

硬件通过虚地址-物理地址的映射实现；

*并行机具有多个独立的局部物理地址空间--NUMA

硬件实现—硬件通过虚地址-物理地址映射实现；

软件实现—产生缺页事件，由OS介入实现映射。

b.通信与复制

第一次访问或产生访问扑空时发生通信，将数据副本复制到本地，否则只是数据副本重用；

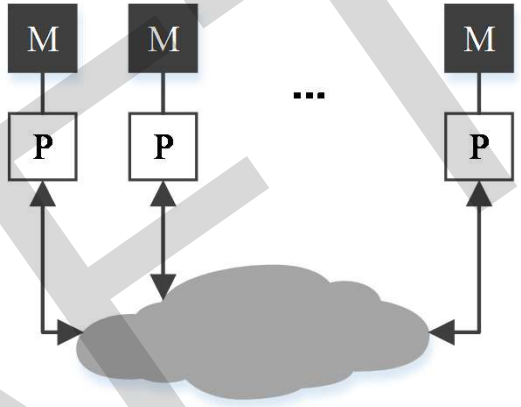
共享数据不需要重新命名(与本地数据使用方法相同)；

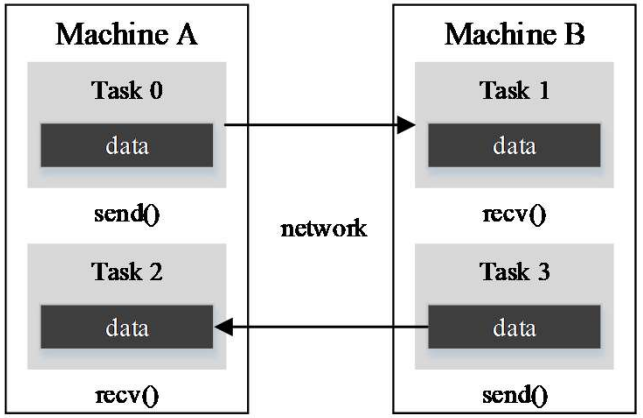
必须保持共享数据的一致性(硬件或软件)；

通信粒度较细。

消息传递模型

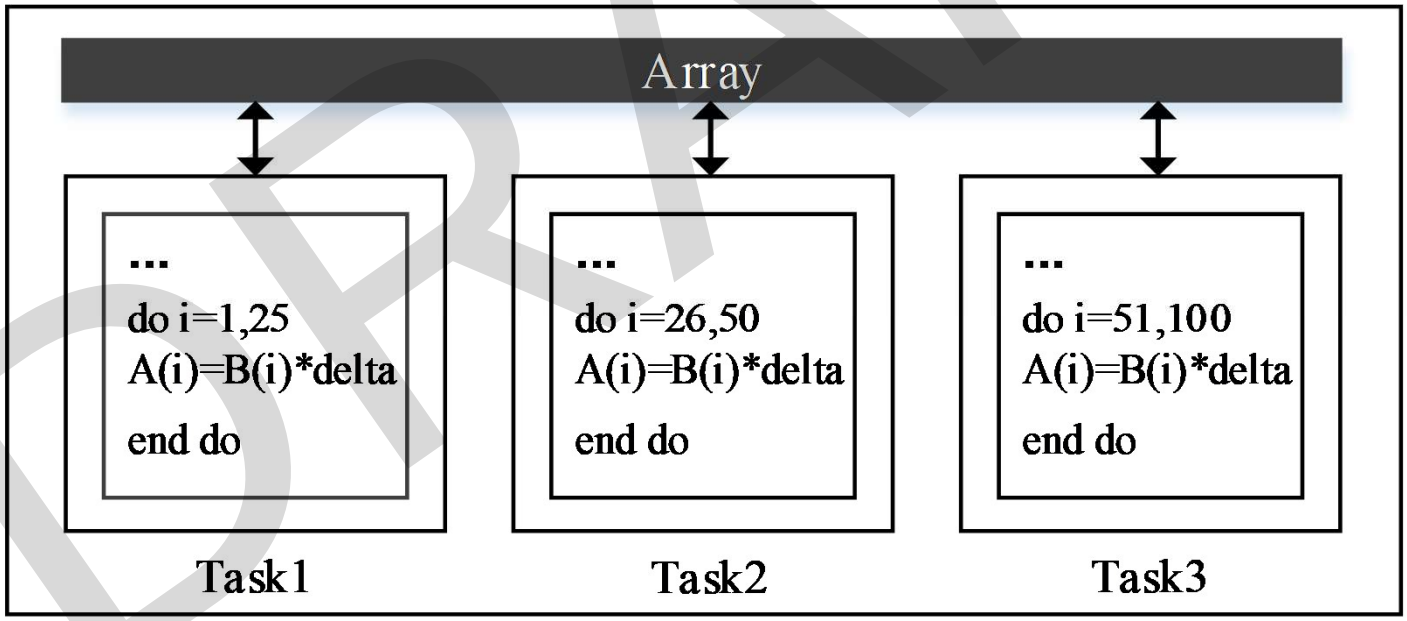
- 各个并行执行的部分之间通过传递消息来交换信息、协调步伐、控制执行
- 一般是面向分布式内存的，但是它也可适用于共享内存的并行机
- 将各个并行执行部分之间复杂的信息交换和协调、控制的任务交给了开发者，这在一定程度上增加了开发者的负担



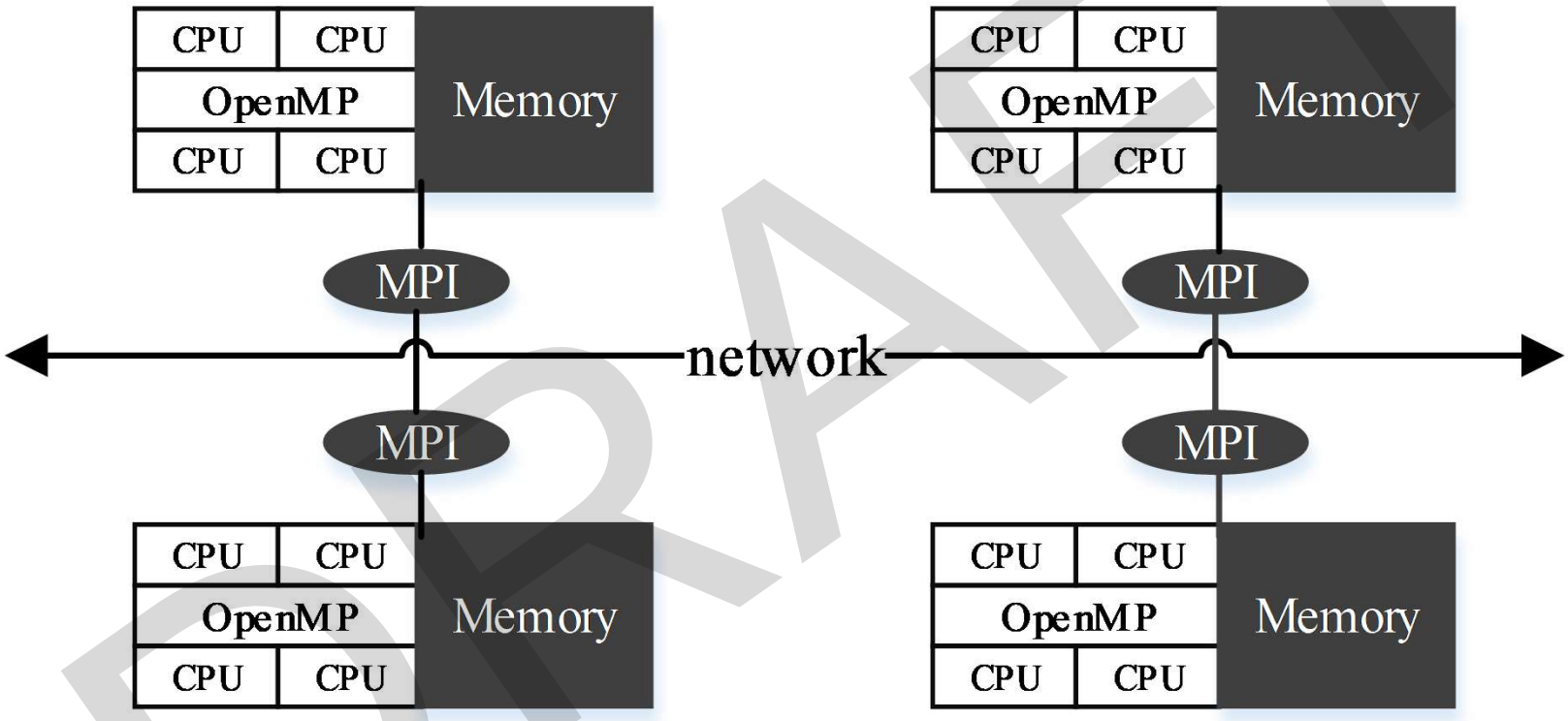


数据并行模型

- 一般包含多个任务，多个任务需要操作同一个数据结构，但每一个任务操作的是数据的不同部分
- 在共享内存架构中，所有任务都通过共享内存来访问数据；在分布式内存架构中则会将数据分割并且保存到每个任务的局部内存中

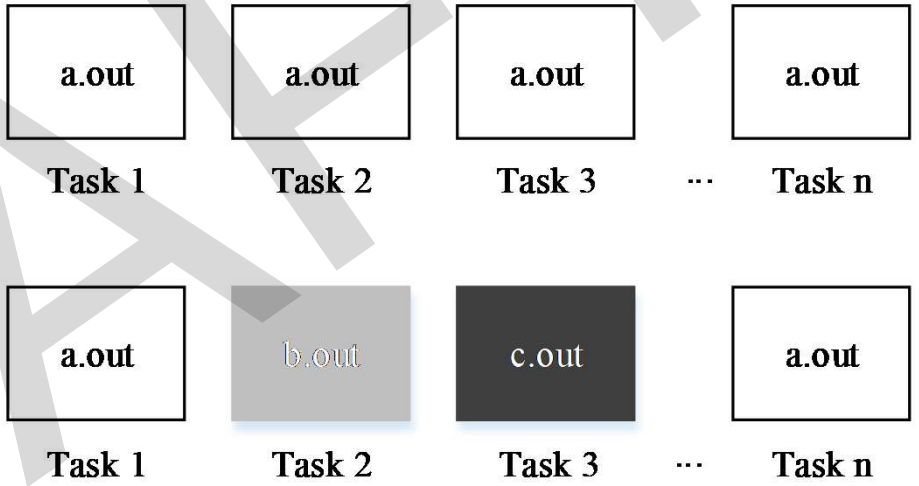


混合模型


CPU
CPU
OpenMP
CPU
CPU
Memory
MPI
network
MPI
CPU
CPU
OpenMP
CPU
CPU
Memory
MPI
MPI
CPU
CPU
OpenMP
CPU
CPU
Memory
CPU
CPU
OpenMP
CPU
CPU
Memory

SPMD和MPMD

- SPMD（Single Program Multiple Data，单程序多数据模型）
  - 系统中各处理器均运行相同的程序，但对不同的数据执行操作
- MPMD（Multiple Program Multiple Data，多程序多数据模型）
  - 每个处理器执行不同的代码副本，各自对数据完成不同的运算



目录

并行编程语言

方法	优点	缺点	例子
库函数	容易实现，不需要使用新的编译器	没有编译检查、分析和优化	MPI
新语言	可以针对特定并行机的硬件结构设计专门的并行语言	难以实现，需要编写新的编译器。因为通常是给特定并行机器设计的，所以存在兼容性差的问题	Julia 、Limbo
编译制导	是库函数和新语言两者优缺点的折中，其显著特点是通过“普通串行程序+格式注释”的形式，就可以在任何串行平台上编译		OpenMP