



中山大學

SUN YAT-SEN UNIVERSITY

计算机体系结构

主讲教师：胡淼

中山大学计算机学院

2024 年 秋季

教学内容

- 第 1 章 绪论
- 第 2 章 基准评测集
- **第 3 章 高性能计算机的体系结构**
- 第 4 章 高性能处理器的并行计算技术
- 第 5 章 高性能计算机的存储层次
- 第 6 章 高性能计算机的互连网络
- 第 7 章 异构计算体系结构
- 第 8 章 领域专用体系结构

第 3 章 高性能计算机的体系结构

教学目的

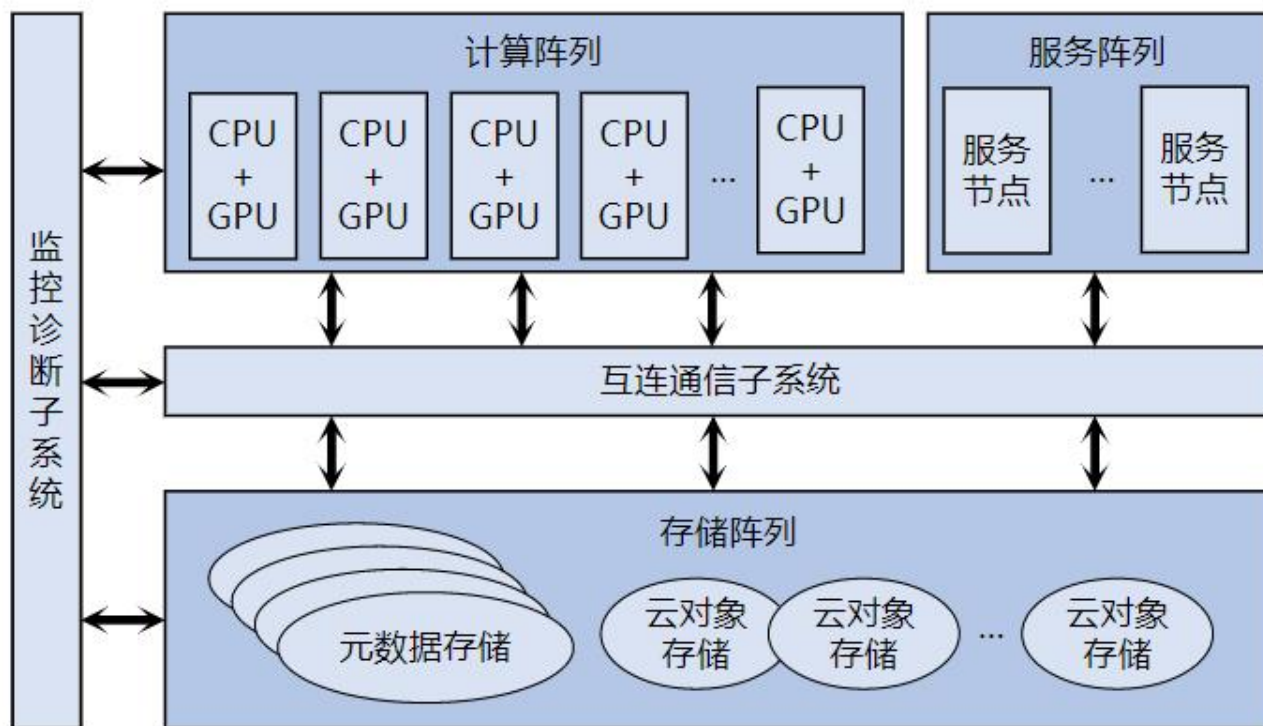
- 掌握：Flynn分类法；
- 熟悉：超级计算机硬件组成；
- 了解：非冯诺依曼体系结构。

目录

- (一) 超级计算机硬件
- (二) 高性能计算机体系结构分类
- (三) 非冯·诺依曼体系结构

(一) 超级计算机硬件

超级计算机硬件



超级计算机硬件

- 从硬件组成上看，超级计算机系统的硬件由五个子系统组成：
 - 计算阵列
 - 存储阵列
 - 服务阵列
 - 互连通信子系统
 - 监控诊断子系统

计算阵列

- 天河一号超级计算机由 **7168** 个计算节点组成
 - 每个计算机节点配置有两个Intel CPU 和一个 NVIDIA GPU
- 天河二号拥有约 **17920** 个计算节点
 - 每节点配备两颗 Xeon E5 系列 12 核心的中央处理器、三个至强融核 (Xeon Phi) 57 核心的协处理器 (运算加速卡)

计算阵列

- 加速器常被用于增加计算的吞吐量，但是这种提升是以降低可编程性微代价而获得
 - 加速器使用的控制逻辑通常与现有的处理器指令集架构 (ISA) 不兼容

存储阵列

- 存储阵列通常采用**层次式混合共享存储架构**，实现大容量、高带宽、低延迟的共享存储功能
 - 天河一号超级计算机的 I/O 存储系统使用 Luster 文件系统，拥有 6 个 I/O 管理节点和 128 个 I/O 存储节点。系统的总内存为 **262 TB**，磁盘容量为 **2 PB**
 - 天河二号超级计算机每个节点拥有 64GB 主存，而每个 Xeon Phi 协处理器板载 8GB 内存，故每节点共 88GB 内存，整体总计内存 **1.408 PB**，外存包含 **12.4 PB** 容量的硬盘阵列

服务阵列

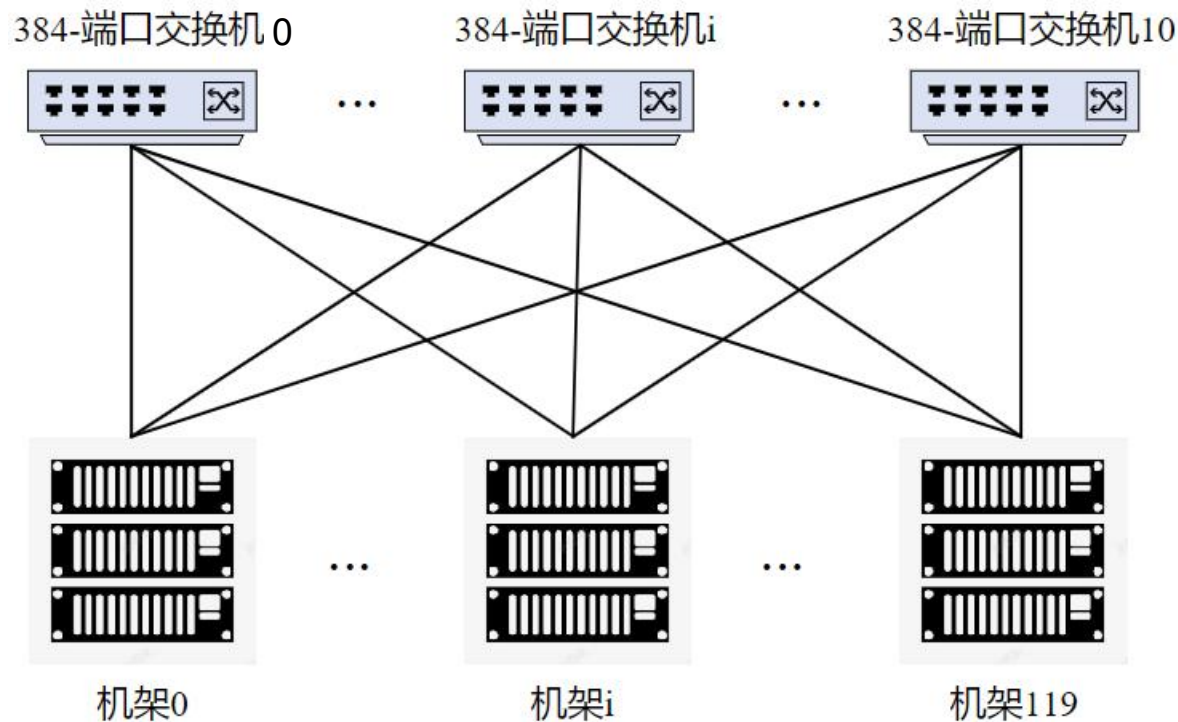
- 服务阵列采用了商用服务器，属于大容量胖节点
 - 天河一号系统共有 140 个机架，包括 112 个计算机架、8 个服务机架、6 个通信机架和 14 个 I/O 机架
 - 天河二号的共有 125 个机柜，每个机柜容纳 4 个机架，共计 500 个机架

互连通信子系统

- 互连网络是大规模并行处理系统的核心
 - 天河系列超级计算机互连通信子系统使用了自主定制的高速互连网络，拓扑结构使用了**胖树拓扑**结构，可高效地进行均衡扩展
 - 互连网络支持基于硬件实现的**自动消息交换**机制的集体操作，如多播和广播集体

互连通信子系统

第二层包含11个384端口交换机



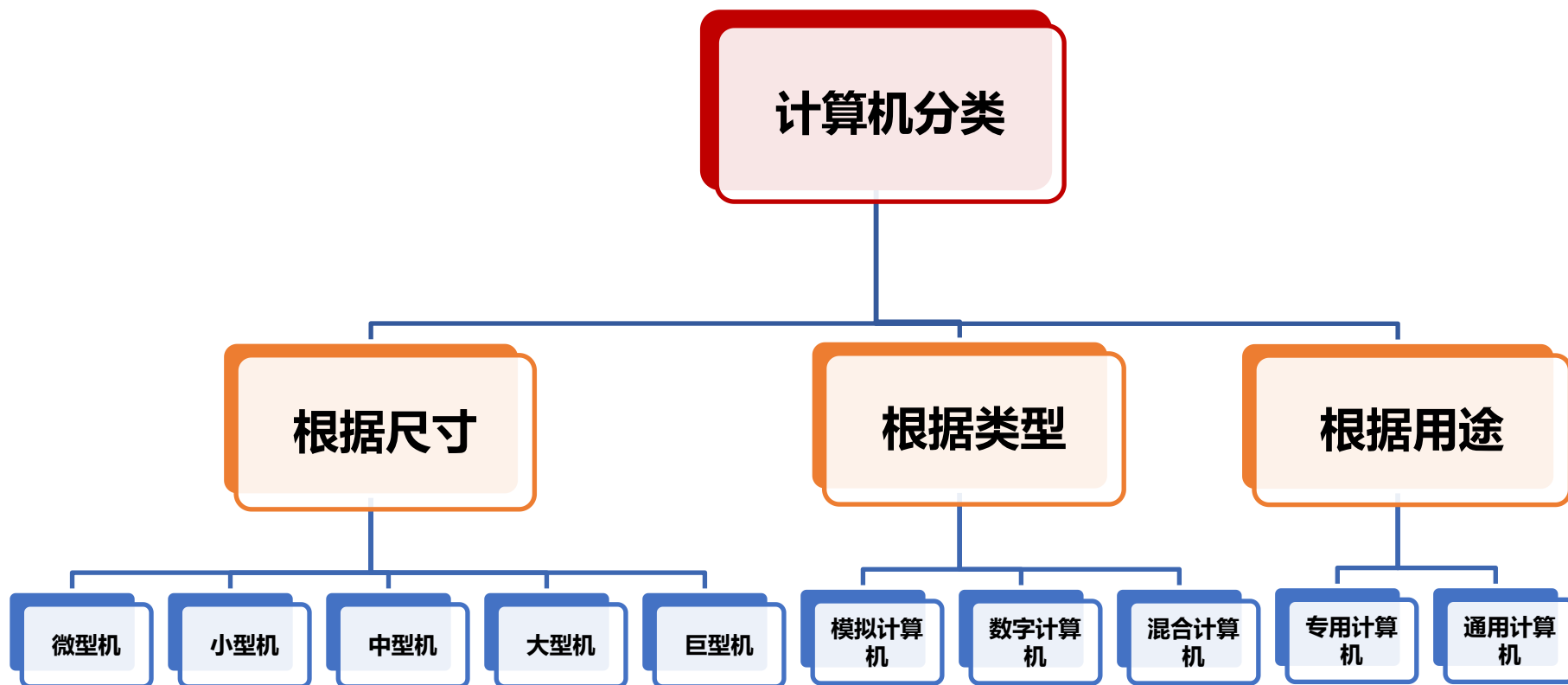
Xue-Jun Yang, Xiang-Ke Liao, Kai Lu, The TianHe-1A SupercomPEter: Its Hardware and Software [J], 2011, 26(3): 344-351.

监控诊断子系统

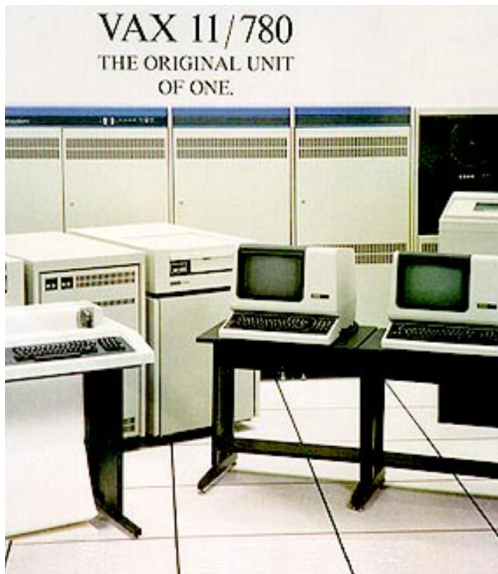
- 监控诊断子系统实现了整体系统**实时安全监测**和**诊断调试**功能
 - 实时监控、控制、诊断和调试整个超级计算机系统的健康状态、功耗与温度等信息

(二) 高性能计算机体系结构分类

传统计算机分类方法



传统计算机分类方法



VAX 11(小型机)



IBM System/3(中型机)



IBM Z9(大型机)

Classes of Computers

- **Internet of Things/Embedded Computers**
- **Personal Mobile Device (PMD)**
- **Desktop Computing**
- **Servers**
- **Clusters / Warehouse Scale Computers**



IoT/Embedded Computers

- **Internet of Things (IoT)** refers to **embedded computers** that are connected to the Internet, typically wirelessly
- **Embedded computers** have the **widest spread** of processing **power** and **cost**
 - Include 8-bit to 32-bit processors that may cost **one penny**, and high-end 64-bit processors for cars and network switches that cost **\$100**
- **Primary goal** is to **meet the performance need at a minimum price**
 - rather than achieving more performance at a higher price

Examples of Embedded Computers



Personal Mobile Device

- **Personal mobile device (PMD)** is a collection of wireless devices with multimedia user interfaces
 - such as cell phones, tablet computers, and so on
 - **Cost** is a prime concern. Applications on PMDs are often web-based and media-oriented
- **PMD vs. Embedded computers**
 - PMDs are platforms that can run externally developed software
 - Use the ability to run third-party software as dividing line between non-embedded and embedded computers
- **Other features:**
 - real-time, energy efficiency, memory usage minimization



Desktop Computing

- **Desktop computing** spans from **low-end netbooks** to high-end, heavily configured **workstations**
 - Desktop computing sales are **declining**
- **Emphasis: optimize price-performance**
 - Performance: measured primarily in terms of **compute performance** and **graphics performance**
- **The newest, highest-performance microprocessors and cost-reduced microprocessors often appear first in desktop systems**



Servers

- **Role:** to provide **larger-scale** and more **reliable file and computing services**
 - the backbone of large-scale enterprise computing
- **Main Characteristics:**
 - **Availability:** E.g., ATM machines or airline reservation systems
 - **Scalability:** Response to an increasing demand for services
 - **Throughput:** E.g., transactions per minute



Clusters/Warehouse-Scale Computers

- Clusters are **collections of desktop computers or servers connected by local area networks to act as a single larger computer**
 - Each node runs its own operating system, and nodes communicate using a networking protocol
- WSCs are the **largest** of the clusters
 - designed so that tens of thousands of servers can act as one
 - **Price-performance** and **power** are critical to WSCs
- Supercomputers vs. WSC
 - supercomputers emphasize **floating-point performance**
 - WSCs emphasize **interactive applications, large-scale storage, dependability, and high Internet bandwidth**



Classes of Computers

- **Five mainstream computing classes and their system characteristics**

Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Internet of things/ embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of microprocessor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

传统分类方法的缺陷

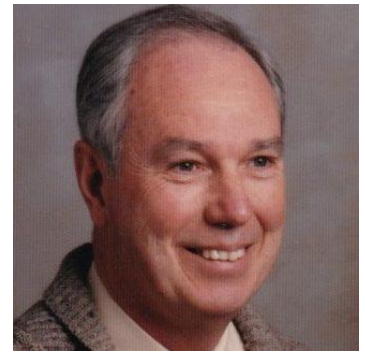
- **存在的缺陷：**

- 由于计算机硬件技术的发展，划分标准需随时间变化
- 这样分类不能反映机器的系统结构特征

- **Flynn分类法：**

- 为了解决原有分类方法的缺点，1972年Michael J. Flynn提出了一种基于数据流和指令流的并行性关系的分类方法

Michael J. Flynn



- **弗林 (Michael J. Flynn): 斯坦福大学教授**
 - 计算机体系结构领域的开创者，并以提出Flynn分类法而闻名世界
 - 1960年，他在普渡大学获得博士学位，加入IBM
 - 1975年，他加入斯坦福大学电气工程系并组建了斯坦福体系结构和架构小组，在此期间，他**创立**了两个体系结构的专业组织：IEEE计算机体系结构技术委员会和ACM SIGARCH
 - 1999年他从斯坦福大学退休并继续担任名誉教授
 - Flynn教授因对计算机和数字系统体系结构的技术贡献而于1992年获得ACM / IEEE Eckert-Mauchley奖
 - ACM Fellow、IEEE Fellow

Flynn分类法的相关概念

- **指令流 (Instruction):**

- 指机器执行的指令序列，即一系列将数据送入数据处理单元进行修改的命令

- **数据流 (Data):**

- 指由指令流调用的数据序列，包括输入数据和中间结果，但不包括输出数据

- **多倍性 (Multiple):**

- 指在系统性能瓶颈部件上处于同一执行阶段的指令或数据的最大可能个数

Flynn分类法

- 面向计算机体系结构的一种分类方法
 - 基于指令和数据来划分

	Single Instruction	Multiple Instruction
Single Data	SISD	MISD
Multiple Data	SIMD	MIMD

- 其中“I”和“D”分别代表指令流和数据流
- 每个类别又可以根据多倍性分为两类，一类用“S”表示单数据流，另一类用“M”表示多数据流

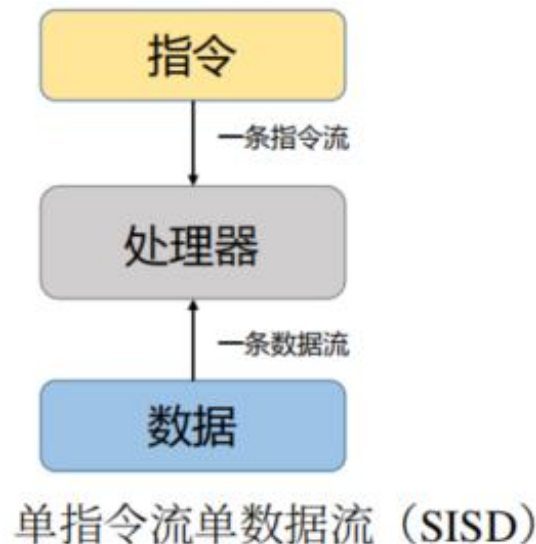
Flynn分类法

名称	指令流	数据流	举例
SISD	1个	1个	传统冯诺依曼计算机
SIMD	1个	多个	向量计算机，阵列计算机
MISD	多个	1个	很少用
MIMD	多个	多个	多处理机，多计算机系统

- 另外，随着的图形处理器(GPU)近年来的不断发展，一种基于SIMD的新的架构SIMT出现了，全称是：SIMT-单指令流多线程流（Single Instruction stream, Multiple Threads stream）

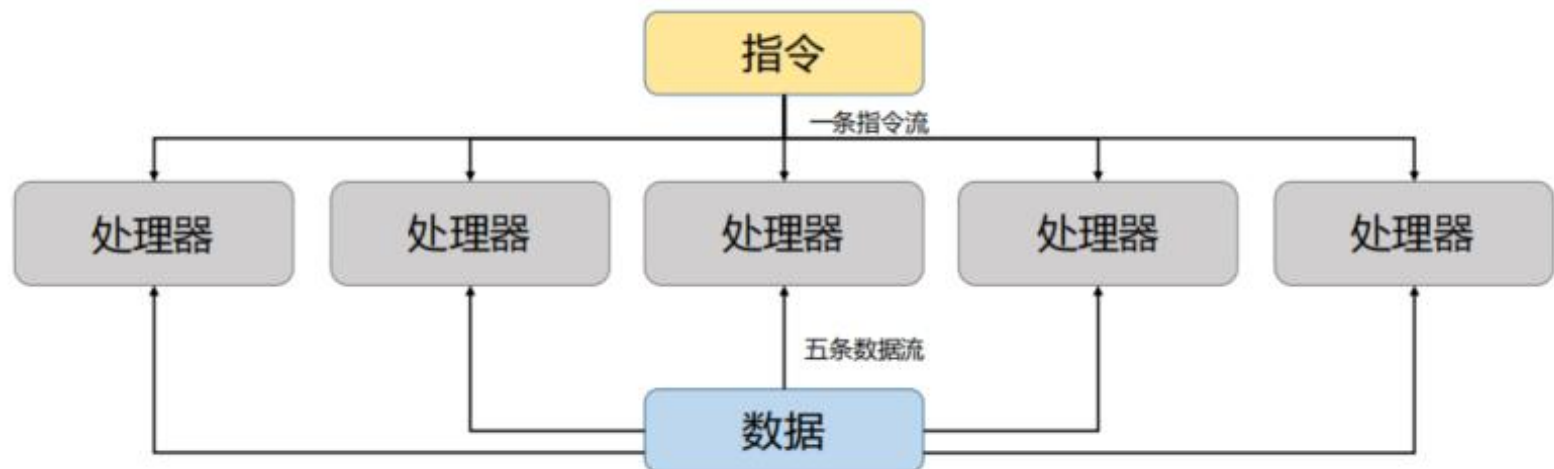
SISD体系结构

- SISD计算机是一种传统的串行计算机
 - 它从硬件上就不支持任何并行化计算，**所有的指令都是串行执行**
 - 在一个时钟周期内只能执行一条数据流，因此这种机器被称为单指令单数据流机器
 - 早期计算机多为 SISD 计算机，如早期的 IBM PC 机、Intel 8086/8088 微处理器、早期的巨型机等

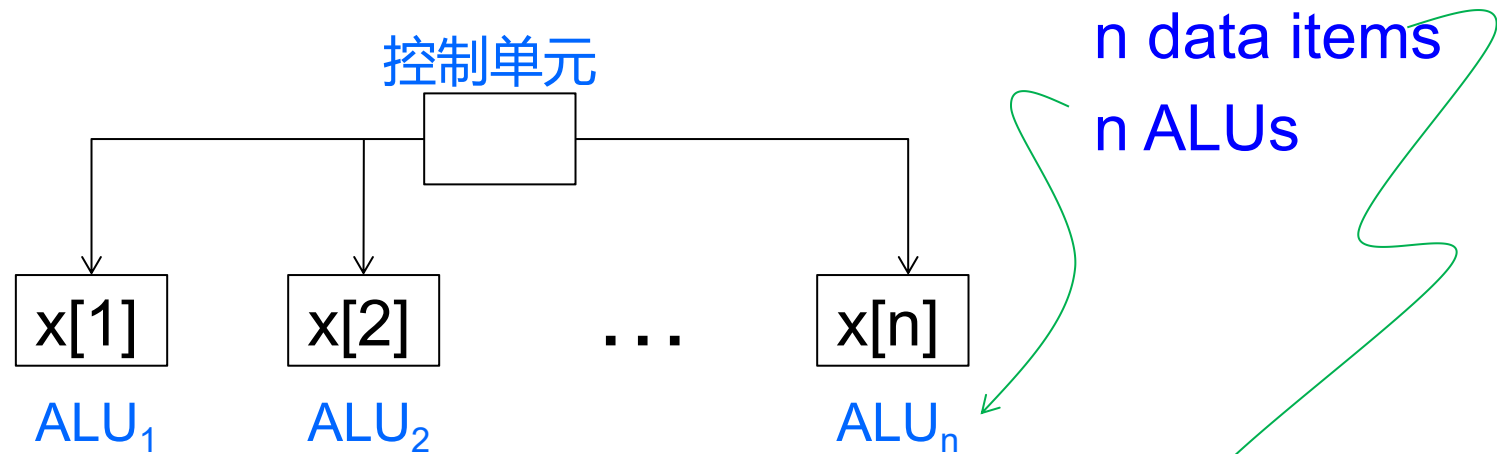


SIMD体系结构

- SIMD计算机可以实现 **数据级并行**，对多个不同的数据流并行执行相同的数据处理操作
 - 该类型计算机主要适用于解决使用向量和矩阵等复杂科学计算和大规模工程计算的问题
 - 这类机器大多应用于数字信号处理、图像处理等领域，SIMD根据类型又可以分为 **阵列计算机** 和 **向量计算机**



SIMD示例



```
for (i = 0; i < n; i++)  
    x[i] += y[i];
```

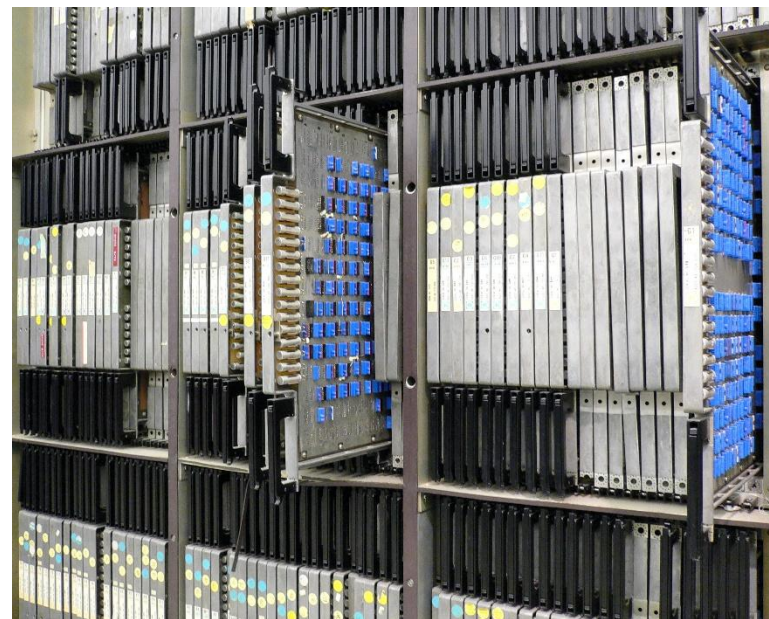
SIMD

- **问题:**
 - 如果没有和数据项一样多的ALU，如何处理？
- **解决方法:**
 - 把任务进行切分，分批处理

Round3	ALU ₁	ALU ₂	ALU ₃	ALU ₄
1	X[0]	X[1]	X[2]	X[3]
2	X[4]	X[5]	X[6]	X[7]
3	X[8]	X[9]	X[10]	X[11]
4	X[12]	X[13]	X[14]	

SIMD体系结构 – 阵列计算机

- 阵列计算机的基本思想是用一个**单一的控制单元提供信号驱动多个处理单元同时运行**
 - 每个处理器单元都由功能增强版的计算单元和本地内存组成
 - 所有的处理器单元都由相同的控制单元发出的指令流控制
 - 每个处理单元可以选择**执行**或者**不执行**控制器发出的指令流，处理单元之间通过互连网络连接
- **目前阵列计算机的发展没有取得较大的成功，发展前景并不明朗**



美国ILLIAC-IV阵列计算机，
于1960年代建造完成，1981年停用

ILLIAC IV的处理单元阵列结构

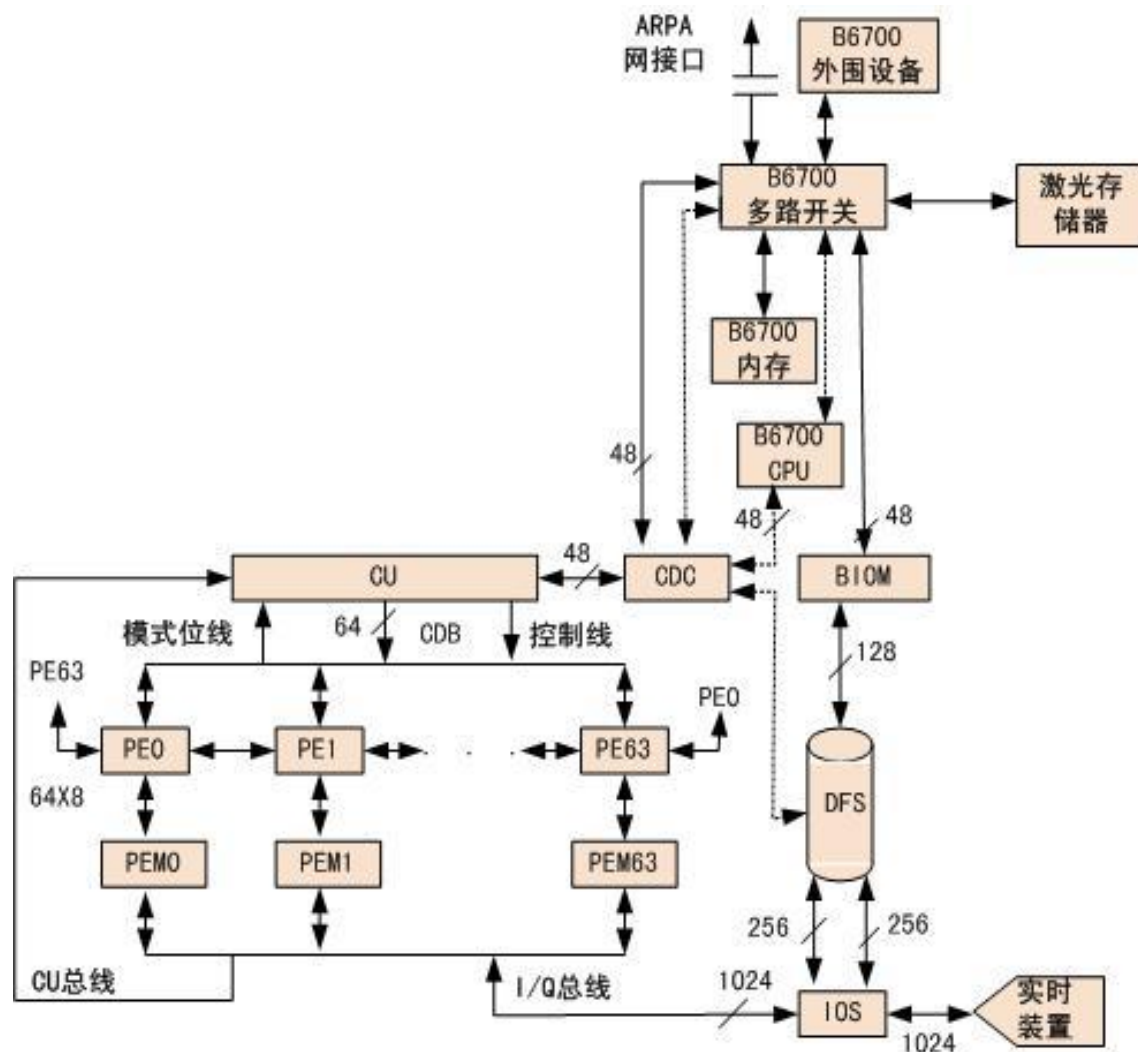
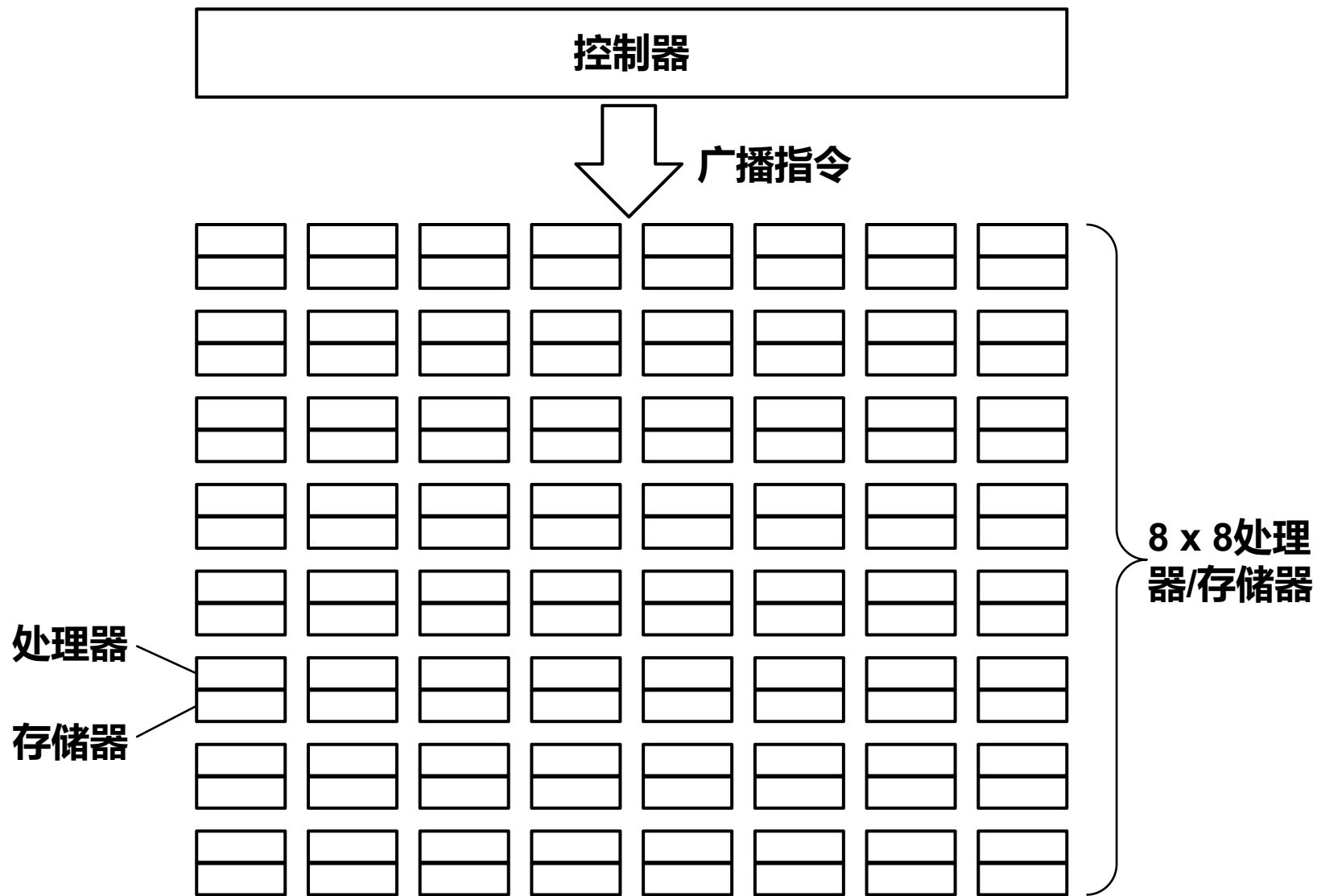


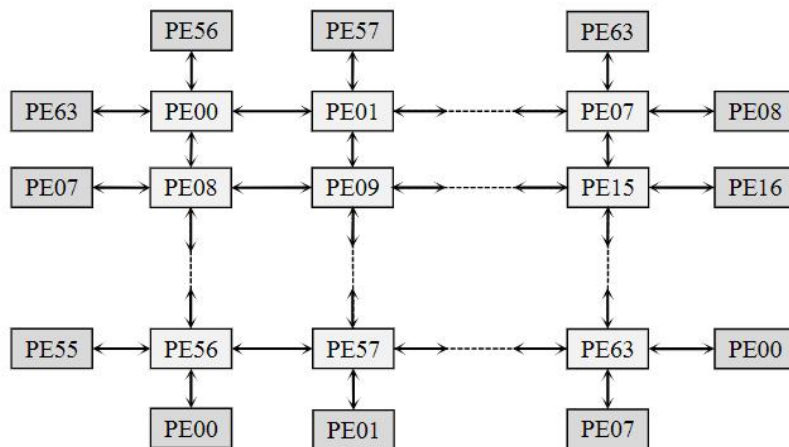
图8.5 ILLIAC IV系统总框图



美国 ILLIAC-IV 阵列计算机基本结构

ILLIAC IV的处理单元阵列结构

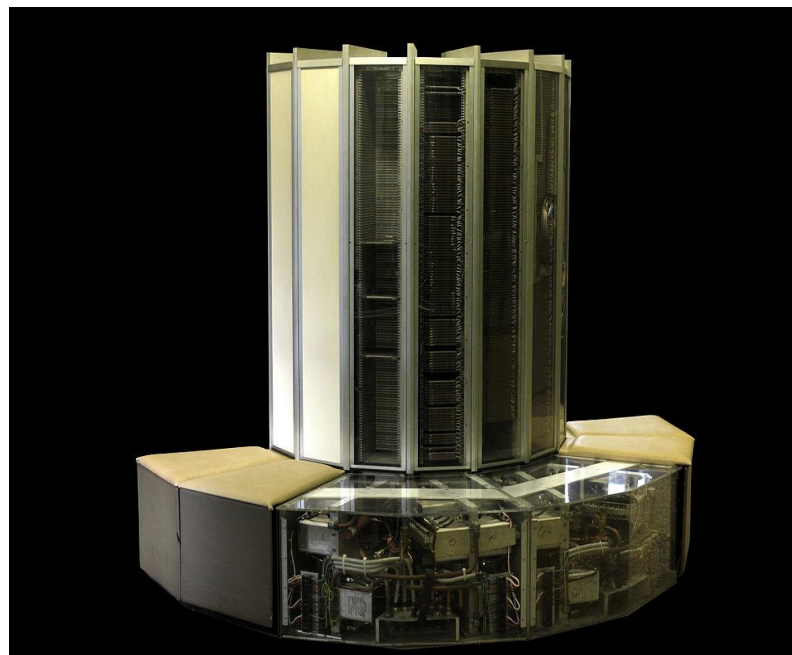
- ILLIAC IV 阵列由64个处理单元、64个处理单元存储器和存储器逻辑部件所组成
 - 64个处理部件PE0 ~ PE63排列成8×8的方阵，任何一个PE_i只与其上、下、左、右4个近邻PE_{i-8}(mod64)、PE_{i+8}(mod64)、PE_{i-1}(mod64)和PE_{i+1}(mod64)直接相连
 - 循此规则，南北方向上同一列的PE两端相连成一个环，东西方向上每一行的东端PE与下一行的西端PE相连，最下面一行的东端PE则与最上面一行的西端PE相连，从而构成一个闭合的螺线形状，所以称其为 **闭合螺线阵列**
 - 要将PE63的信息传送到PE10，最快可经过路径：
 - PE63→PE7→PE8→PE9→PE10
 - 要将PE9的信息传送到PE45，最快可经过路径：
 - PE9→PE1→PE57→PE56→PE48→PE47→PE46→PE45



ILLIAC-IV阵列计算机处理单元互连图

SIMD体系结构 – 向量计算机

- 向量计算机是指专门对**向量**进行处理的计算机
 - 它主要是以**流水线结构**为主，以向量作为基本操作单元，操作数和结果都以向量的形式存在
 - 向量的处理方法有多种不同的，主要包括横向，纵向和纵横处理方法
 - 衍生出了很多不同种类的向量计算机



第一台向量计算机 Cray-1，1975年推出，此后该型号计算机在科学计算领域占据了数十年的统治地位

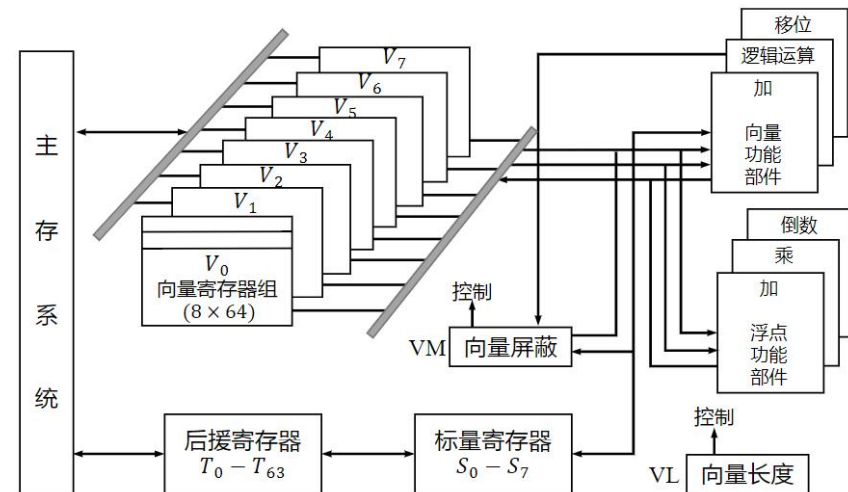
Cray-1向量计算机

- **向量寄存器组V**

- 由512个64位的寄存器组成，分成8块
- 标号：V0 ~ V7
- 每一个块称为一个向量寄存器，可存放一个长度（即元素个数）不超过64的向量
- 每个向量寄存器可以每拍向功能部件提供一个数据元素，或者每拍接收一个从功能部件来的结果元素

- **标量寄存器S和后援寄存器T**

- 标量寄存器有8个：S0 ~ S7 64位
- 标量保存寄存器（Scalar-save Registers）用于在标量寄存器和存储器之间提供缓冲，标号分别为T0 ~ T63



Cray-1 向量计算机的基本结构

SIMD体系结构 – 向量计算机



我国自主研发的第一台大型向量计算机“757”机于1983年交付使用，该研究成果获得了1985年国家科技进步一等奖

- “上世纪60年代初期，科学院计算所承担了军用、民用、通用、专用、各式各样的计算机任务。任务是五花八门，但是特点都是相同的：**没有合同，没有经费**。我们从没有考虑有没有经费、报酬、日后算不算成果这些事情，一心想着认真完成上级交给的任务……”

--- 高庆狮院士



向量处理器 vs. 传统 CPU

- **向量处理器**

- 能够对数组或者数据向量进行操作

- **传统CPU**

- 对单独的数据元素或者标量进行操作

向量处理器的主要部件

- **向量寄存器**

- 能存储由多个操作数组成的向量
- 长度：4-128 个 64 位元素不等

- **向量化和流水化的功能部件**

- 对向量中的每个元素执行同样的操作

- **向量指令**

- 对向量进行操作的指令，而非标量

向量处理器的主要部件

- **交叉存储器 (interleaved memory)**
 - 内存系统由多个内存体组成
 - 每个内存体能够独立同时访问
- **步长式存储器访问 (strided memory access)**
 - 能够访问向量中固定间隔 (步长) 的元素
 - 如: 第1,5,9个元素
- **硬件散射/聚集 (hardware scatter/gather)**
 - 指对无固定间隔的数据进行读 (gather) 和写 (scatter)

向量处理器的优点

- 优点：
 - 速度快
 - 容易使用
 - 向量编译器擅长识别向量化的代码
 - 能识别不能向量化的循环，并提供循环不能向量化的原因
 - 用户能对是否重写代码来支持向量化做出明智决定
 - 很高的内存带宽
 - 充分利用高速缓存行中的每个元素

向量处理器的缺点

- 缺点：
 - 不能处理不规则的数据结构和其他并行结构
 - 可扩展性受到限制
 - 可扩展性是指能够处理更大问题的能力
 - 新一代系统通过增加向量处理器的数目，而非增加向量长度来进行扩展
 - 对长向量的支持需要定制，非常昂贵

阵列计算机 vs 向量计算机

- 数据并行性:

- 以数据并行为主，将任务分配给多个处理器，每个处理器独立执行相同操作的不同数据元素。每个处理器通过独立处理多个数据元素来实现并行计算
- 向量计算机则通过执行相同的操作来同时处理大型连续数据向量，具有更高的数据并行性

- 硬件架构:

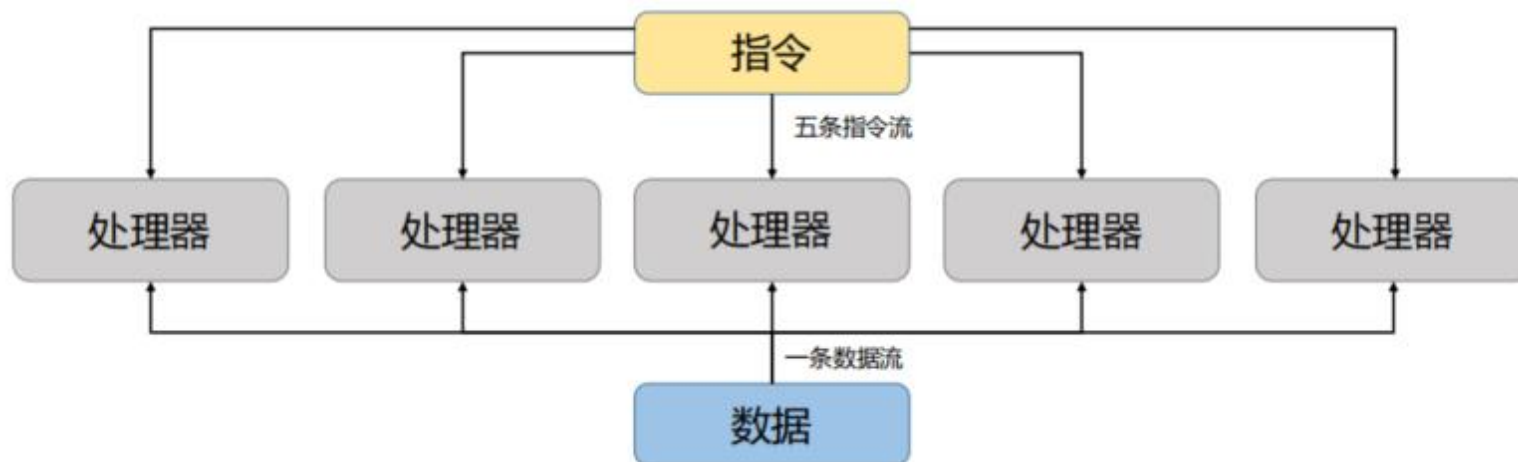
- 通常是通过大量的相同处理器节点组成的，每个节点都有自己的本地存储器，可以独立访问和处理数据
- 向量计算机则具有特殊的**硬件向量寄存器**，可以在单个指令周期内同时处理整个向量。它们通常具有更高的数据带宽和更强大的浮点性能

- 编程模型:

- 阵列计算机通常使用数据并行语言或库来编写程序，例如MATLAB和CUDA等。程序员需要显式地指定数据元素之间的并行性和通信操作
- 向量计算机则具有**专门的向量指令集和编程模型**

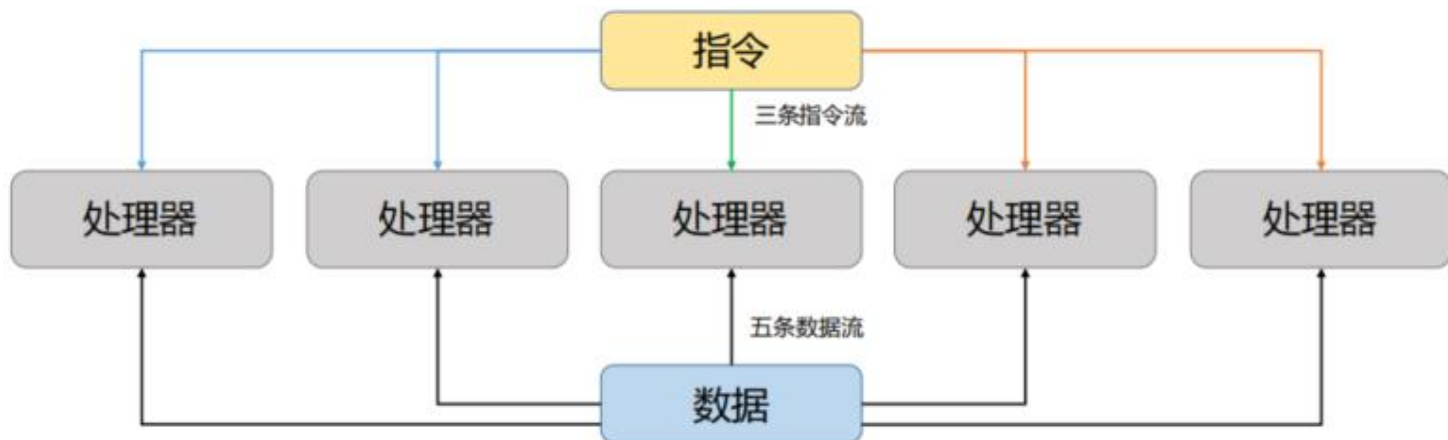
MISD体系结构

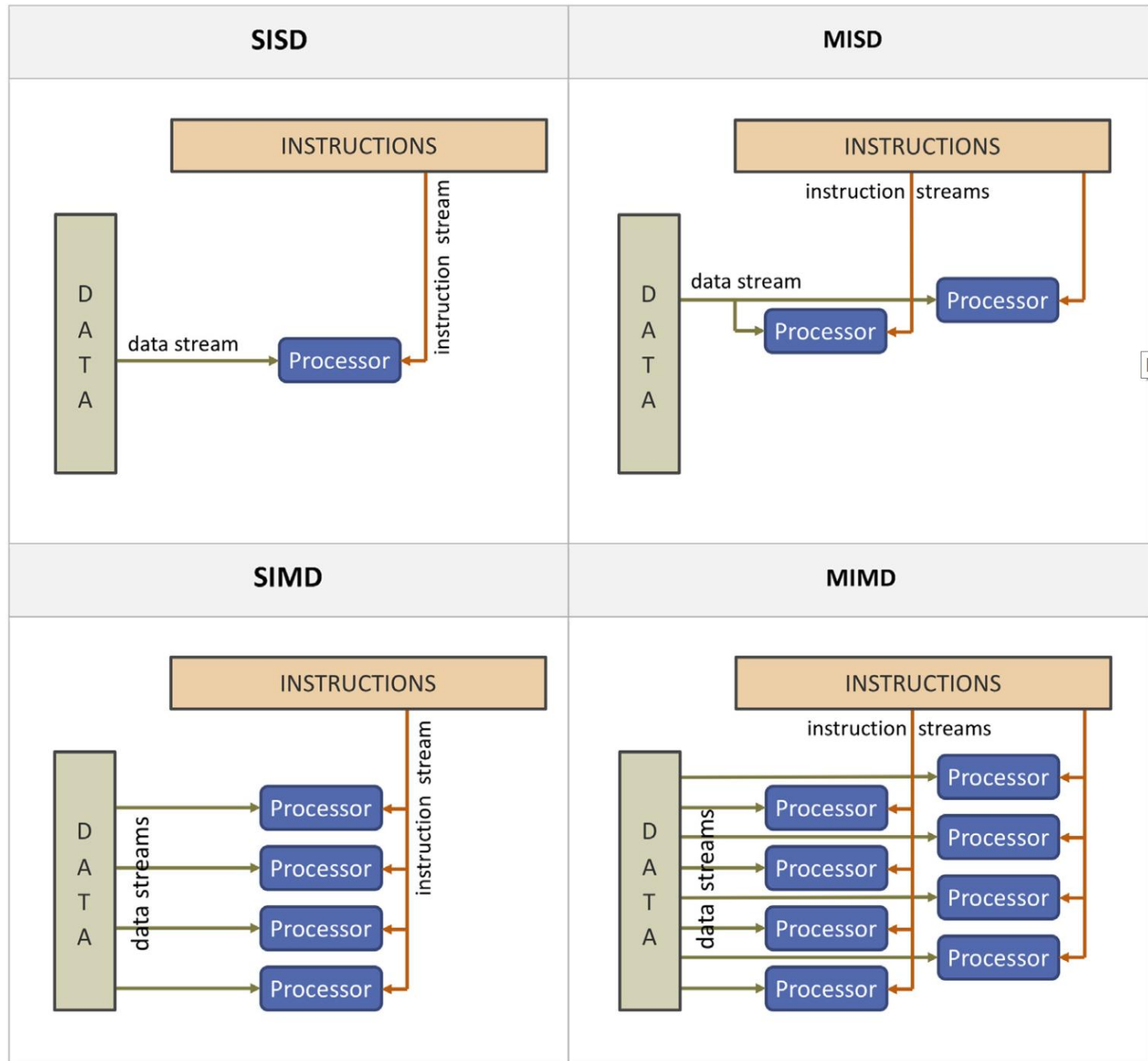
- 在MISD中，每个处理器都按照 **不同指令流** 的要求来对 **同一个数据流** 以进行 **不同的处理**
 - 存在一定争议，有人认为这种类型的计算机至今都未出现
 - 也有其他人认为有一些类似的例子，例如 **共享内存的多处理器系统** 和 **计算机中的流水线结构**



MIMD体系结构

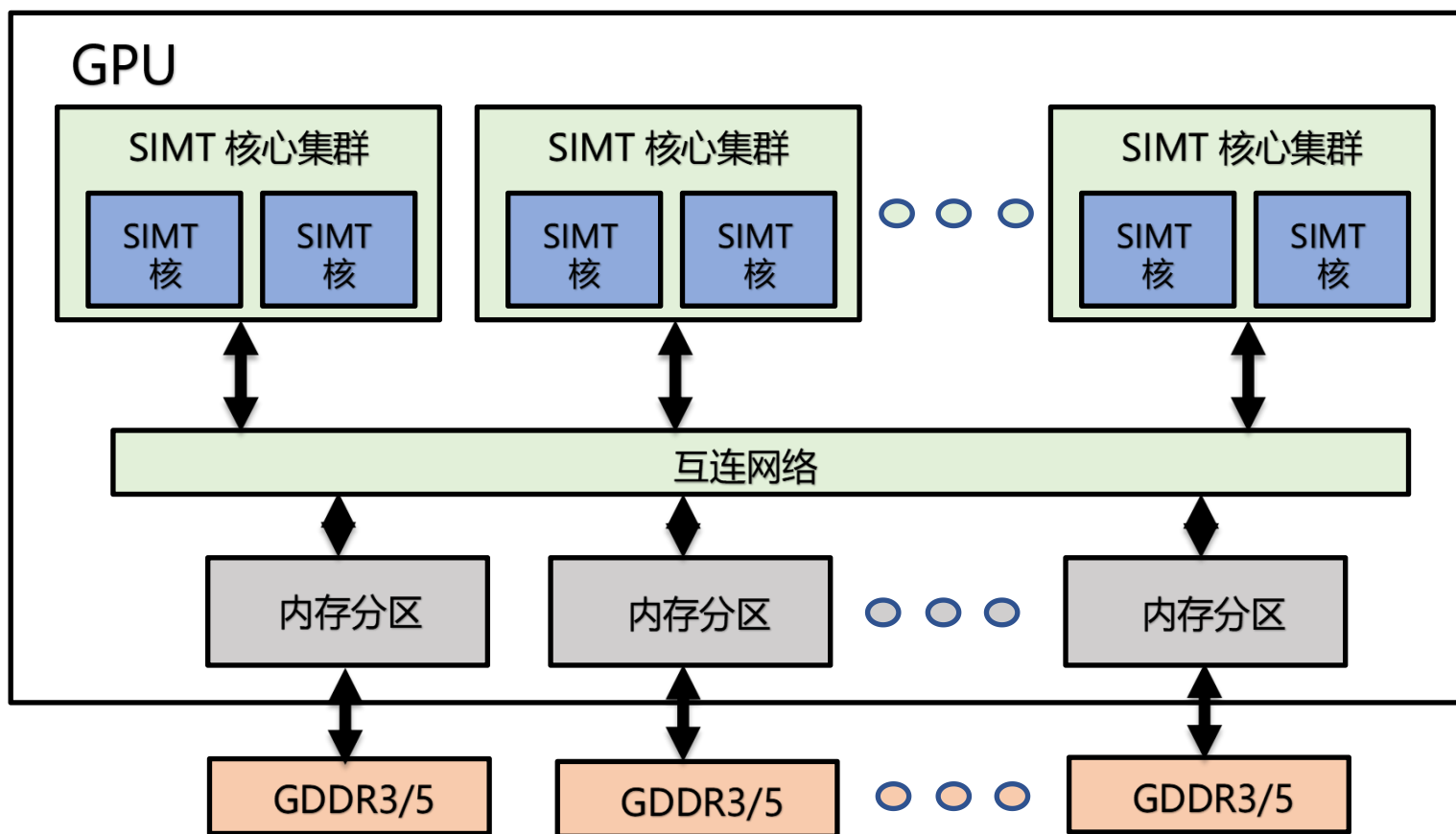
- MIMD中的每个处理器都有自己的指令流，也可以和其他处理器共享指令流，对自己的数据进行处理
 - 这是**最广泛应用的并行体系结构的形式**，现代流行的并行处理结构都可以划分为这一类
 - 超算、计算机集群、分布式系统、多处理器计算机和多核计算机都划分为这种类型





扩展：SIMT体系结构

- SIMT是一种并行计算中使用的模型，主要是将SIMD与多线程结合在一起，广泛应用于GPU上的计算单元中



SIMT vs SIMD

- **SIMT比SIMD更灵活**

1. SIMT允许一条指令的**多数据分开寻址**，而 SIMD 是必须连续在一起的片段
2. SIMT 可以支持编写**线程级的并行代码**，而 SIMD 不支持编写线程级的代码

- **SIMD中的向量中元素相互之间可以自由通信**

- SIMD的向量存在于相同的地址空间

- **SIMT中的每个线程的寄存器都是私有的**

- SIMT线程之间只能通过**共享内存**和**同步机制**进行通信

图形处理单元 (GPU)

- **实时图形应用编程接口**

- 使用点、线、三角形来表示物体的表面
- 使用图形处理流水线将物体表面的内部表示转换为一个像素的数组
- 该像素数组可以在屏幕上显示
- 流水线的许多阶段是可编程的
 - 通过着色函数 (shader) 来说明

图形处理单元 (GPU)

- **GPU使用SIMD并行来优化性能**
 - 对邻近元素使用相同的着色函数
 - 等同于使用相同的控制流
- **GPU严重依赖硬件多线程**
 - 避免内存访问的延迟
- **GPU不是纯粹的SIMD系统**
 - 在一个给定的核上ALU使用了SIMD并行
 - GPU有很多个核，每个核都能独立执行指令流

Flynn分类法的局限性

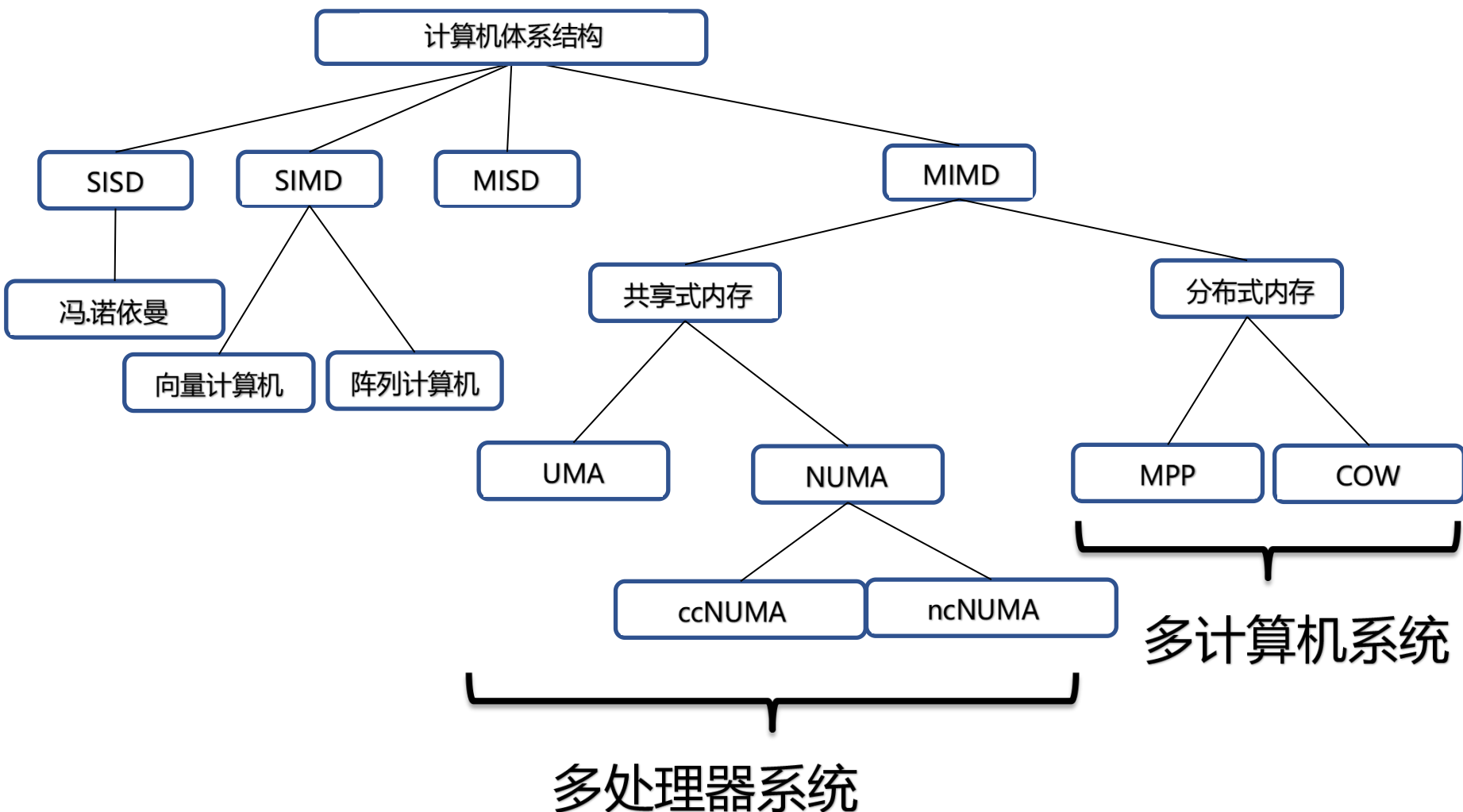
- **局限性：**

- 分类的对象主要是**控制驱动**方式下的串行处理和并行处理计算机，对于**非控制驱动方式的计算机不适合**
- 把两个不同等级的功能并列对待导致**MISD计算机不存在**
- **分类太粗**，对流水线处理机的划分不明确，如标量流水线为SISD，向量流水线为SIMD

- **其他分类方法：**

- 冯氏分类法：美籍华人教授冯泽云1972年提出按**最大并行度**来定量描述各种计算机系统
- Wolfagan Handler 在 1977 年提出根据并行度和流水线分类
- 1978 年 D.J.Kuck 提出按控制流和执行流分类

Flynn分类法的扩展

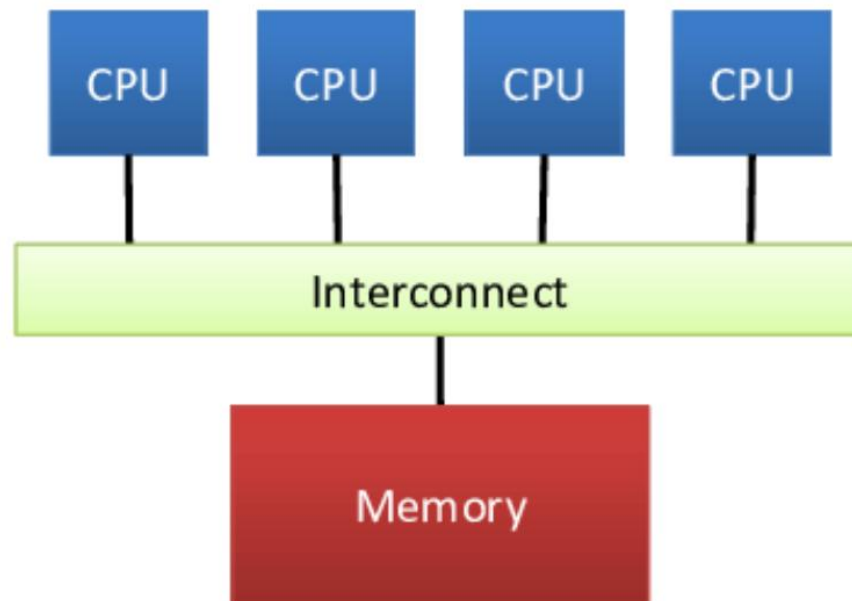


MIMD细化分类

- 根据不同的CPU是如何**组织和共享内存**的，MIMD机器分为如下的两类：
 - **共享式内存 (Shared Memory)**
 - **分布式内存 (Distributed Memory)**
- 其中有些资料将**基于分布式内存的MIMD计算机**称为基于**消息驱动 (Message-Passing)** 的计算机

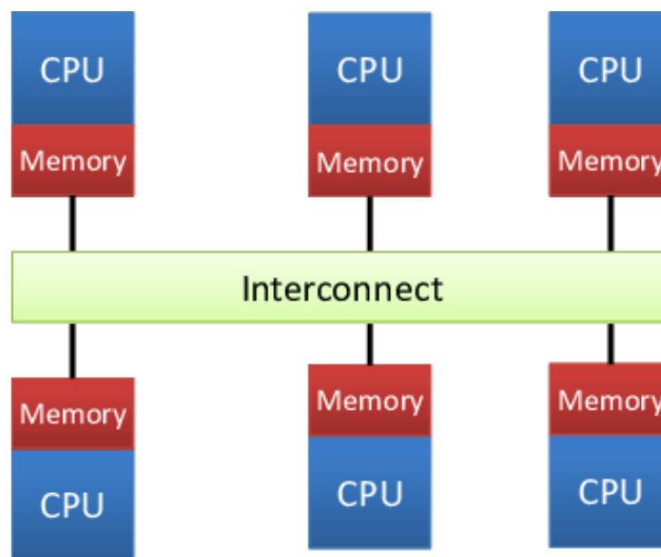
MIMD体系结构 – 共享式内存

- 共享式内存系统就是处理器之间共享内存，通过**共享内存**进行通信
 - 所有的处理器都通过软件或者硬件的方式连接到一个“**全局可用**”的存储器



MIMD体系结构 – 分布式内存

- 分布式内存系统（消息驱动）是处理器之间**不共享内存**，通过**消息驱动**来通信
 - 对于要共享的数据，它必须作为消息从一个处理器传递到另一个处理器
 - 如果两个处理器之间没有运行软件协议加以辅助，那么一个处理器就无法访问另一个存储器的数据



① 共享式内存系统

共享式内存系统

共享式内存系统可以分为如下两类：

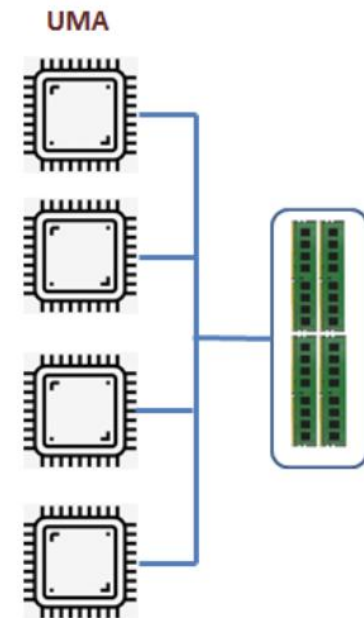
- **集中共享内存系统 (CSM):**
 - 又叫**对称多处理器系统(SMP)**、**一致存储访问系统(UMA)**
- **分布式共享内存系统 (DSM):**
 - 又叫**非一致存储器访问系统(NUMA)**

在这两种体系结构中，线程之间的通信都是通过**共享内存地址**来完成的，所谓“**共享内存**”指的就是将所有的存储器抽象成一个整体存储地址

1) 集中共享式内存

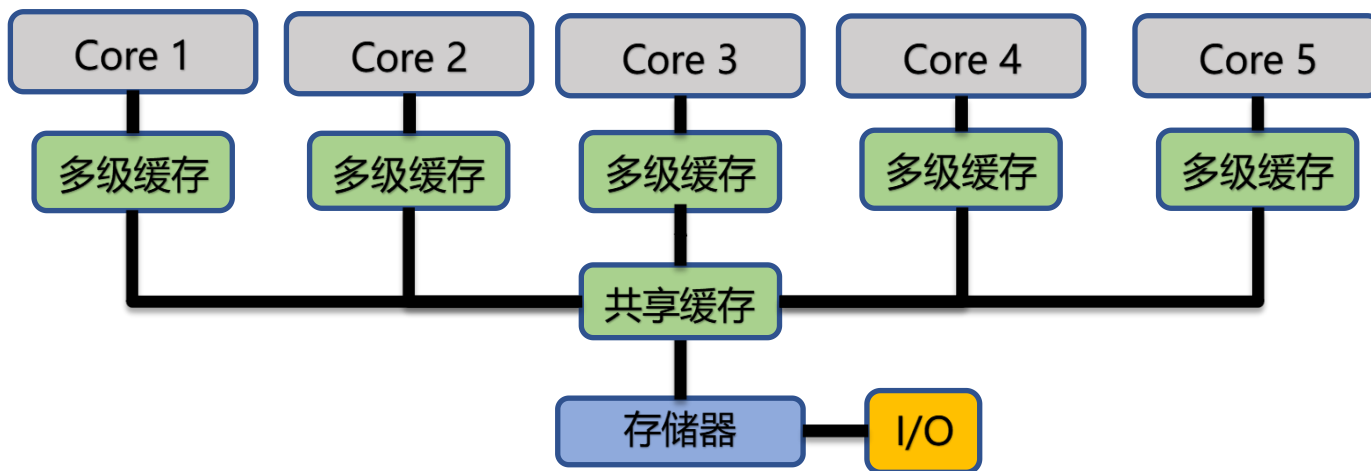
- 集中共享式内存(Centralized Shared Memory),
 - 又叫**对称多处理器** (Symmetric Multiprocessors),
 - **一致存储访问系统** (Uniform Memory Access)

- 此类多处理器中处理器数目较少, 所以处理器之间能够共享一个集中式的存储器
- 所有的存储器能够平等的访问它, 这就是**对称**一词的由来
- 又因为每个处理器都能平等地访问存储器, 所以它们访问存储器的延迟都是相同的, 因此又被叫做**一致**存储访问系统

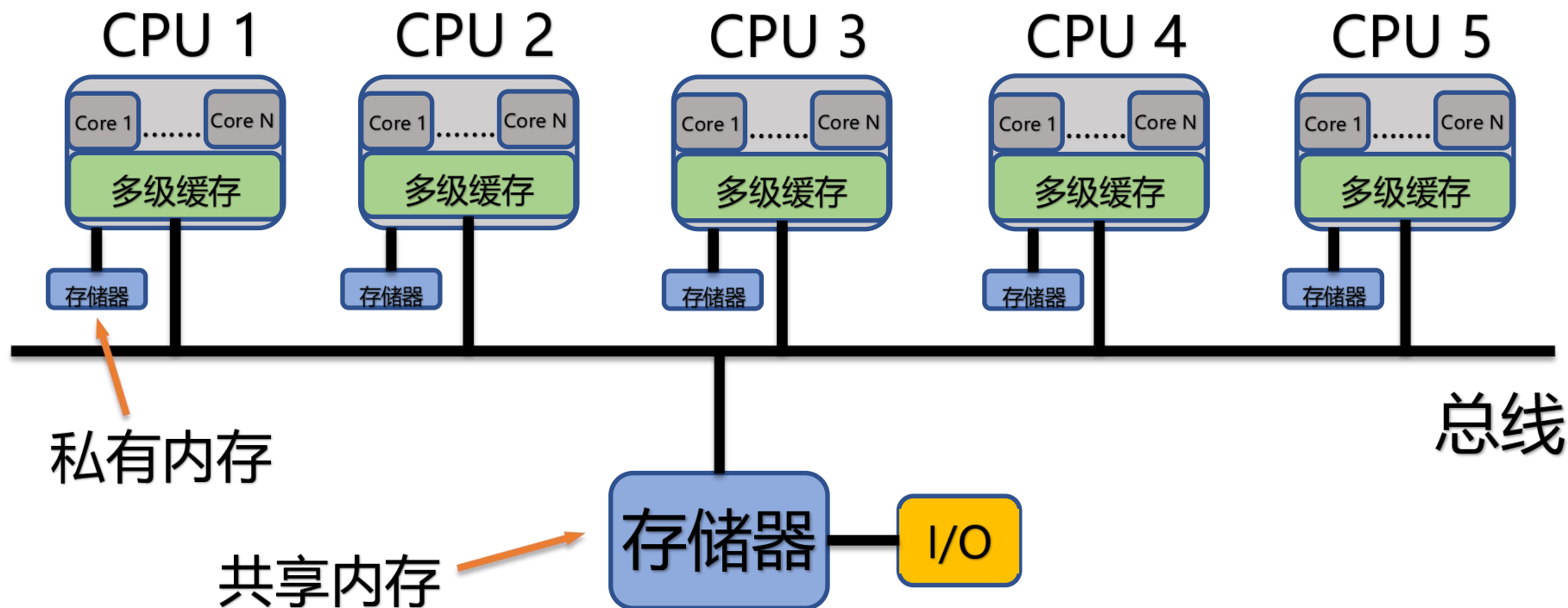


集中共享式内存系统的特点

- 有一个存储器被所有处理器**均匀共享**
- 所有处理器访问共享的存储器的**延迟相同**
- 每个处理器可以拥有**私有内存或高速缓存**



基于多核芯片的集中共享内存系统的基本结构

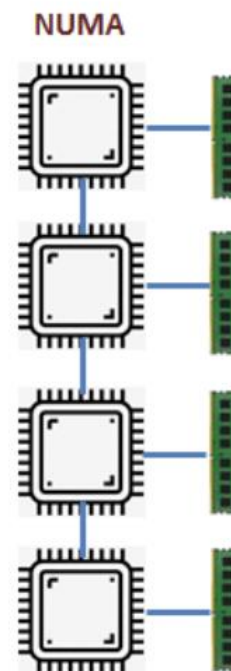


- 上图展示了基于多个芯片的集中共享内存系统（带缓存和私有内存）
- 与多核芯片不同，这里每个芯片可以认为是一个处理器，将所有的处理器通过总线或者互连网络连接到同一个存储器，而不是在一个芯片的内部，并且每个处理器都有自己的私有内存和共享内存

2) 分布式共享内存系统

- **分布式共享内存系统 (Distributed Shared Memory),**
 - 又叫非一致存储访问系统(Non-Uniform Memory Access)

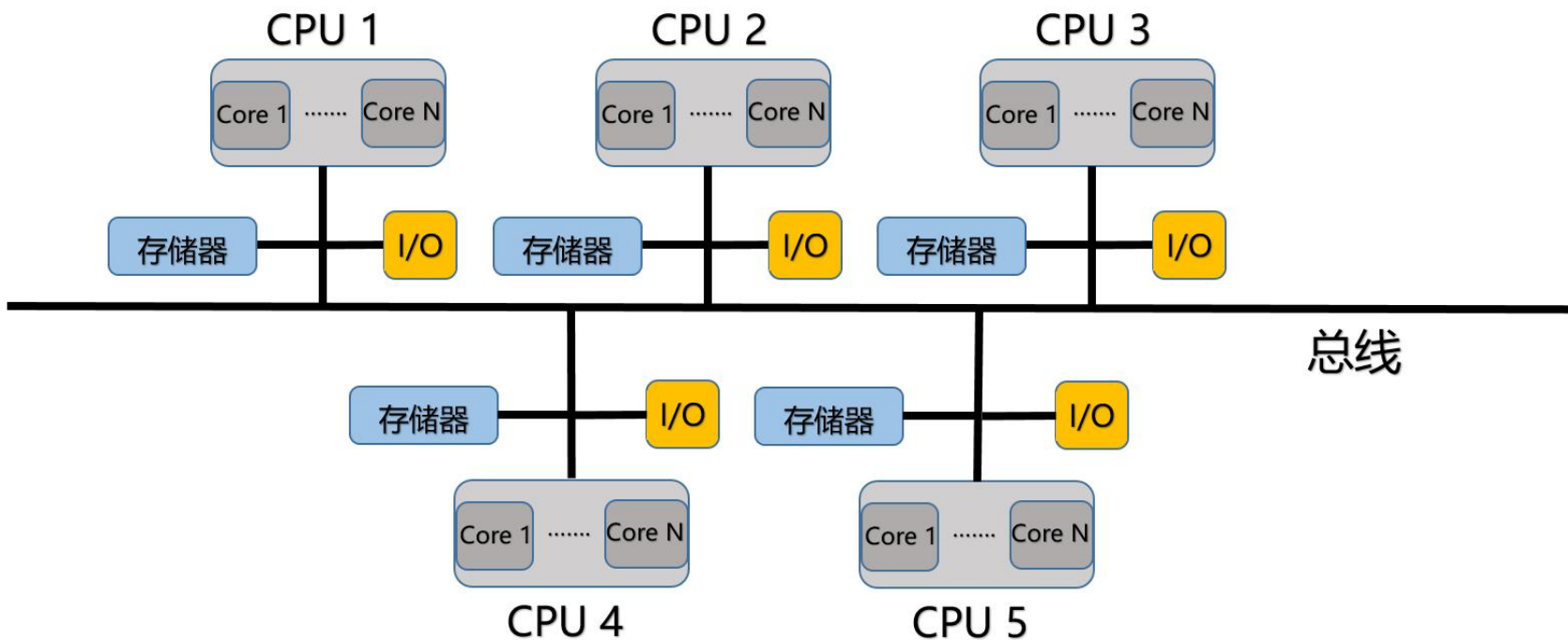
- 每个处理器都拥有自己的存储器，也可以访问其他节点的存储器
- 这类系统之所以叫做非一致存储访问系统，是因为每个结点访问**本地内存**和访问其它结点的**远程内存**的**延迟是不同的**



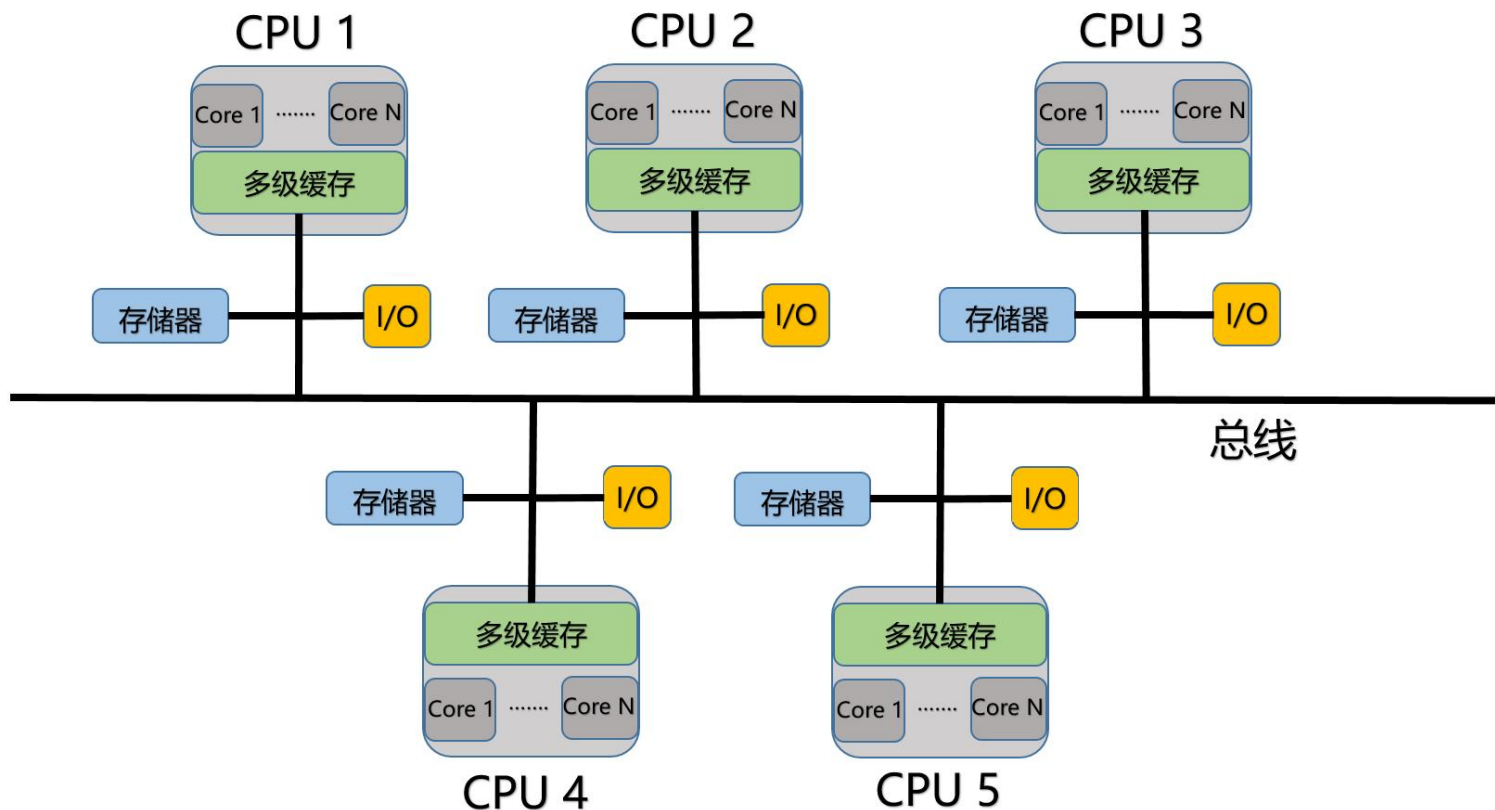
分布式共享内存系统的特点

- 所有的处理器都能访问一个单一的地址空间
- 使用LOAD和STORE指令访问远程内存
- 访问远程内存比访问本地内存延迟要高
- 每个处理器可以使用高速缓存

目前几乎所有的多核心多处理器系统都使用了分布式存储器



- 上图展示了 **分布式共享内存系统（不带缓存）**，又叫 **NC-NUMA (No Cache-NUMA)**



- 上图展示了 **分布式共享内存系统（带缓存）**，又叫 **CC-NUMA (Cache Coherent-NUMA)**
 - 这两种类型的共享存储器在物理上是分布在**不同的处理器的本地存储器**之上
 - 但是他们都通过总线或者互连网络相连接，可以把所有这些本地存储**抽象成一个全局的存储器**，并能够被任何处理器访问

分布式共享内存

- 利用分布式共享内存技术，可以把几十个甚至上百个CPU 集中在一台计算机中
- 随着多核心处理器的推广，这种结构在当前的计算机中非常普遍。然而这种技术需要在处理器之间的数据传送和同步上花费更多的资源，因此设计的协议规则也更加复杂，需要在软件层面进行专门设计以充分提升分布式共享内存的带宽
- 目前大量多核心多处理器系统使用了分布式存储器结构
 - 华为鲲鹏处理器
 - 多路服务器

② 分布式内存系统

分布式内存系统

分布式内存系统可以分为如下两类：

① 大规模并行处理器系统 (MPP)

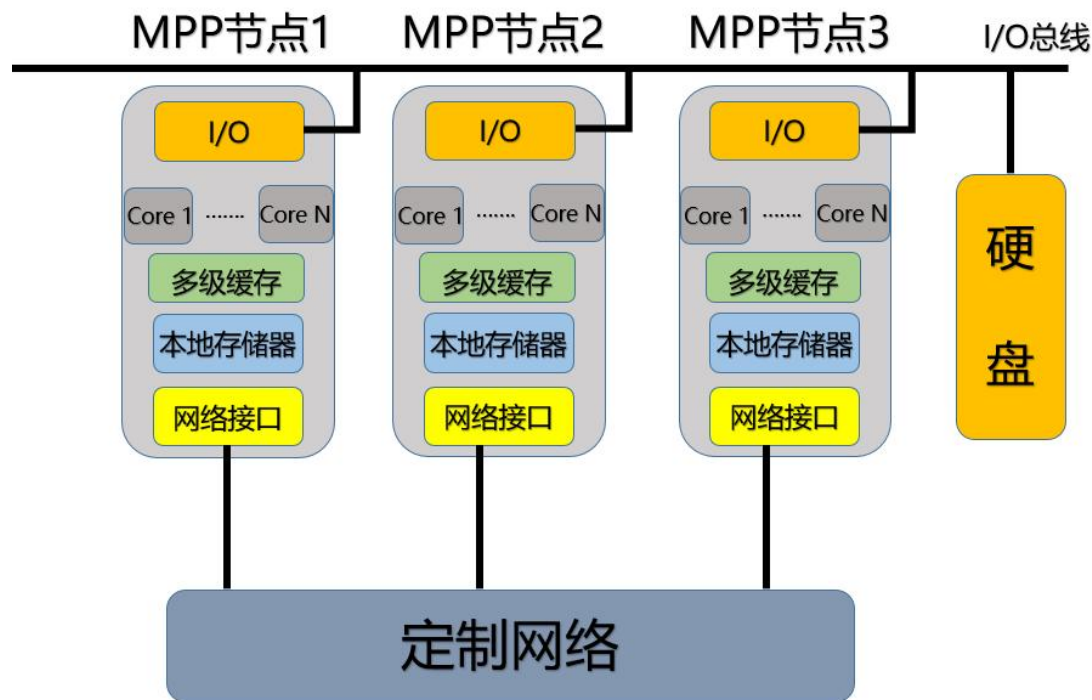
② 工作站机群系统 (COW)

- 在这种计算机体系结构中，每台计算机使用**消息机制**（如以太网）连接起来
- 每台计算机都有自己的处理器，每个处理器都有**自己的私有内存**，私有内存只提供自己的处理器进行访问
- **其他的计算机不能直接访问**，每个计算机都有自己独立的物理地址空间

1) 大规模并行处理系统

- **MPP：大规模并行处理器（Massively Parallel Processors）系统**
 - MPP系统是由成百上千台计算机组成的大规模高性能计算机系统
 - 主要应用于科学技术、工程模拟等以计算为主的科研工作中
 - 该系统一般开发困难，价格高，市场有限，是国家和公司综合实力的象征

大规模并行处理系统基本架构

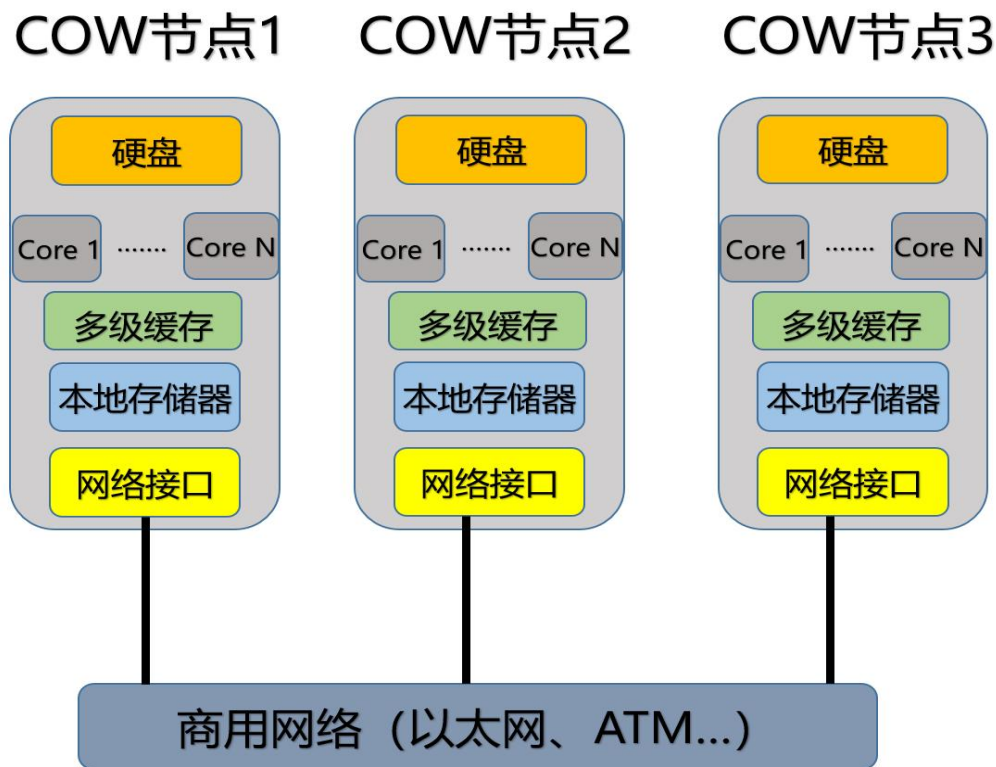


- MPP中一般每个节点可以认为是一个**没有硬盘的计算机**
- MPP节点一般只驻留**操作系统内核**，由一条I/O总线连接到**同一个硬盘**上面
- MPP使用的网络一般情况不是我们使用的高速以太网，它一般使用制造商专有的**定制高速通信网络**

2) 工作站机群系统

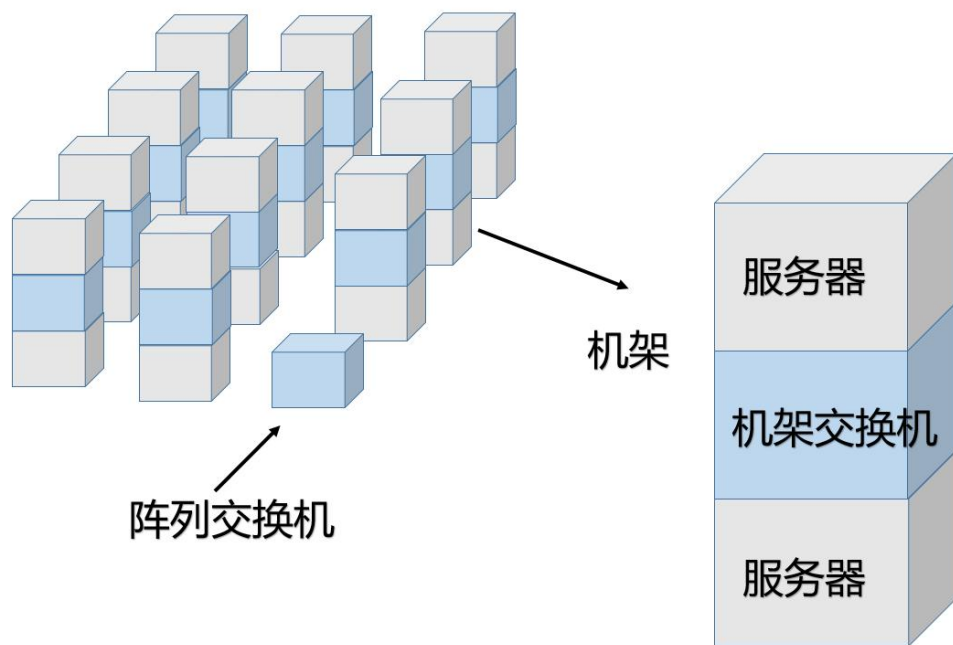
- COW: 工作站机群系统 (Cluster Of Workstations)
 - 又叫仓库级计算机 (Workstations Cluster)
 - COW系统是由大量的家用计算机或者工作站通过商用网络连接在一起而构成的多计算机系统
 - 很多大公司的数据中心就是一个典型的例子
 - 随着云计算的蓬勃发展，COW正在变得越来越重要
 - 相比过去为科学家和工程师们提供高性能计算的角色，现在的COW已经成为为全世界提供信息技术的基础

工作站机群系统的基本架构

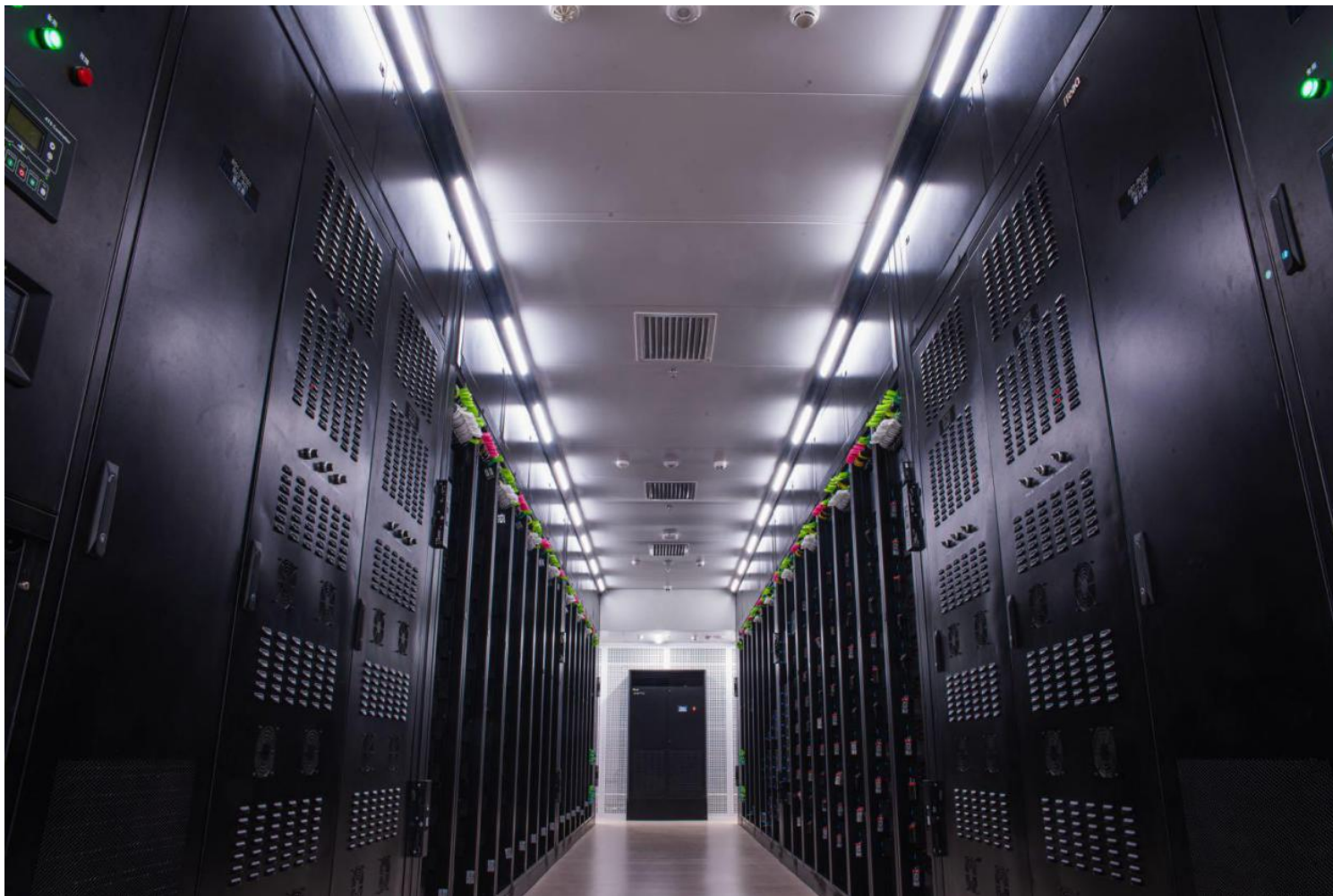


- COW中每个节点都可以认为是一台**独立的计算机**，它们有自己的硬盘、CPU、存储器等，在商用网络的协作下组成一个工作站机群系统

COW 交换机层次结构



- **COW 通常由多个服务器阵列排列而成**
 - 其中机架是容纳服务器、交换机的外壳框架，一个机架上往往有多个服务器，服务器之间通过机架交换机进行通信
- **服务器阵列是由多个机架排列而成，阵列内部机架间通过阵列交换机通信**



腾讯云计算数据中心机房



Google的水下数据中心

- COW的成本极高，它包含了机房、配电与制冷基础设施，服务器和联网设备，一般情况下一个COW能够容纳上万台服务器



神威太湖之光超级计算机

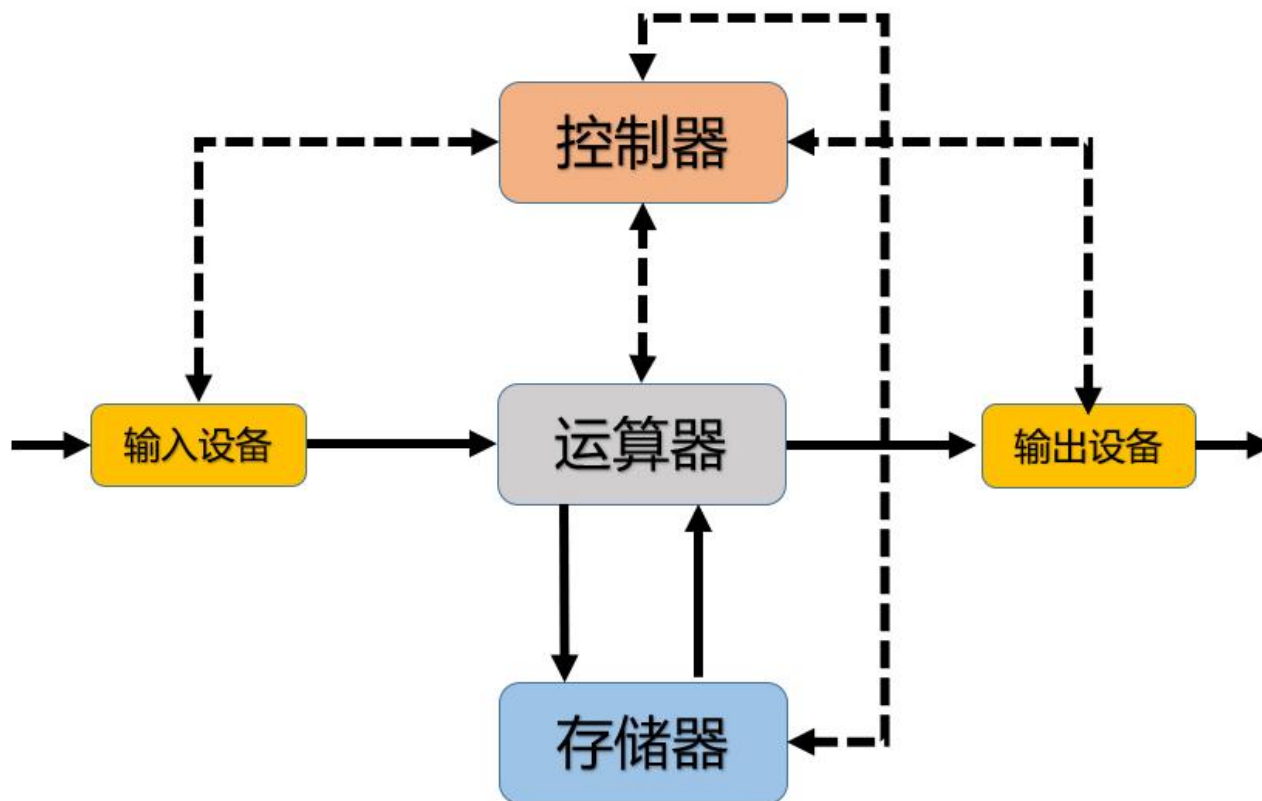
- 根据2018年6月全球超级计算机排行榜Top500名单中，有**437台**超算采用的是**COW架构**，另外**63台**采用的是**MPP架构**
- 前十名中，有**6台**使用的是MPP架构，中国的神威·太湖之光使用的是MPP架构

COW vs HPC

- 部分HPC采用的是COW的架构，但是一般不采用商业互联网和商业芯片，HPC一般使用定制的芯片和通信网络
- HPC强调线程级并行或数据级并行，而COW则强调请求级并行，即可能有多个网络请求同时访问一台机器
- HPC常常满负载持续数周完成大规模运行作业，而COW是面向并发请求的，通常不会满负载

(三) 非冯·诺依曼体系结构

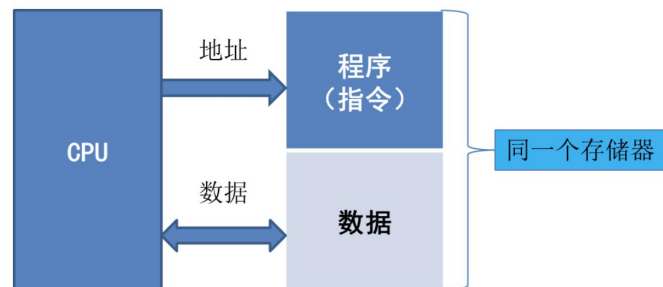
传统冯·诺依曼体系结构



以运算器为核心的冯·诺依曼计算机构成图

冯·诺依曼体系结构的特点

- 单处理机结构，机器以**运算器**为中心
- 采用**程序存储**思想
- **指令和数据无差别**的存储在存储器内
- 数据以**二进制形式**表示
- 软件和硬件**完全分离**
- 指令由**操作码和操作数**组成
- 指令都是**顺序执行**的



非冯诺依曼结构出现的原因

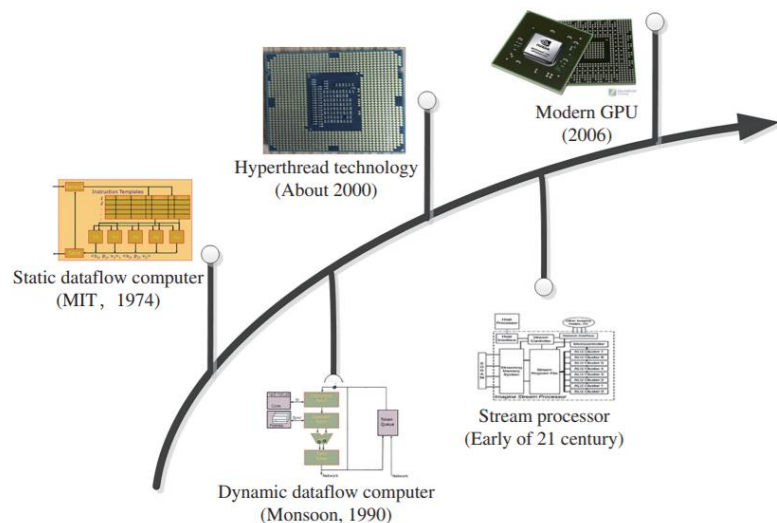
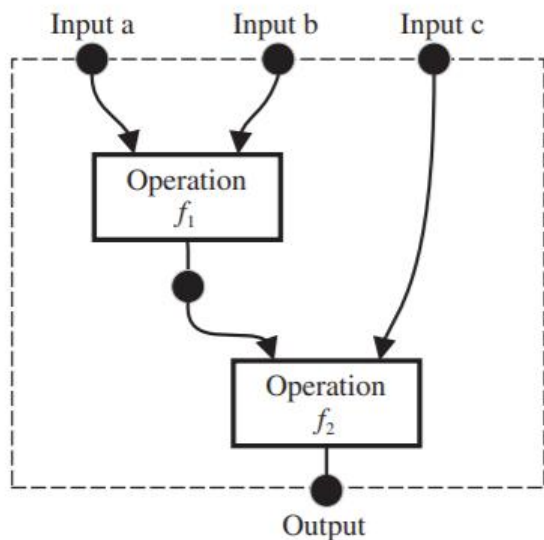
- 传统的冯·诺依曼型计算机的缺点：
 - 从本质上讲是采取串行顺序处理的工作机制
 - 即使有关数据已经准备好，也必须逐条执行指令序列
 - 指令与数据在同一存储器
 - 在高速运行时，不能达到同时取指令和取操作数，从而形成了传输过程的瓶颈
- 非冯·诺依曼化的探讨：
 - 对传统冯·诺依曼机进行改造，如采用多个处理部件形成流水处理
 - 用多个冯·诺依曼机组成多机系统，支持并行算法结构
 - 从根本上改变冯·诺依曼机的控制流驱动方式，如采用数据流

非冯·诺依曼结构计算机

- **归约机(Reduction Machine)**
 - 一种基于函数式语言编程的计算机，可根据表达式中的运算信息处理相应的数据
- **数据流计算机(Data Flow Computer)**
 - 一种基于数据流的计算机，每条指令的执行都是由数据来驱动的
- **量子计算机(Quantum Computer)**
 - 一种基于量子逻辑的计算设备，利用不同量子态来记录状态，并且使用量子算法来操作数据
- **光子计算机(Optical Computer)**
 - 以光子替代电子的先进计算机，将光子设备集成到当前的电子计算机中，形成光电混合的系统

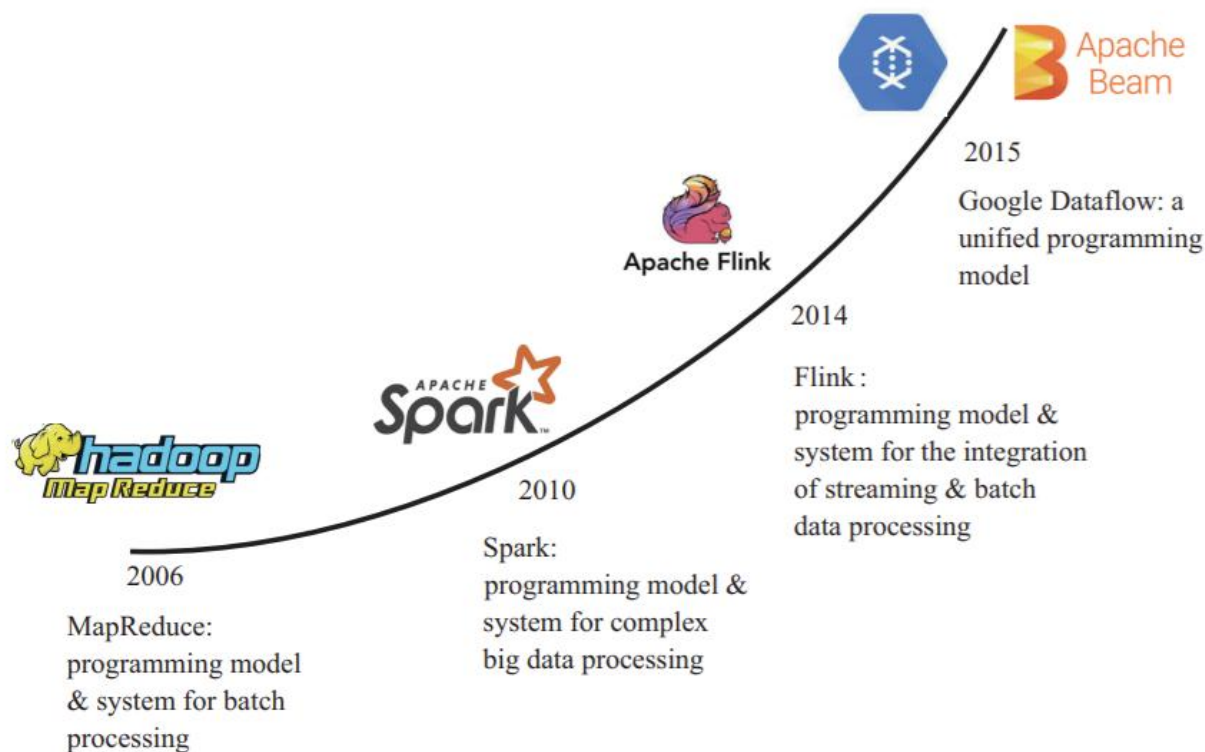
数据流计算机

- 数据流计算思想回答了 **“一个运算操作能够被执行的充分条件是什么”** 这个科学问题
 - 即**数据就绪开始计算**，同时提出了用数据流图描述计算任务的具体方法
 - 创新之处就在于**打破了传统串行执行指令的思维禁锢**
 - 数据流计算机是探索利用数据流计算思想提升计算机系统性能的实践者和先驱者



数据流思想在常用计算模型中的应用

- 数据流计算思想给云计算与大数据分析系统领域的编程模型带来了重大变化
 - 指令是根据数据的可用性而不是按照严格的控制流顺序来执行的



数据流计算机的先驱

数据流程编程由Jack Dennis和他在MIT的研究生于1960年代开创



Jack Dennis (1931-) MIT教授

- IEEE John von Neumann Medal, 2013
- ACM SIGOPS Hall of Fame, 2012
- Member of the National Academy of Engineering (NAE), 2009
- Eckert-Mauchly Award, 1984
- IEEE Fellow
- ACM Fellow

高光荣教授 (1945-2021)

- 美国特拉华大学教授
- 新中国成立后第一位获得麻省理工学院计算机博士的学者
- 毕生致力于数据流架构领域研究
- ACM Fellow/IEEE Fellow, 2007
- CCF “海外杰出贡献奖”, 2013
- “罗摩克里希纳劳奖”, 2017 (全球第七位、中国大陆第一位获得该奖项的科学家)

王兴 (1979年-) 美团CEO

- 2001年毕业于清华大学无线电专业
- 2003年中断美国特拉华大学博士学业回国创业, 创立校内网, 2006年被收购;
- 2007-09年创办经营饭否网;
- 2010年创办团购网站美团网

IBM Cyclops64超级计算机系统

本章小结

- 超级计算机硬件
- 高性能计算机的分类
 - **SISD, SIMD, MISD 和 MIMD**
 - 其中**共享内存系统**又分为：集中式共享内存系统 (CSM/SMP/UMA) 和分布式共享内存系统(DSM/NUMA)
 - 而**分布式内存系统**分为：大规模并行处理系统(MPP) 和 工作站机群系统 (COW/WSC)
- 非冯·诺依曼体系结构