

# 计算机系统结构

## 第八章 多处理器

曹 强

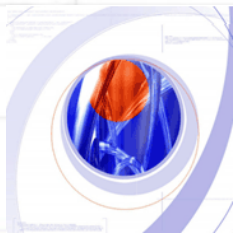
武汉光电国家研究中心

华中科技大学计算机科学与技术学院



# 第8章 多处理器

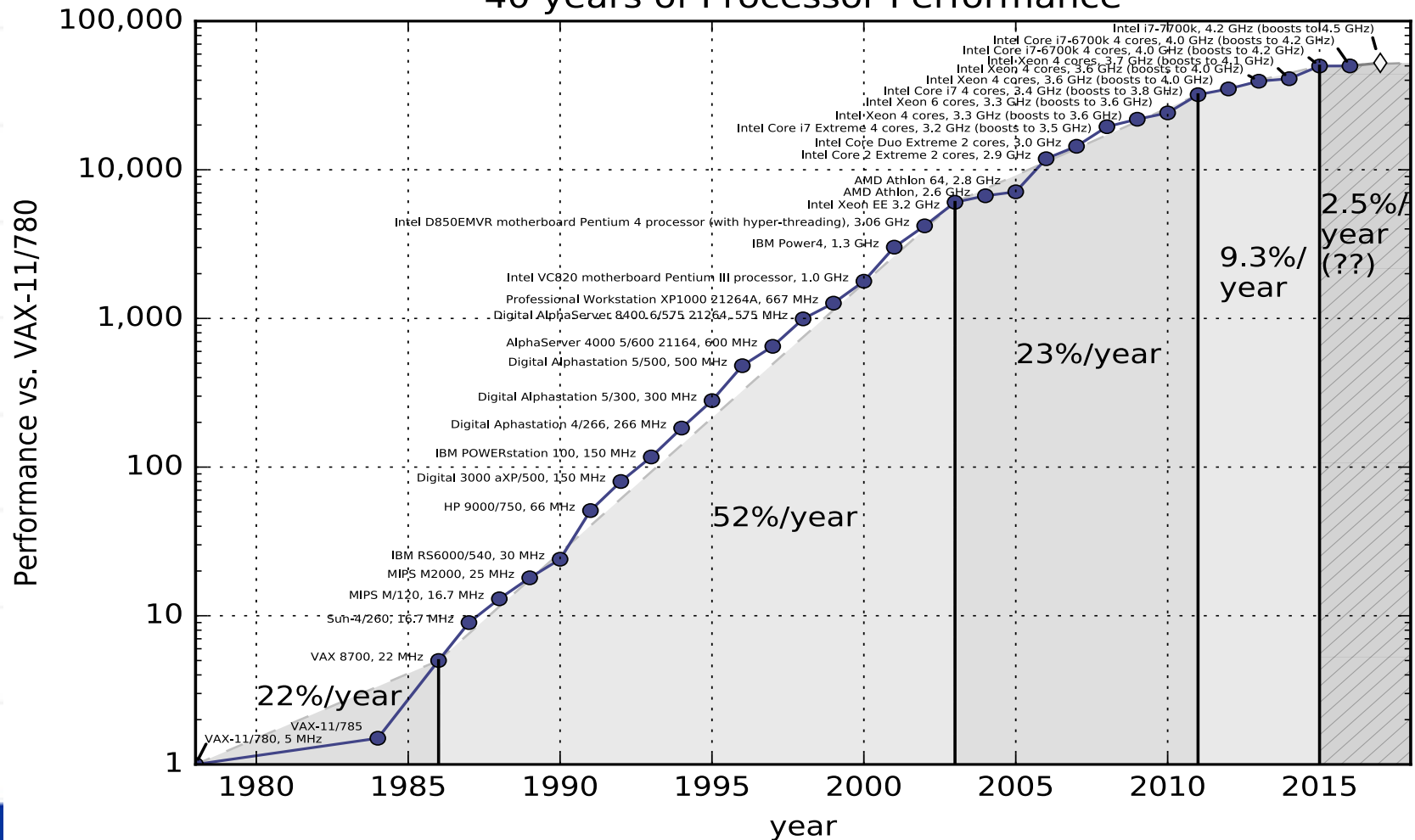
- 8.1 多处理器概念
- 8.2 对称式共享存储器系统结构
- 8.3 同步
- 8.4 同步性能问题



## 8.1 引言

## 1. 单处理器系统结构性性能提升的边际效应不断降低

40 years of Processor Performance



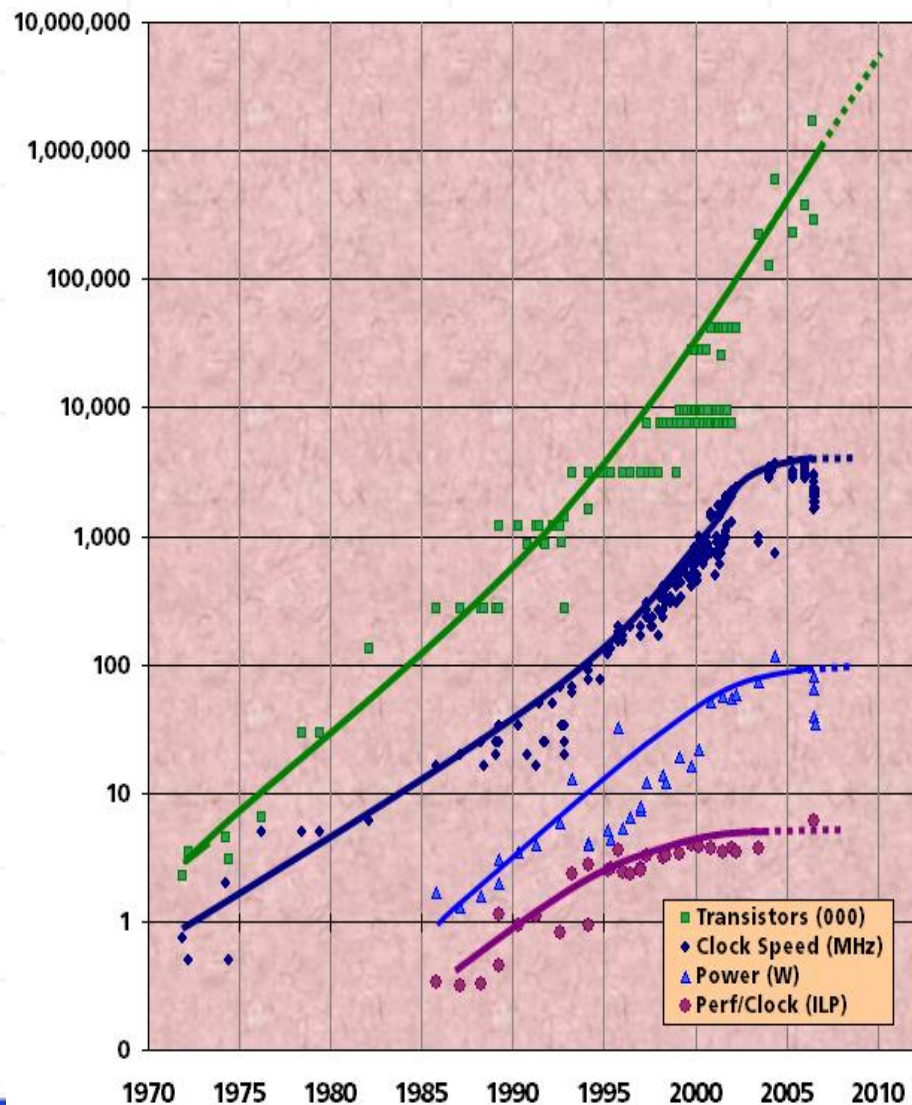
## 8.1 引言

### 2.通过增加处理器核数提升整体计算能力

- (1) 摩尔定律推动晶体管数量每2年增加2倍左右
- (2) 时钟频率几乎不变
- (3) 处理器功率几乎不变
- (4) 指令并行度几乎不变

本章重点：中小规模多处理器  
(处理器或核数 < 32)

(多处理机设计的主流)



### 3. 并行计算机系统结构的分类

#### 1. Flynn分类法

SISD、SIMD、MISD、MIMD

#### 2. MIMD已成为通用多处理机系统结构的选择，原因：

- MIMD具有灵活性； □
- MIMD可以充分利用商品化微处理器在性能价格比方面的优势。

计算机机群系统（cluster）是一类广泛被采用的MIMD机器。

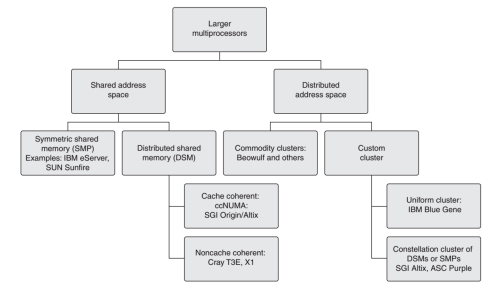
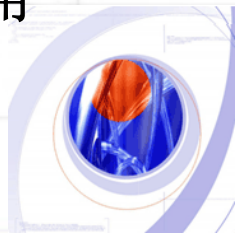
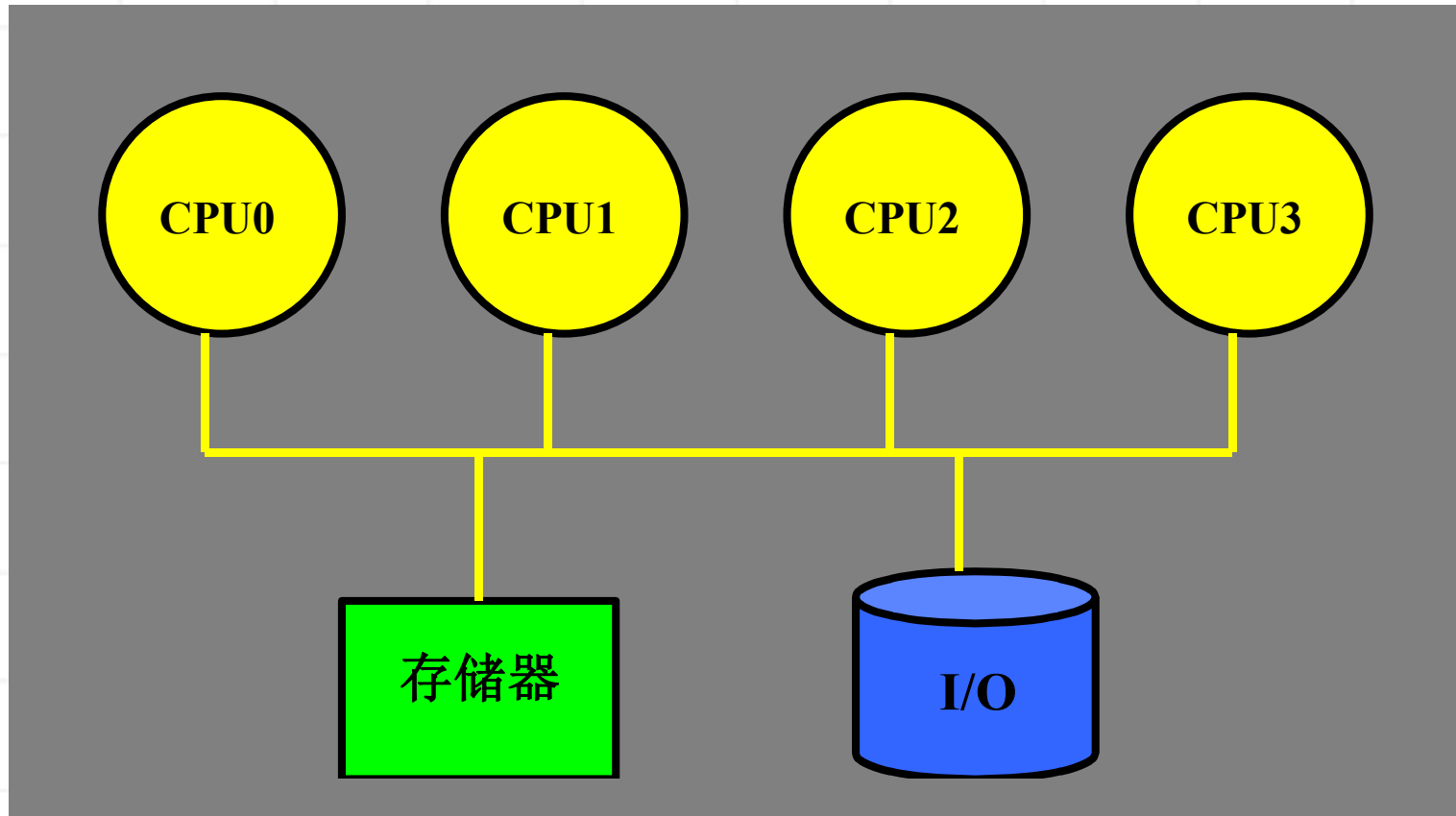
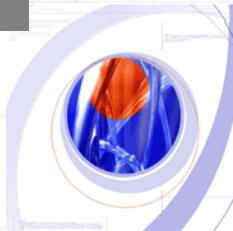


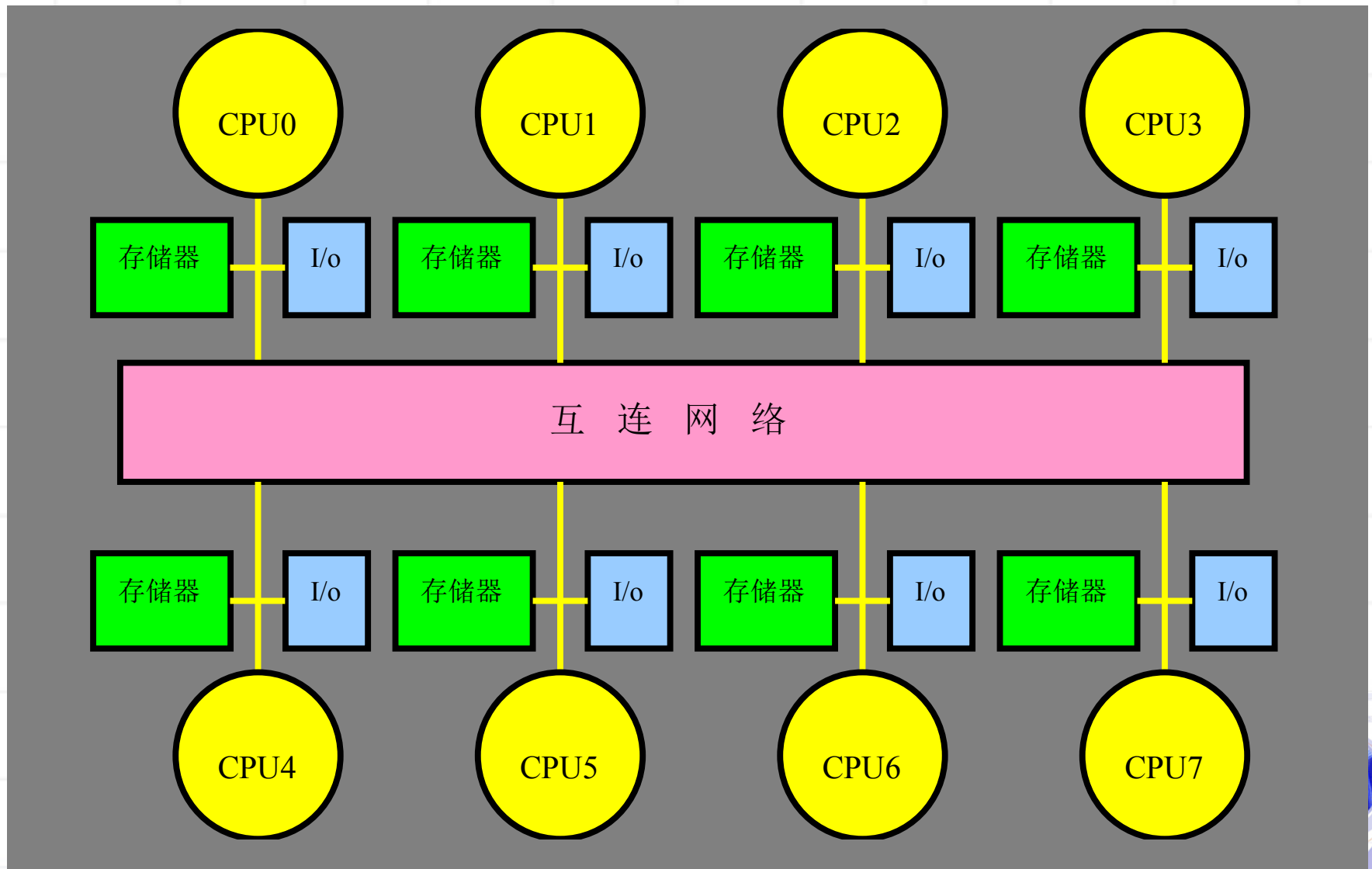
Figure 1.21 The space of large-scale multiprocessors and the relation of different classes.

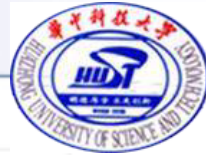




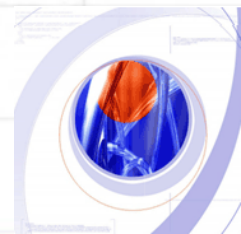
对称式共享存储器多处理机的基本结构



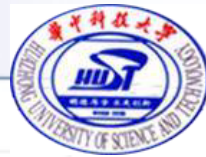




- 将存储器分布到各结点有两个**优点**
  - 如果大多数的访问是针对本结点的局部存储器，则可降低对存储器和互连网络的带宽要求；
  - 对本地存储器的访问延迟时间小。
- 最主要的**缺点**
  - 处理器之间的通信较为复杂，且各处理器之间访问延迟较大。







## 8.1.2 存储器系统结构和通信机制

### 1. 两种存储器系统结构和通信机制

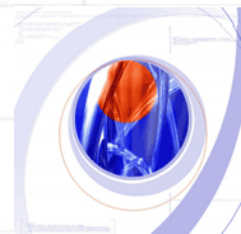
#### ➤ 共享全局地址空间

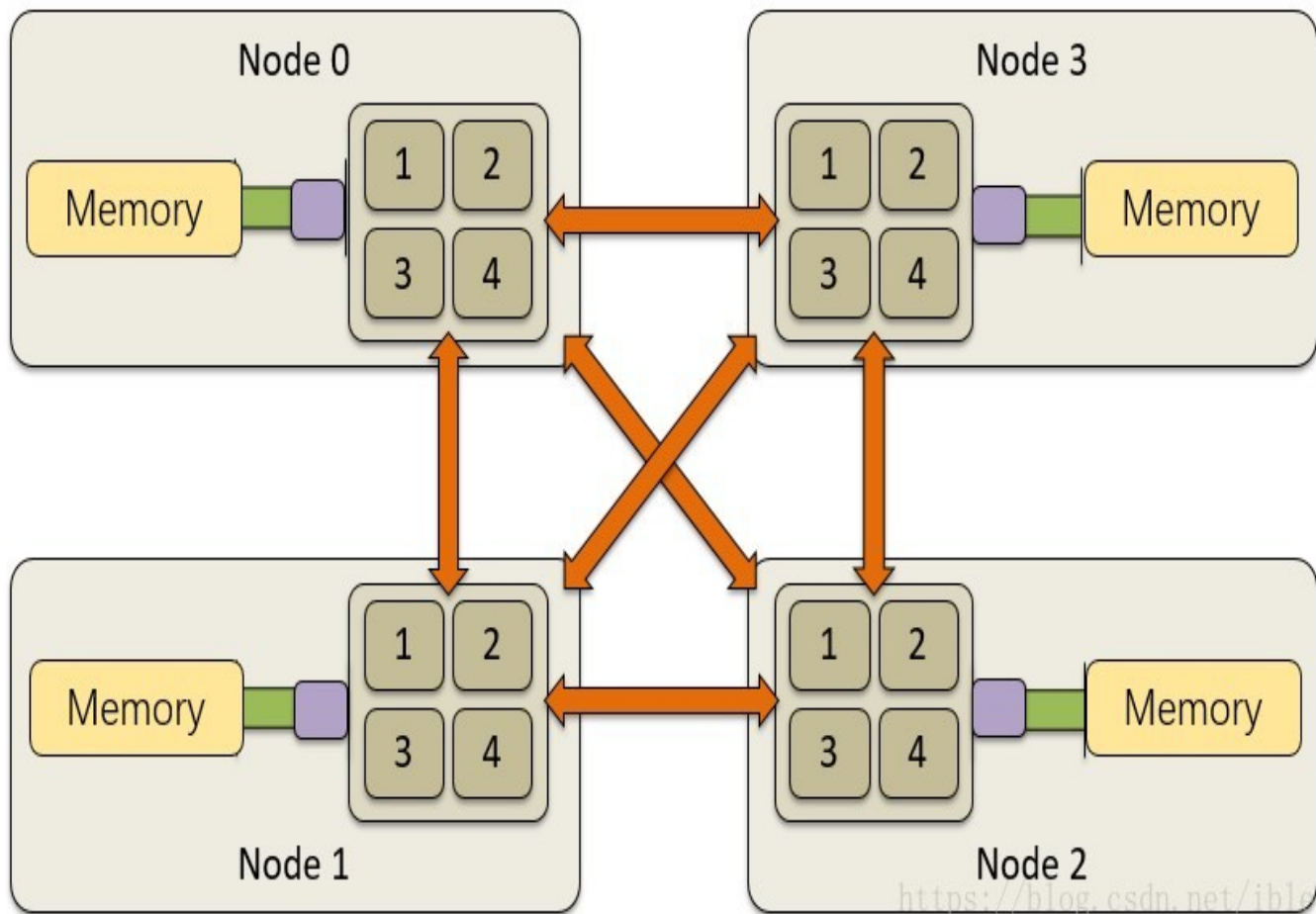
- 物理共享的存储器具有统一逻辑地址
- 物理上分离的所有存储器作为一个统一的共享逻辑空间进行编址。分布式共享存储器系统

(DSM: Distributed Shared-Memory)

#### ➤ 分布式的独立地址空间

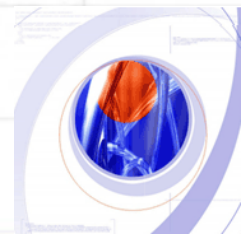
- 整个系统的地址空间由多个独立逻辑地址空间构成
- 不同结点中的地址空间之间是相互独立的
- 每个结点中的存储器只能由本地的处理器进行访问，远程的处理器不能直接对其进行访问。





4路多核处理服务器结构

NUMA机器 (NUMA: Non-Uniform Memory Access)



共享  
存储器  
通信  
机制

共享全局地址空间

UMA

NUMA

对称式共享存  
储多处理器

分布式共享内  
存多处理器

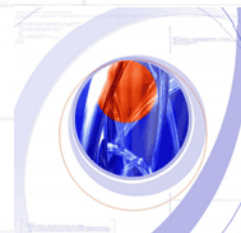
独立地址空间

分布式存储器  
多处理器

消息传递通信  
机制

对称式共享存  
储器多处理器

分布式存储器  
多处理器





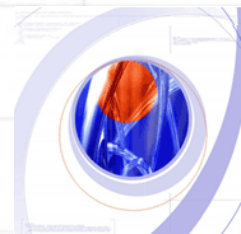
## 2. 通信机制

### ➤ 共享存储器通信机制

- 共享地址空间的计算机系统采用
- 处理器之间是通过用load和store指令对相同存储器地址进行读/写操作来实现的。

### ➤ 消息传递通信机制

- 多个独立地址空间的计算机采用
- 通过处理器间显式地传递消息来完成
- 消息传递多处理机中，处理器之间是通过发送消息来进行通信的，这些消息请求进行某些操作或者传送数据。





**例如：**一个处理器要对远程存储器上的数据进行访问或操作：

- 发送消息，请求传递数据或对数据进行操作；

**远程进程调用** (RPC, Remote Process Call)

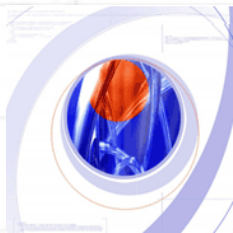
- 目的处理器接收到消息以后，执行相应的操作或代替远程处理器进行访问，并发送一个应答消息将结果返回。

#### □ **同步消息传递**

请求处理器发送一个消息后一直要等到应答结果才继续运行。

#### □ **异步消息传递**

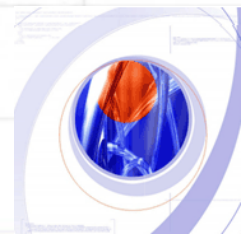
当请求处理器发送一个消息后可以处理其他事情，数据发送方在得到所需数据后，通知请求处理器。



### 3. 不同通信机制的优点

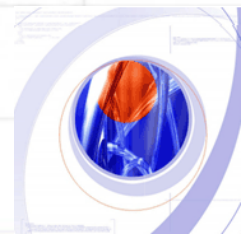
#### ➤ 共享存储器通信的主要优点

- 与常用的对称式多处理机使用的通信机制兼容。
- 易于编程，同时在简化编译器设计方面也占有优势。
- 采用大家所熟悉的共享存储器模型开发应用程序，而把重点放到解决对性能影响较大的数据访问上。
- 当通信数据量较小时，通信开销较低，带宽利用较好。
- 可以通过采用Cache技术来减少远程通信的频度，减少了通信延迟以及对共享数据的访问冲突。



## ➤ 消息传递通信机制的主要优点

- 硬件较简单。
- 通信是显式的，因此更容易搞清楚何时发生通信以及通信开销是多少。
- 显式通信可以让编程者重点注意并行计算的主要通信开销，使之有可能开发出结构更好、性能更高的并行程序。
- 同步很自然地与发送消息相关联，能减少不当的同步带来错误的可能性。

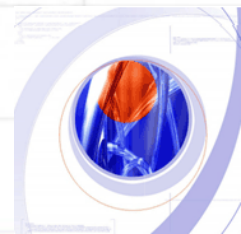






➤ 可在支持上面任何一种通信机制的硬件模型上建立所需的通信模式平台。

- 在共享存储器上支持消息传递相对简单。
- 在消息传递的硬件上支持共享存储器就困难得多。  
所有对共享存储器的访问均要求操作系统提供地址转换和存储保护功能，即将存储器访问转换为消息的发送和接收。



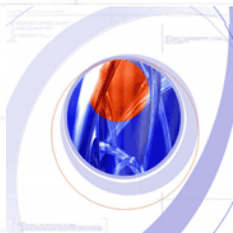


### 8.1.3 并行处理面临的挑战

并行处理面临着两个重要的挑战

- 程序中的并行性有限
- 相对较大的通信开销

$$\text{系统加速比} = \frac{1}{(1 - \text{可加速部分比例}) + \frac{\text{可加速部分比例}}{\text{理论加速比}}}$$





## 1. 第一个挑战

有限的并行性使计算机要达到很高的加速比十分困难。

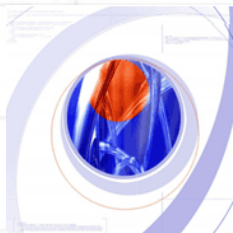
**例8.1** 假设有100个处理器达到80的加速比，求原计算程序中串行部分最多可占多大的比例？

**解** Amdahl定律为：

$$\text{加速比} = \frac{1}{\frac{\text{可加速部分比例}}{\text{理论加速比}} + (1 - \text{可加速部分比例})}$$

$$80 = \frac{1}{\frac{\text{并行比例}}{100} + (1 - \text{并行比例})}$$

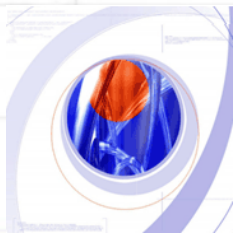
由上式可得：并行比例 = 0.9975



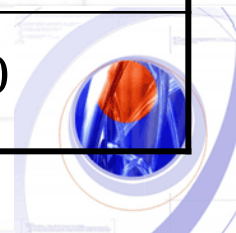


## 2. 第二个挑战：多处理机中远程访问的延迟较大

- 在现有的机器中，处理器之间的数据通信大约需要50~1000个时钟周期。
- 主要取决于：
  - 通信机制、互连网络的种类和机器的规模
- 在几种不同的共享存储器并行计算机中远程访问一个字的典型延迟

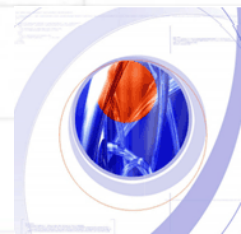


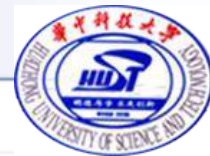
机器	通信机制	互连网络	处理机最大数量	典型远程存储器访问时间 (ns)
Sun Starfire servers	SMP	多总线	64	500
SGI Origin 3000	NUMA	胖超立方体	512	500
Cray T3E	NUMA	3维环网	2048	300
HP V series	SMP	8×8交叉开关	32	1000
HP AlphaServer GS	SMP	开关总线	32	400





例8.2 假设有一台32台处理器的多处理机，对远程存储器访问时间为200ns。除了通信以外，假设所有其它访问均命中局部存储器。当发出一个远程请求时，本处理器挂起。处理器的时钟频率为2GHz，如果指令基本的CPI为0.5（设所有访存均命中Cache），求在没有远程访问的情况下和有0.2%的指令需要远程访问的情况下，前者比后者快多少？





解 有0.2%远程访问的机器的实际CPI为:

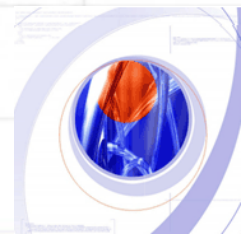
$$\begin{aligned}\text{CPI} &= \text{基本CPI} + \text{远程访问率} \times \text{远程访问开销} \\ &= 0.5 + 0.2\% \times \text{远程访问开销}\end{aligned}$$

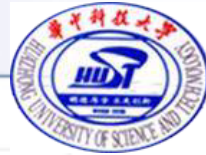
远程访问开销为:

$$\text{远程访问时间/时钟周期时间} = 200\text{ns}/0.5\text{ns} = 400\text{个时钟周期}$$

$$\therefore \text{CPI} = 0.5 + 0.2\% \times 400 = 1.3$$

因此在没有远程访问的情况下的机器速度是有0.2%远程访问的机器速度的 $1.3/0.5=2.6$ 倍。





### ➤ 问题的解决

- 并行性不足：采用并行性更好的算法
- 远程访问延迟的降低：靠系统结构支持和编程技术

3. 在并行处理中，影响性能（负载平衡、同步和存储器访问延迟等）的关键因素常依赖于：

### 应用程序的高层特性

如数据的分配，并行算法的结构以及在空间和时间上对数据的访问模式等。

### ➤ 依据应用特点可把多机工作负载大致分成两类：

- 单个程序在多处理机上的并行工作负载
- 多个程序在多处理机上的并行工作负载

