第二章
并行计算机体系结构

哈尔滨工业大学
郝萌

2023, Fall Semester

目录

- **并行系统分类**
- 共享内存系统
- 分布式内存系统
- 异构系统架构
- 互连网络

并行计算机

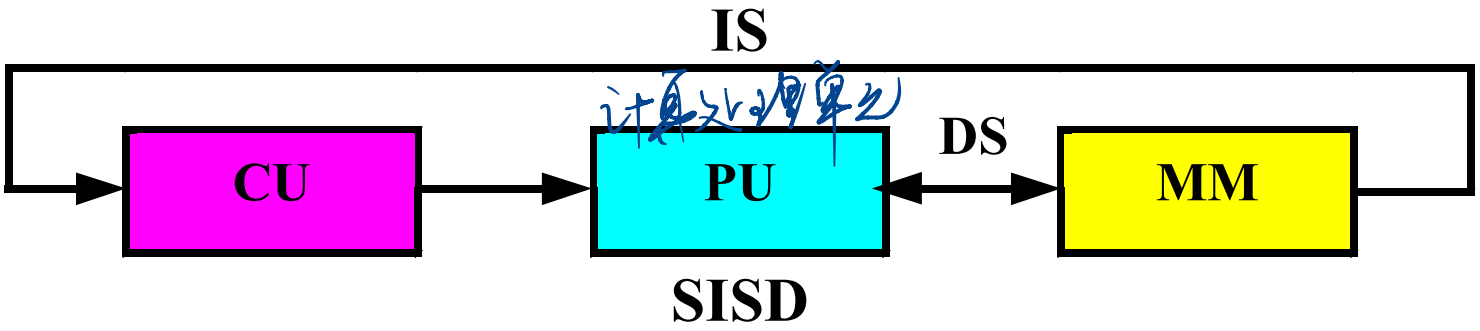
- 由一组处理单元组成
- 各处理单元之间相互通信与协作
- 以更快的速度共同完成一项大规模计算任务

Flynn分类法

S I S D **Single Instruction, Single Data** **串行计算机(von Neumann计算机)**	**S I M D** **Single Instruction, Multiple Data** **适用性很有限(如MPEG类计算、字符串匹配计算)**
M I S D **Multiple Instruction, Single Data** **为完美分类而设置，意义不大**	**M I M D** **Multiple Instruction, Multiple Data** **常见的并行计算机都可归入此类** **MPP/Cluster/SMP/当前基于Cache的Multi-core (Intel、AMD)**

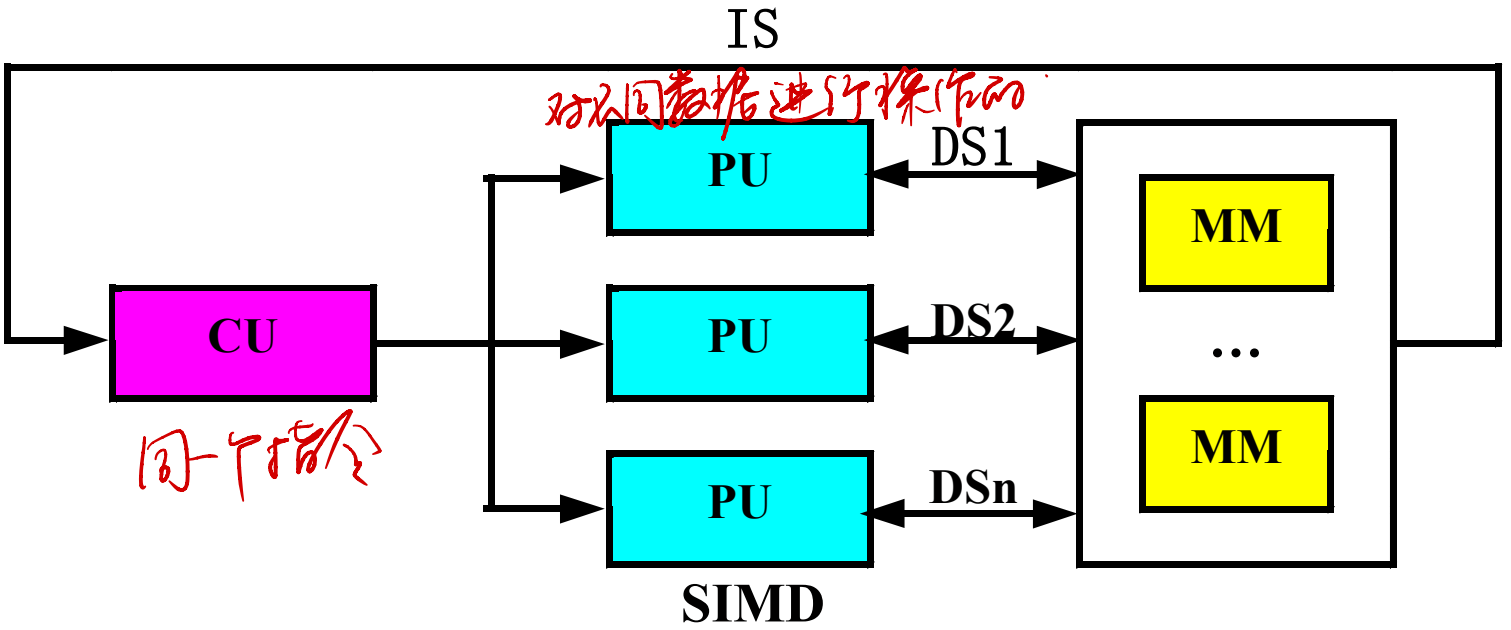
Flynn分类法——SISD

- 处理器串行执行指令
- 或者处理器内采用指令流水线，以时间重叠技术实现了一定程度的指令并行执行
- 甚至处理器是超标量处理器，内有几条指令流水线实现了更大程度上的指令并行执行
- 都是以单一的指令流从存储器取指令，以单一的数据流从存储器取操作数和将结果写回存储器



Flynn分类法——SIMD

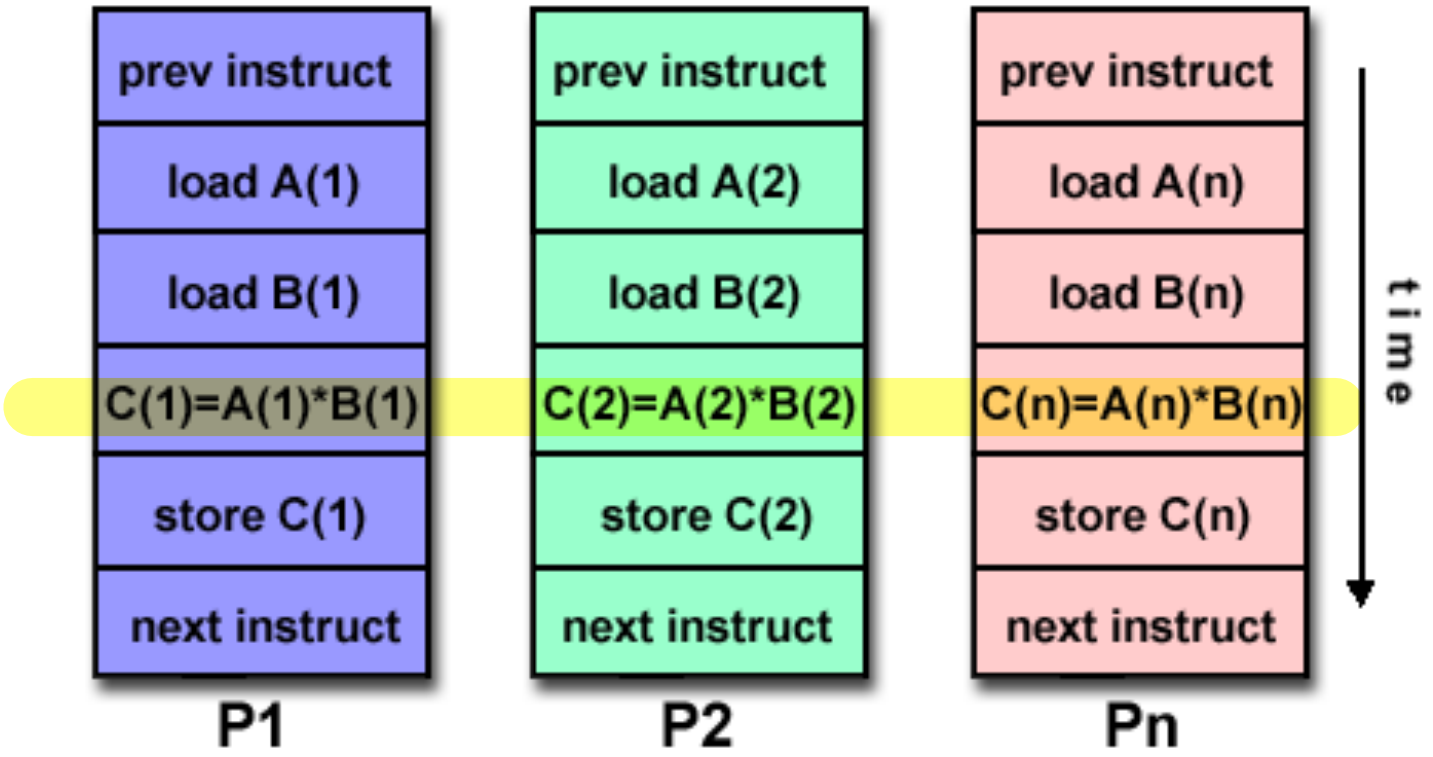
- 有单一的控制部件，但是有多个处理部件
- 计算机以一个控制单元从存储器取单一的指令流，一条指令同时作用到各个处理单元，控制各个处理单元对来自不同数据流的数据组进行操作



Flynn分类法——SIMD

数组操作，正常：循环→一条指令。

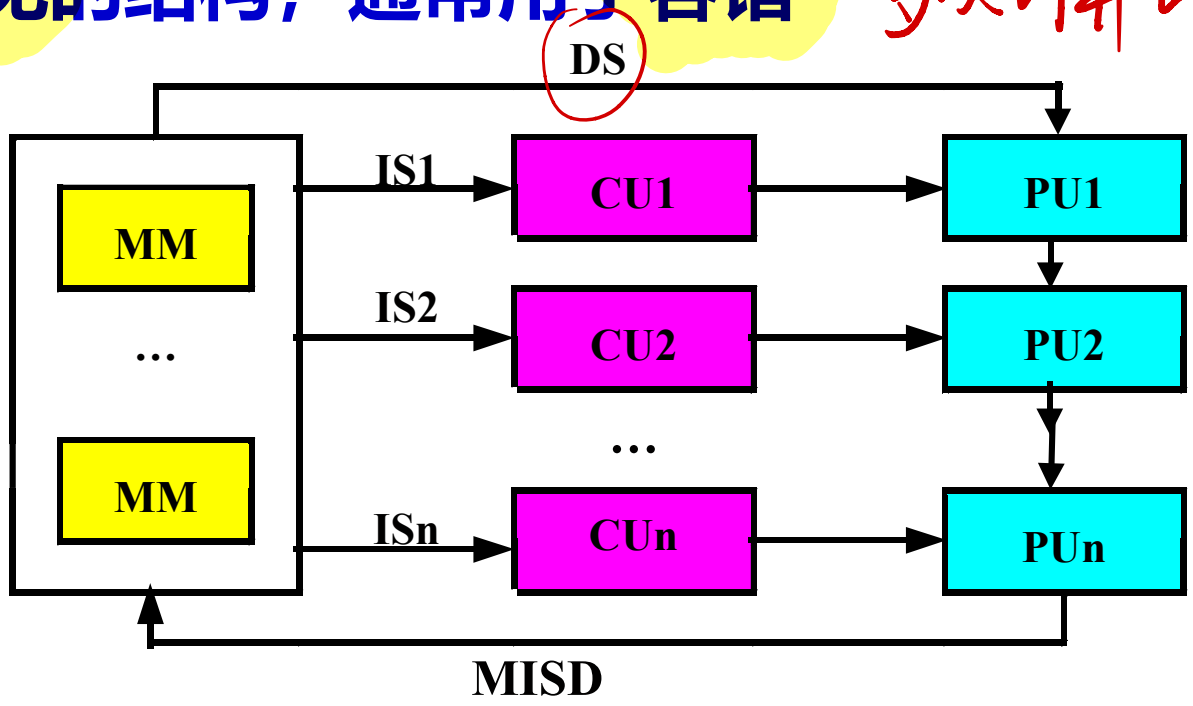
- 多用于矢量运算中，加速指令运算，比如矩阵乘
- 适用于非常规则的计算，例如：视频、音频处理的MPEG算法；密集矩阵的运算



Flynn分类法——MISD

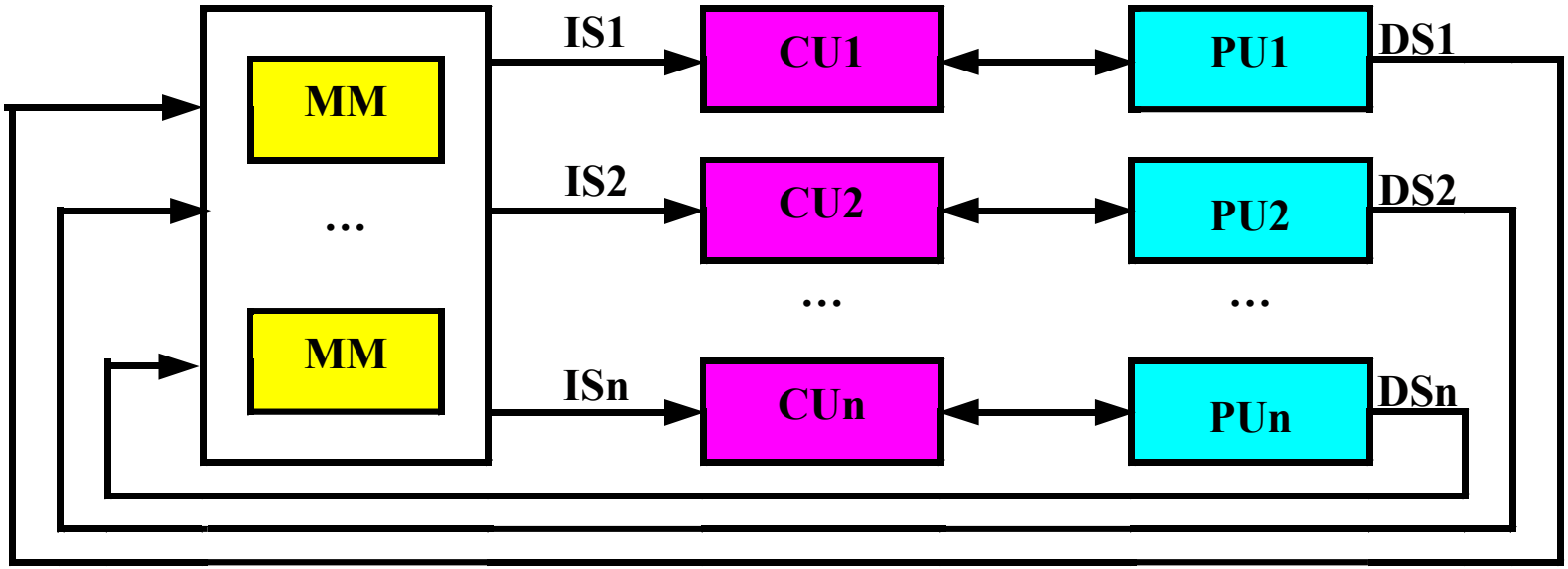
- 多个处理单元，各配有相应的控制单元
- 各个处理单元接收不同的指令，多条指令同时在一份数据上进行操作
- 不常见的结构，通常用于容错

多次计算比较



Flynn分类法——MIMD

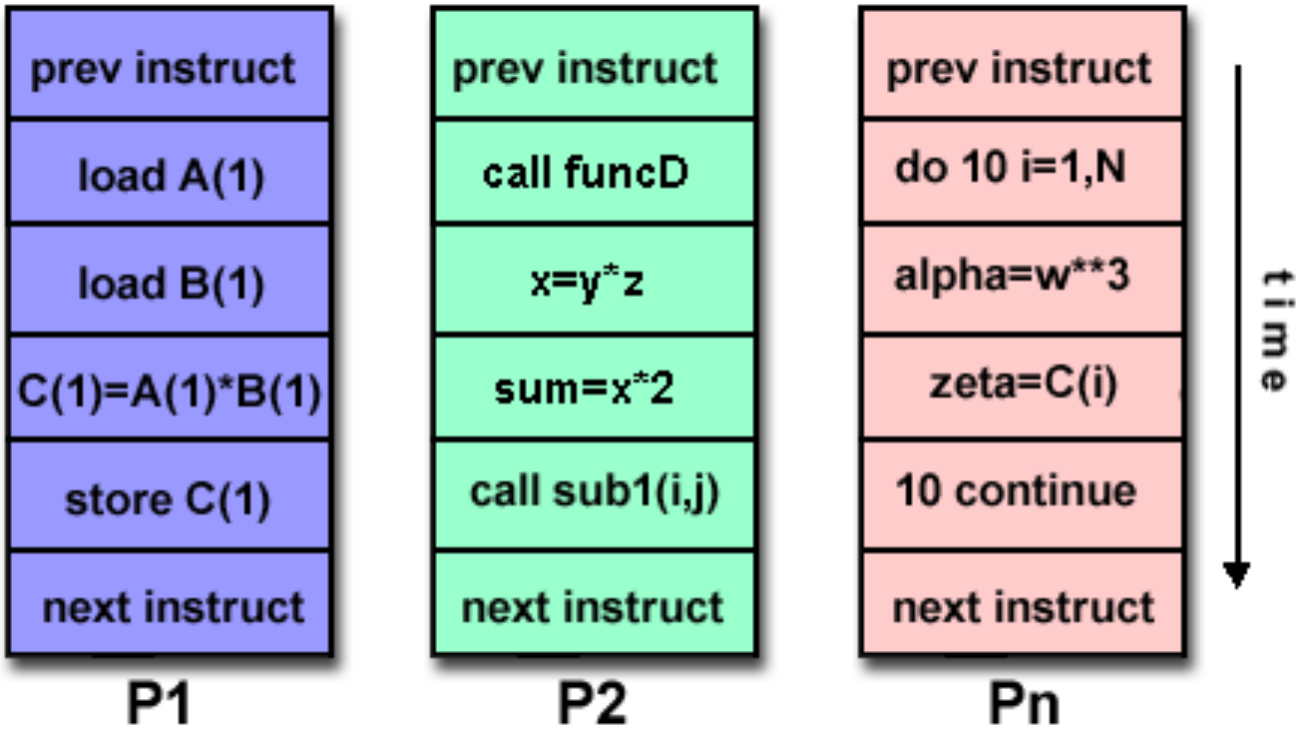
- 有多个处理单元，每个处理单元有相应控制单元
- 各个处理单元可以接收不同的指令并对不同的数据流进行操作



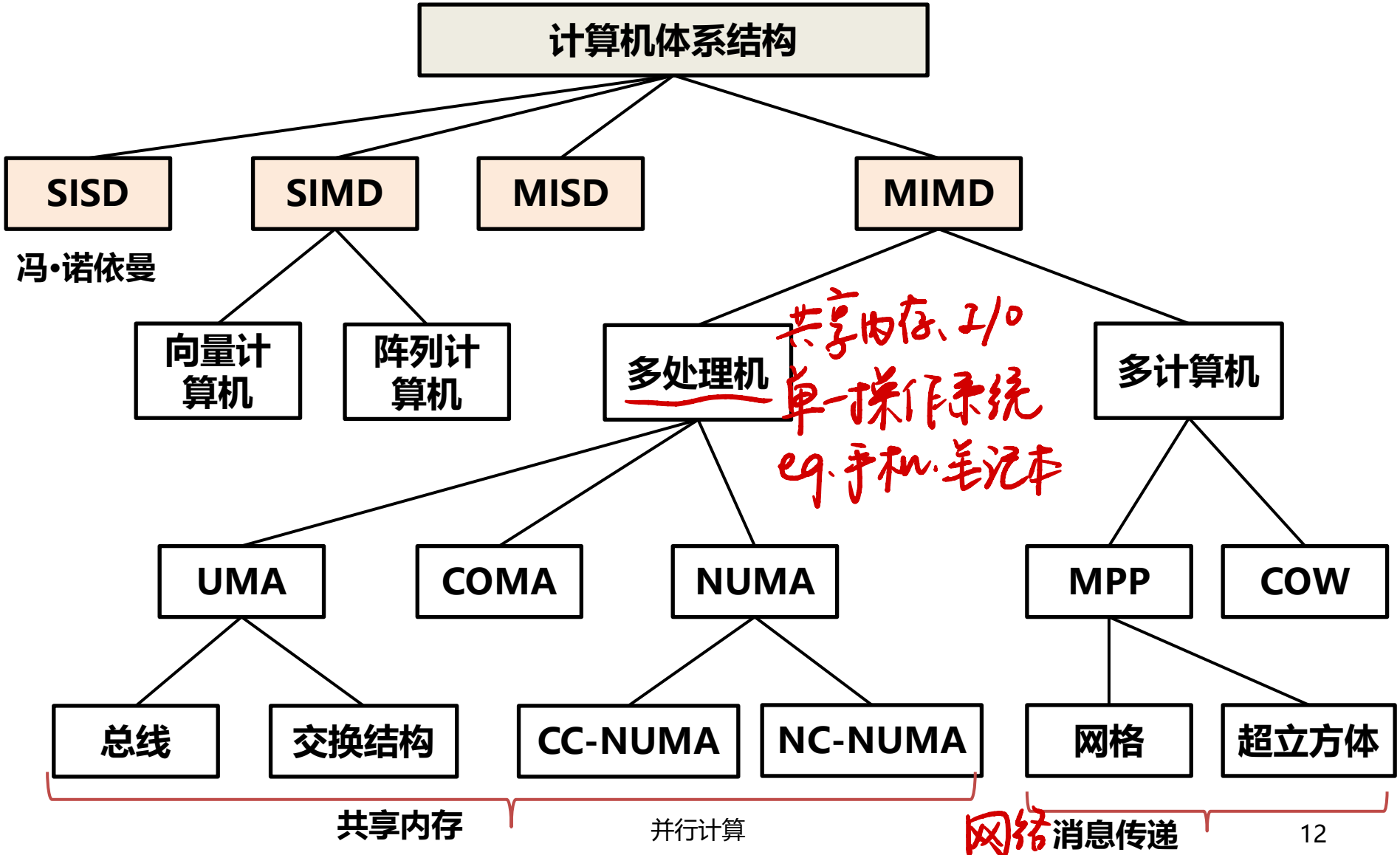
MIMD

Flynn分类法——MIMD

- 大多数现代并行计算机都属于这一类
- 对称多处理机 SMP，大规模并行处理机 MPP，工作站机群 COW，分布式共享存储系统 DSW



并行计算机体系结构图谱



MIMD体系结构

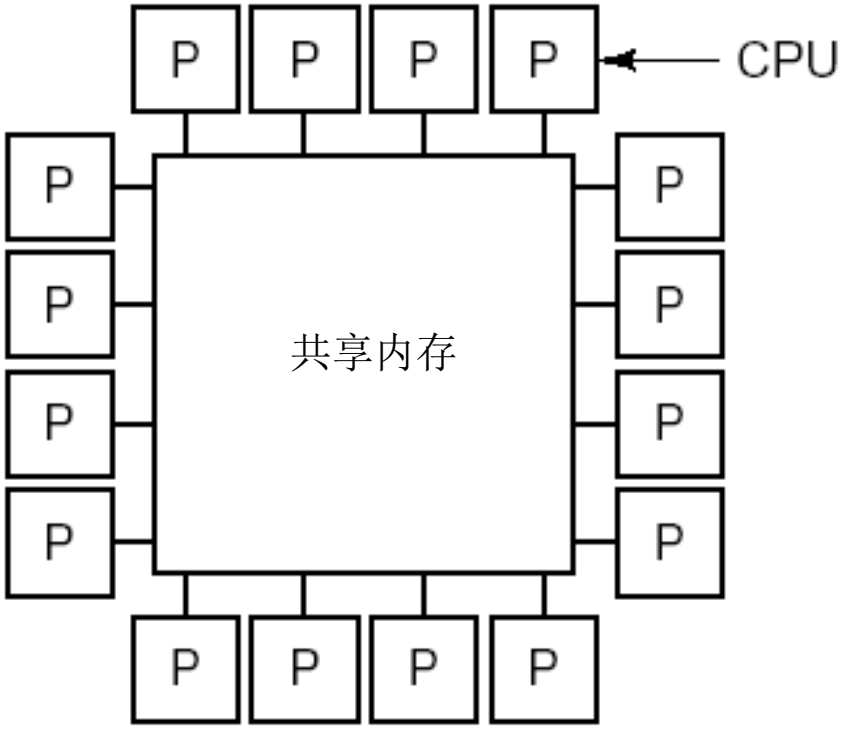
- 多处理机系统——基于共享内存
  - 系统中只有唯一的地址空间，所有的处理器共享该地址空间
  - 唯一的地址空间并不意味着在物理上只有一个存储器。共享地址空间可以通过一个物理上共享的存储器来实现，也可以通过分布式存储器并在硬件和软件的支持下实现
- 多计算机系统——基于消息传递(分布式内存)
  - 每个处理器有自己的存储器，该存储器只能被该处理器访问而不能被其他处理器直接访问，这种存储器称为局部存储器或私有存储器
  - 当处理器A需要向处理器B传送数据时，A把数据以消息的形式发送给B

网络通讯

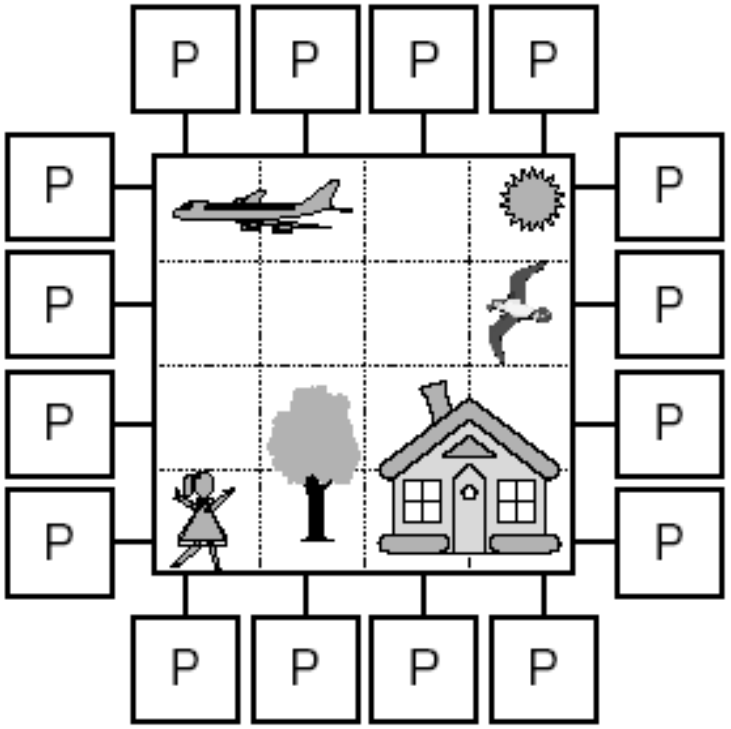
目录

- 并行系统分类
- **共享内存系统**
- 分布式内存系统
- 异构系统架构
- 互连网络

共享内存的多处理机



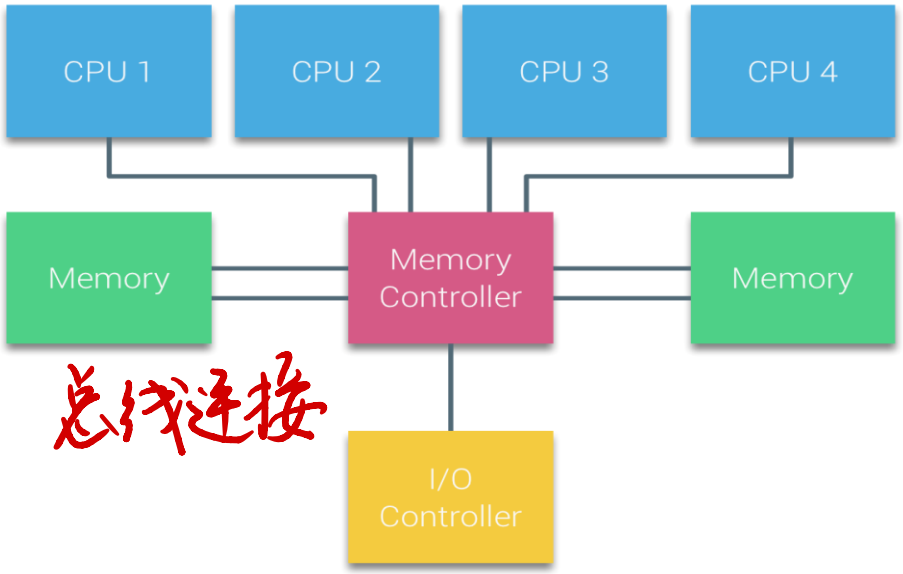
(a) 16个CPU共享一个公共内存的多处理机系统

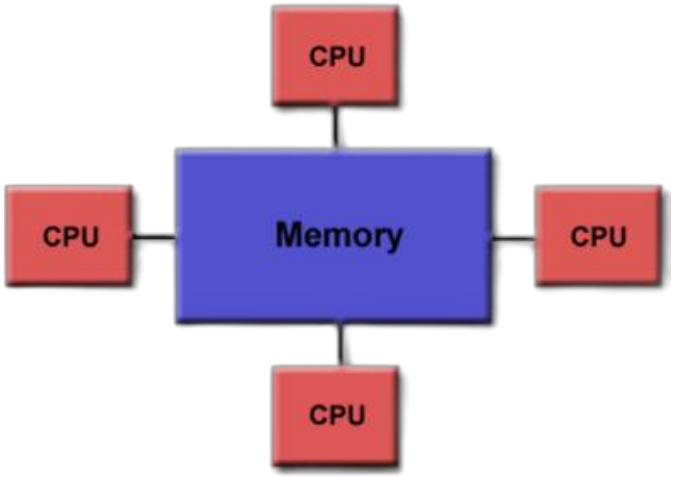


(b) 一个图像分成16块，每块都由不同的CPU分析

共享内存系统——UMA

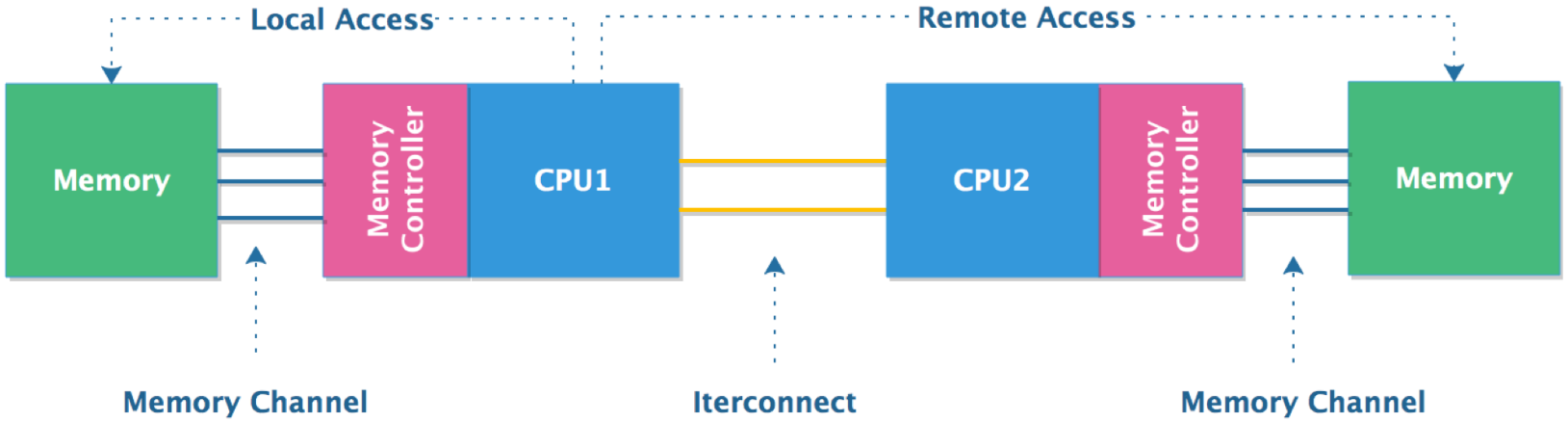
- UMA(Uniform Memory Access) 均匀存储器访问
  - 物理存储器被所有处理器均匀的共享
  - 所有处理器访问任何存储的时间相等
  - 每台处理器可带私有高速缓存
  - 外围设备可以一定形式共享





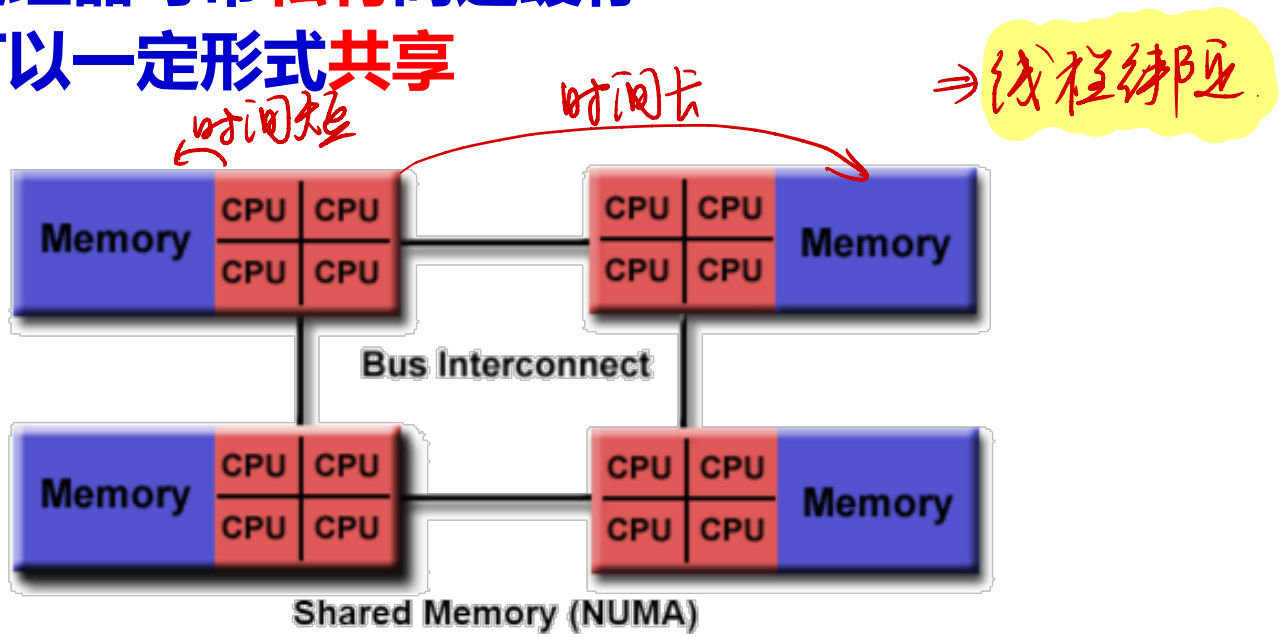
共享内存系统——NUMA

- NUMA (Non-Uniform Memory Access)非均匀存储器访问
  - 共享存储器分布在所有的处理器中
  - 处理器访问存储器的时间不均匀
  - 每台处理器可带私有高速缓存
  - 外设可以一定形式共享



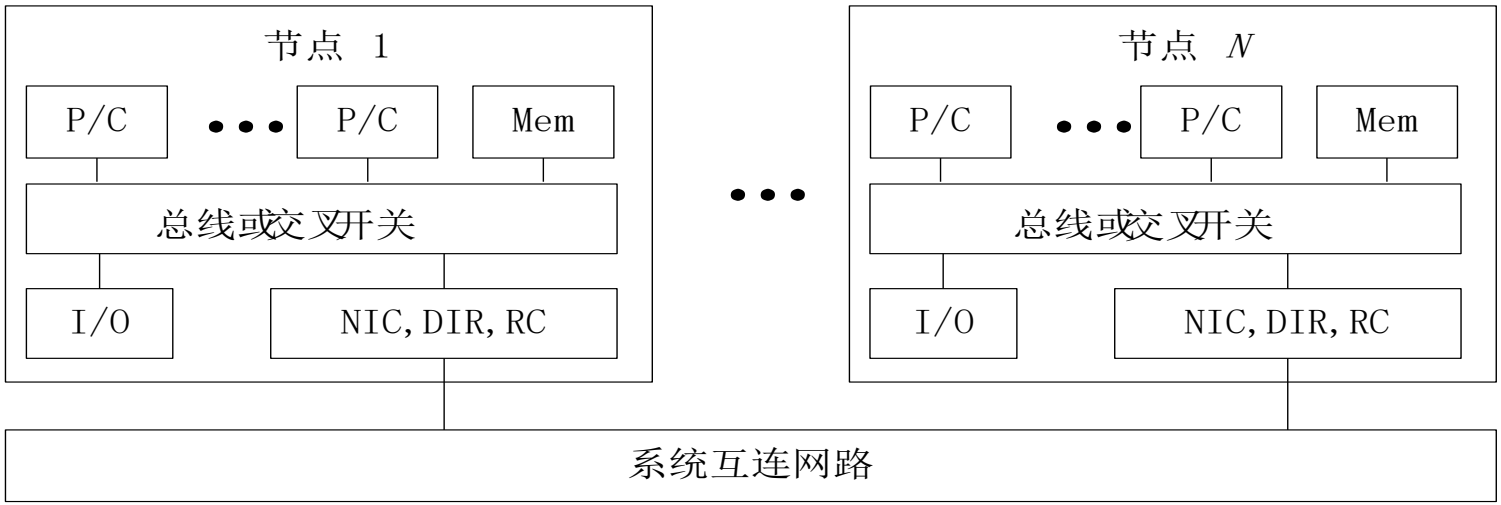
共享内存系统——NUMA

- NUMA (Non-Uniform Memory Access)非均匀存储器访问
  - 共享存储器分布在所有的处理器中
  - 处理器访问存储器的时间不均匀
  - 每台处理器可带私有高速缓存
  - 外设可以一定形式共享



共享内存系统——CC-NUMA

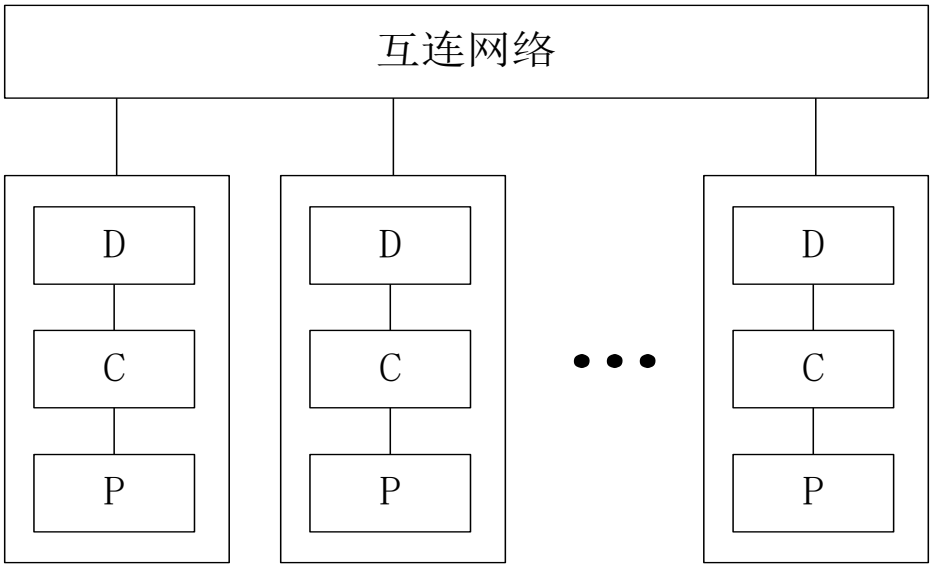
- CC-NUMA (Coherent-Cache Nonuniform Memory Access) 高速缓存一致性非均匀存储访问
  - 大多数使用基于目录的高速缓存一致性协议
  - 保留SMP易于编程的优点，改善常规SMP的可扩放性
  - 分布共享存储的DSM多处理机系统
  - 程序员无需明确地在节点上分配数据



共享内存系统——COMA

- COMA(Cache-Only Memory Access) 全高速缓存存取 没有内存!!!
  - 没有存储层次结构，全部高速缓存组成全局地址空间
  - 利用分布的高速缓存目录进行远程高速缓存的访问
  - COMA中的高速缓存容量一般大于2 级高速缓存容量
  - 数据开始时可任意分配

速度快. 价格高.



目录

- 并行系统分类
- 共享内存系统
- **分布式内存系统**
- 异构系统架构
- 互连网络

分布式内存系统

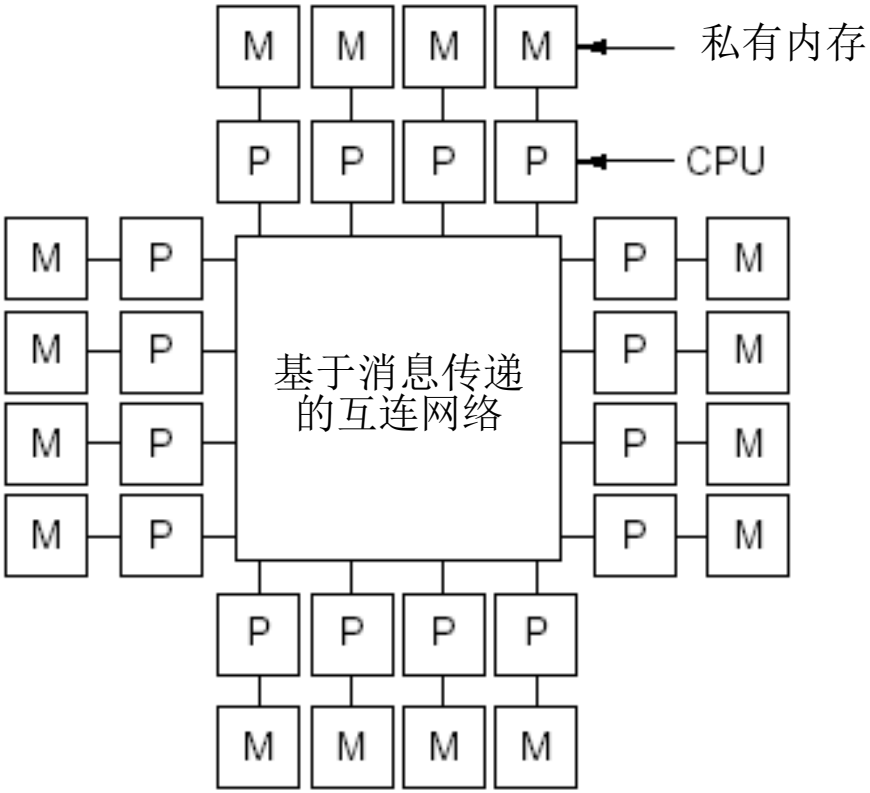
- 连接多台服务器形成一个不同享内存的计算平台

集群: 数十台服务器

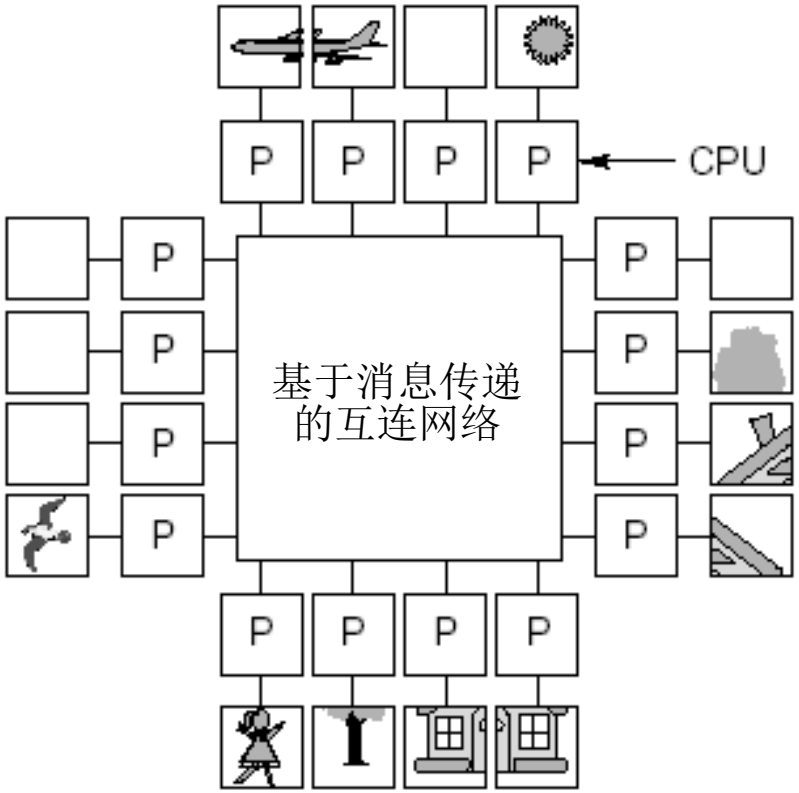
超级计算机: 数百台服务器

数据中心: 数千台服务器

分布式内存系统



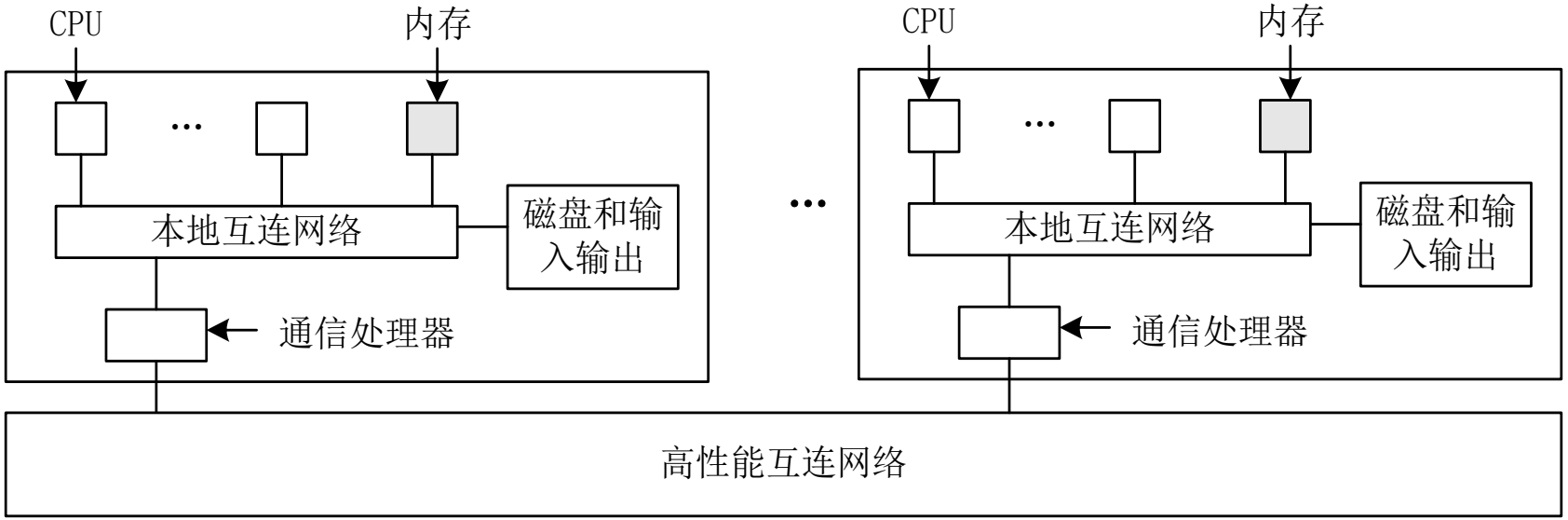
(a) 16个CPU的多计算机系统(每个CPU都有私有内存)



(b) 图13.1中的图像分布在16块内存中

分布式内存系统

- **通用多计算机体系结构**
  - **每个节点都由一个或者多个CPU、RAM、磁盘以及其它的输入/输出设备和通信处理器组成**
  - **通信处理器通过互连网络相互连接起来。可以使用多种不同的拓扑结构，交换策略和寻径算法**

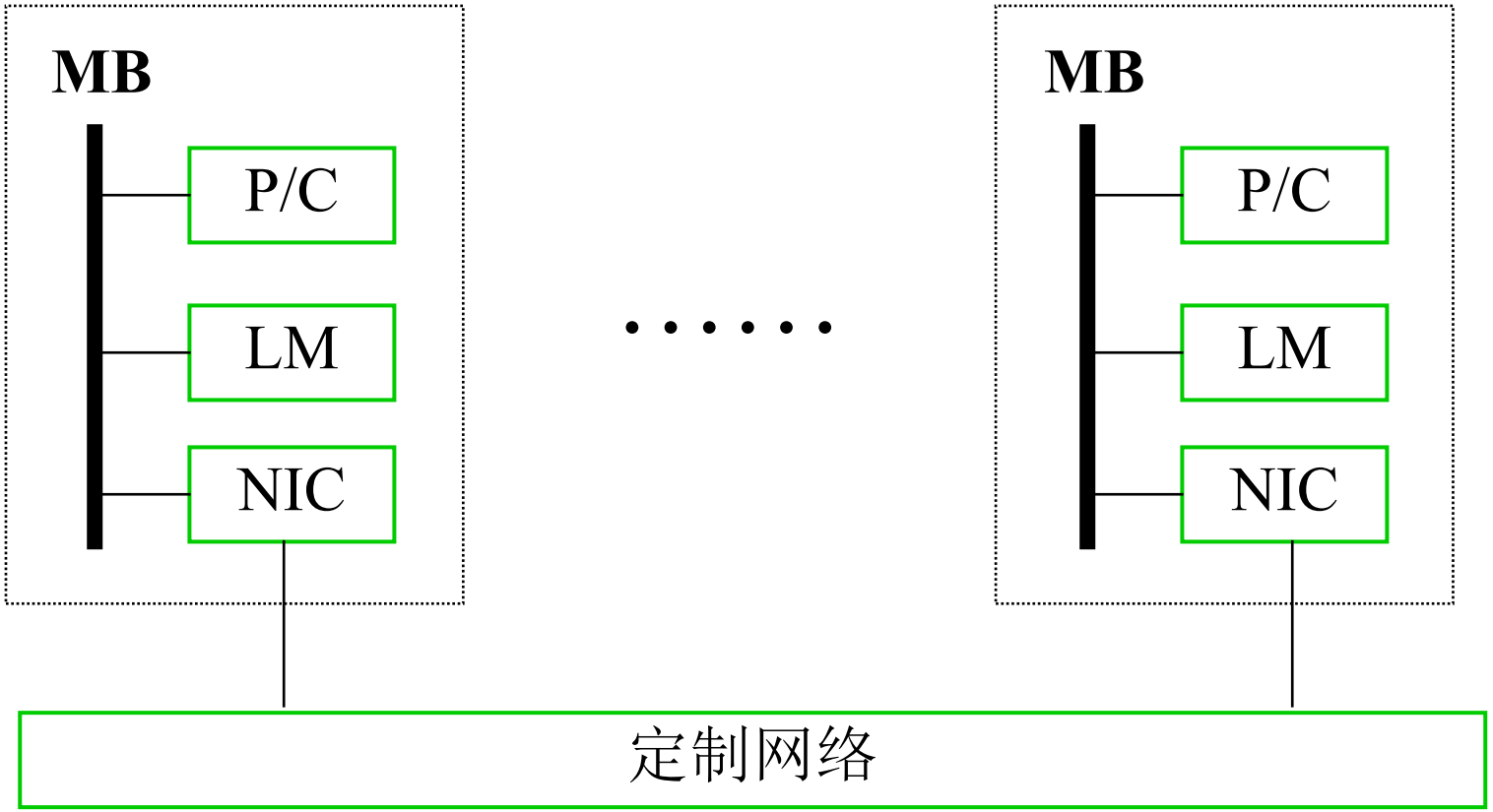


分布式内存系统——MPP

- MPP（Massively Parallel Processor）大规模并行处理机
  - 由成百上千台处理机组成的大规模并行计算机系统
  - 过去主要用于科学计算、工程模拟等以计算为主的场合，目前也广泛应用于商业和网络应用中
  - 开发困难，价格高，市场有限。是国家综合实力的象征
- 系统特点
  - 一般使用标准的商用CPU作为它们的处理器
  - 使用了高性能的私用的互连网络，可以在低延时和高带宽的条件下传递消息
  - 具有强大的输入/输出能力
  - 能够进行特殊的容错处理

分布式内存系统——MPP

LM：本地存储器
NIC：网络接口电路
MB：存储器总线

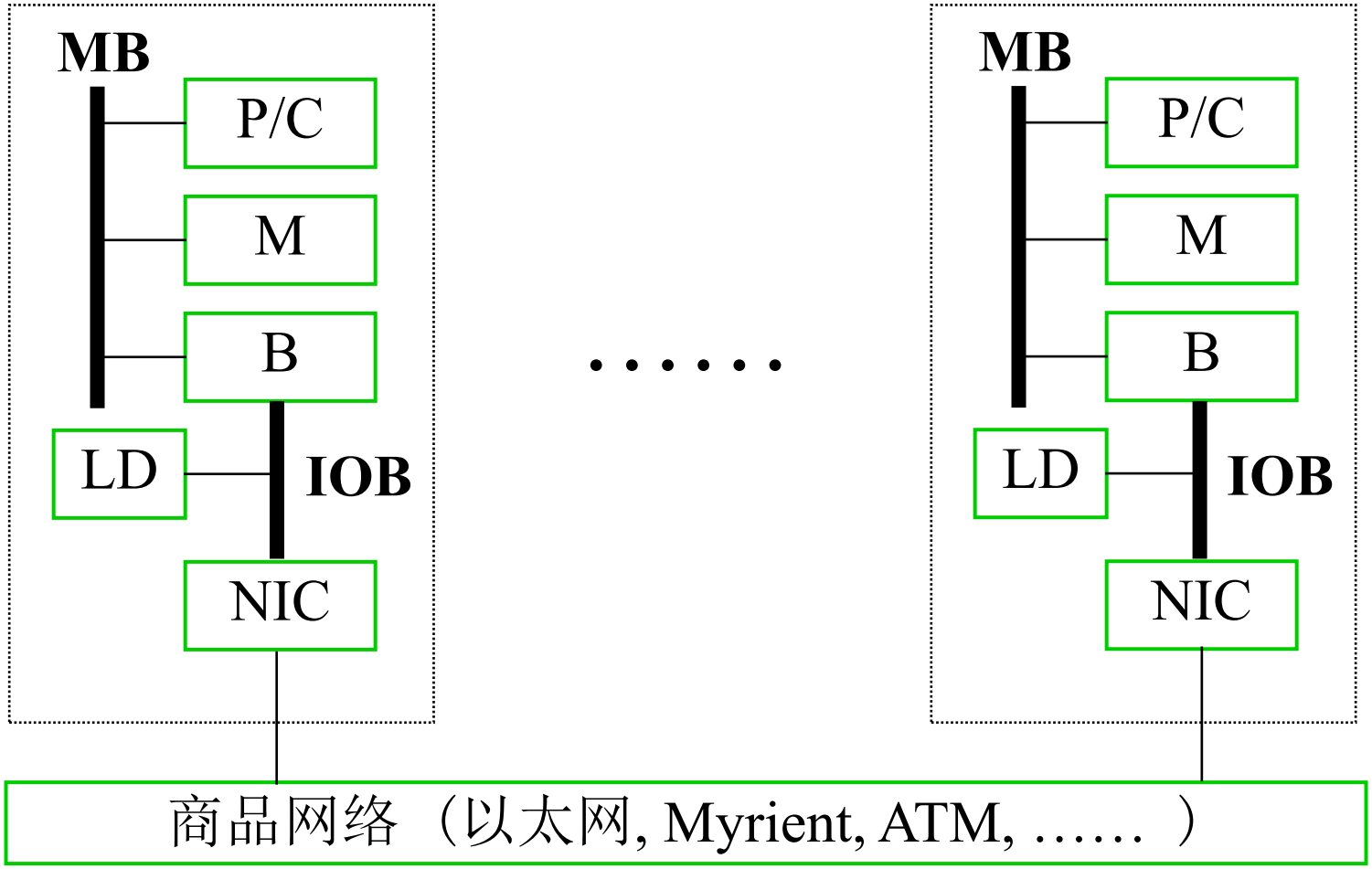


分布式内存系统——Cluster

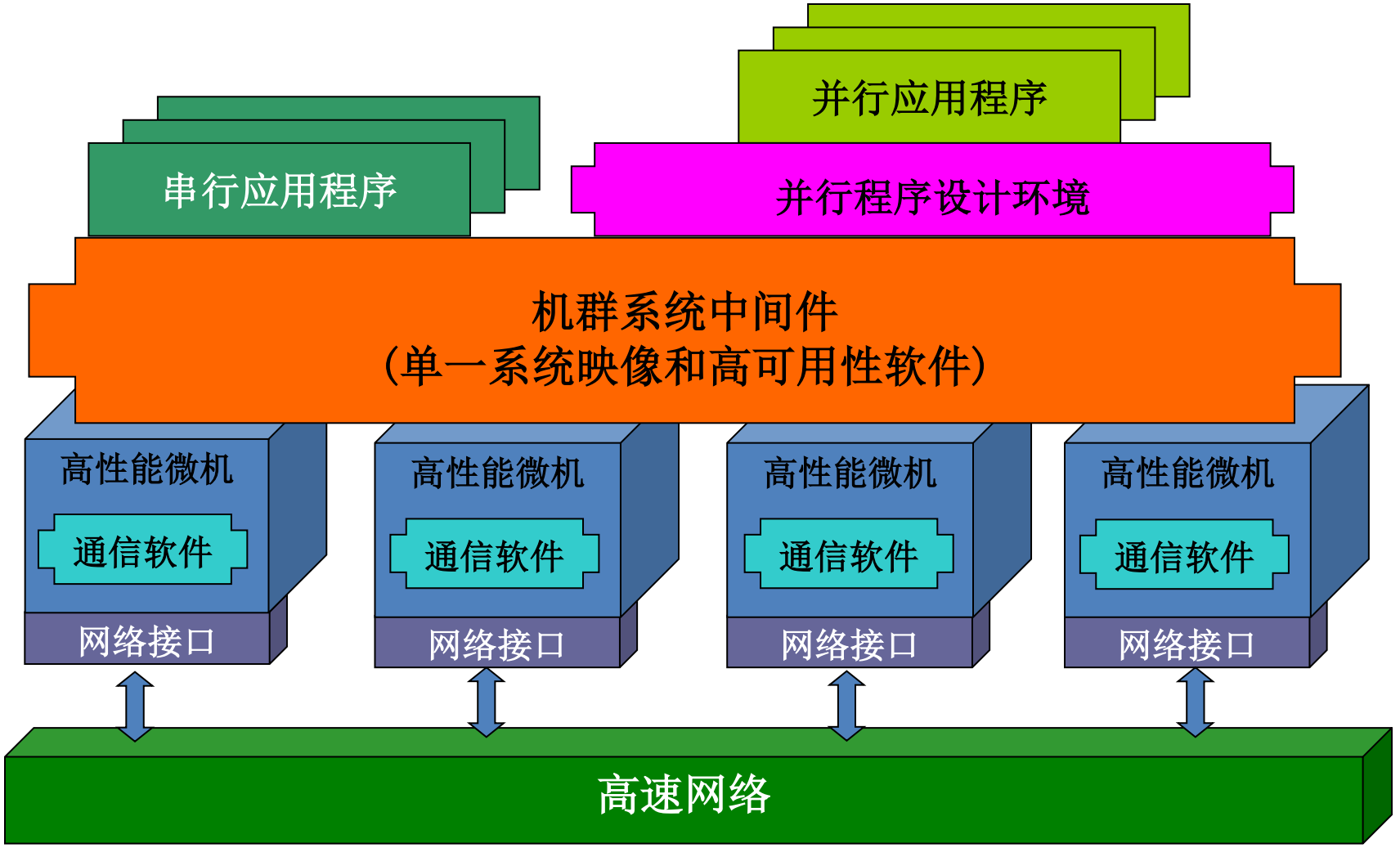
- Cluster 集群（COW机群）
  - 由大量的PC机或者工作站通过商用网络连接在一起构成
  - 可以完全使用可以买到的商用组件装配而成，这些商用组件都是大规模生产的产品，能够获得较高的性价比
- 与MPP的区别（体系结构方面）
  - 节点是更完整的计算机，计算机可以是同构或异构
  - 节点都有自己的磁盘，驻留有自己的完整的操作系统；而MPP系统结点一般没有磁盘，只驻留操作系统内核
  - MPP使用制造厂商专有的高速通信网络；COW一般采用公开销售的标准高速局域网或系统域网，网络通常是与节点计算机的I/O总线相连（松散耦合），而MPP的网络接口是连到处理节点的存储总线上（紧耦合）

分布式内存系统——Cluster

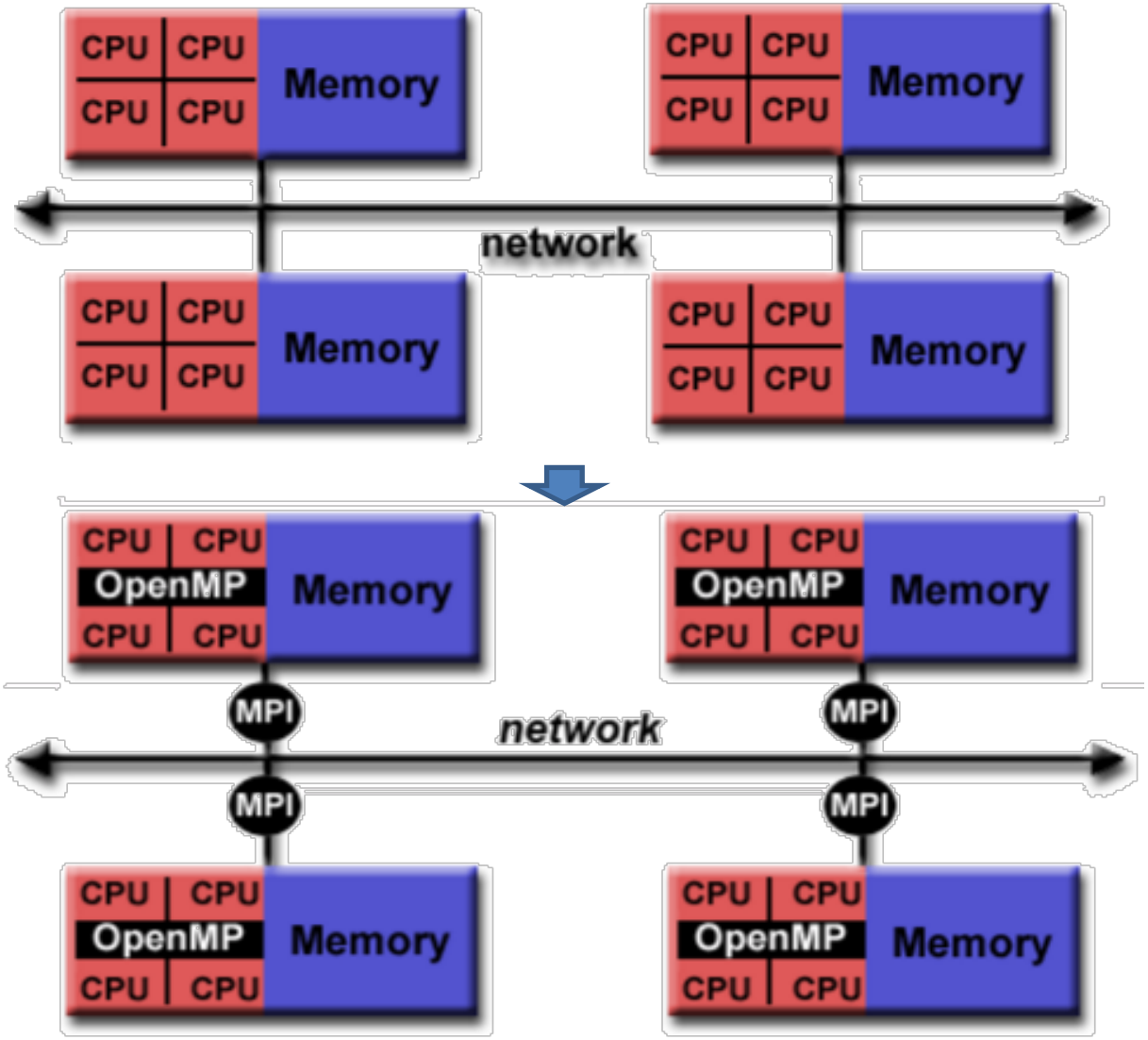
B：存储总线与I./O总线的接口
LD：本地磁盘
IOB：I/O总线



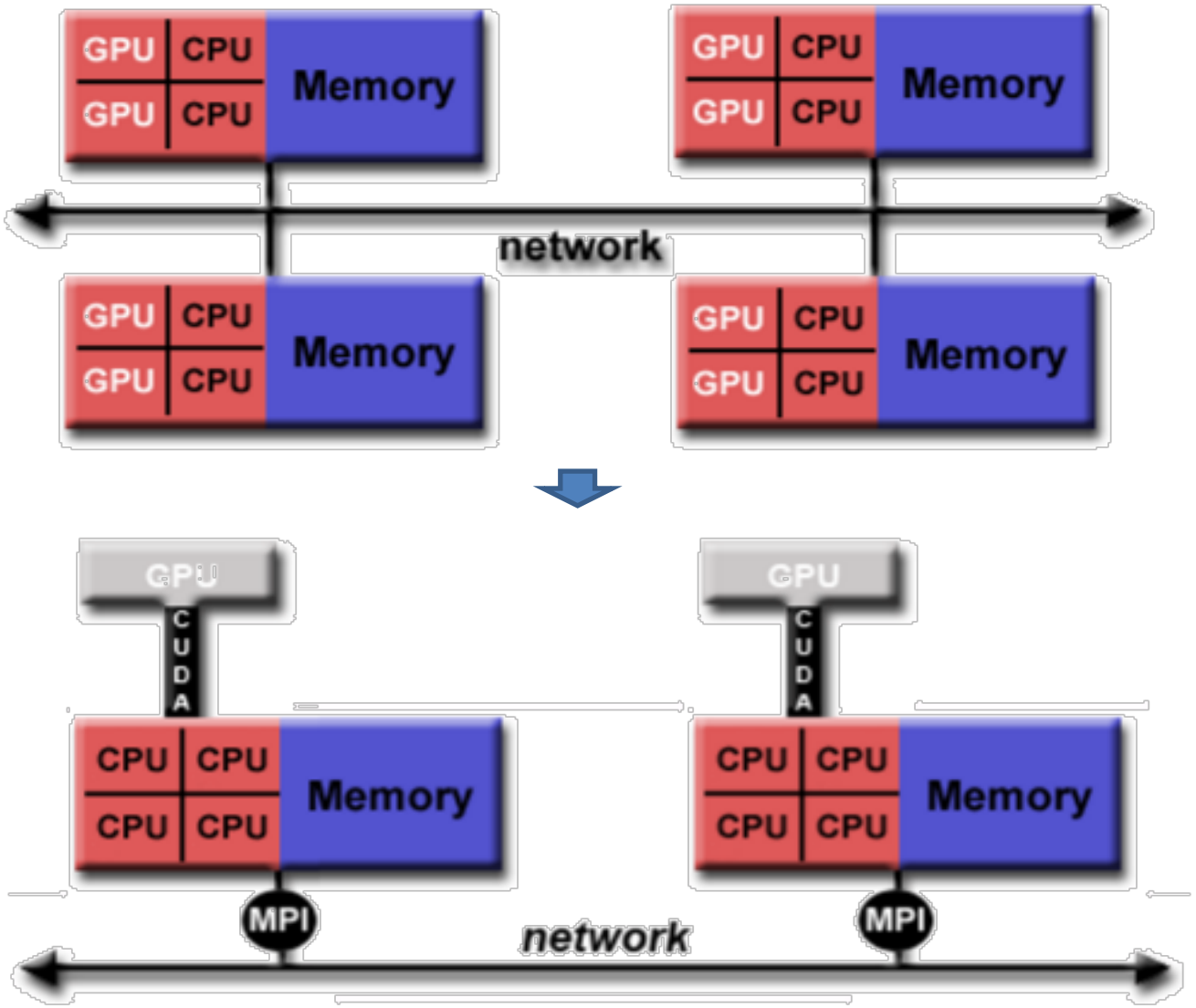
分布式内存系统——Cluster



分布式内存系统



分布式内存系统

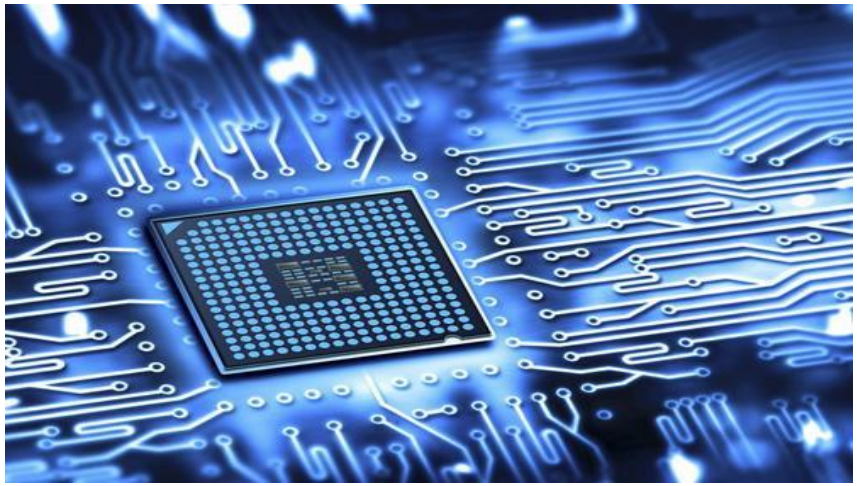


目录

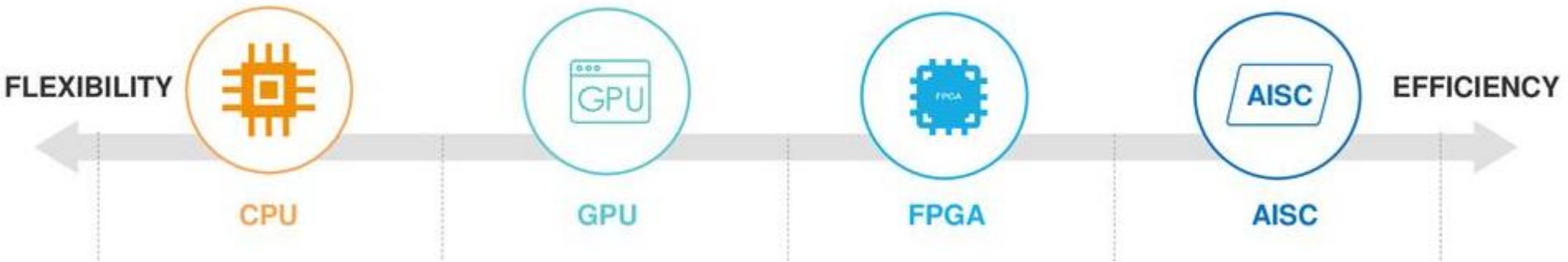
- 并行系统分类
- 共享内存系统
- 分布式内存系统
- **异构系统架构**
- 互连网络

异构系统架构

- 多种处理器或核心
- 使用加速器来提高性能或能源效率，通常结合专门的处理能力来处理特定任务
- 适合执行single instruction, multiple data (SIMD) 和 single instruction, multiple threads (SIMT)



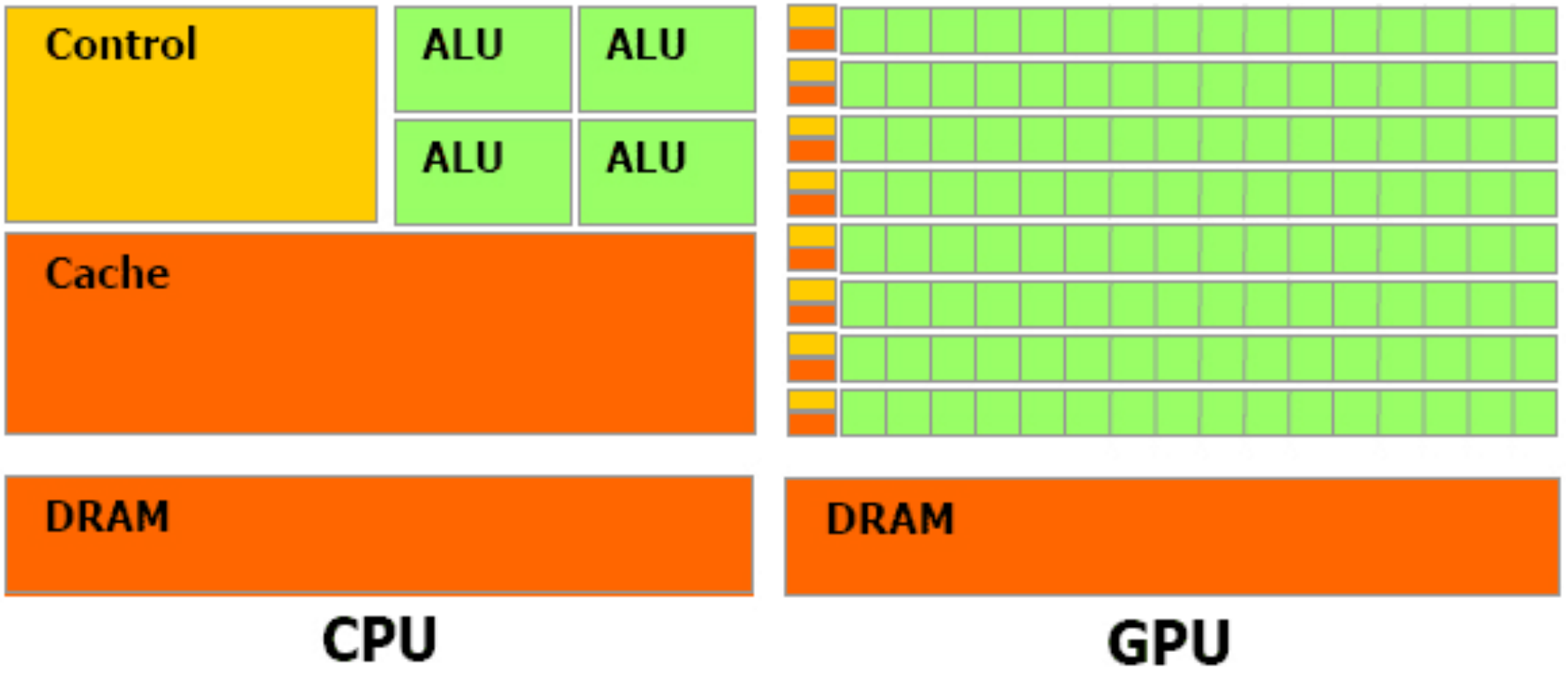
异构系统架构



- CPU
  - 通用型，复杂计算
  - 灵活、易用、通用
  - 性能较低
- GPU
  - 批量数据并行计算
  - 高性能
  - 高功耗
- FPGA
  - 不规则数据并行计算
  - 性能好
  - 能效比高
  - 灵活
- AISC
  - 适用数据并行计算
  - 高性能
  - 低功耗
  - 专用电路，不可修改

异构系统架构

- NVIDIA GPU， CPU和GPU比较



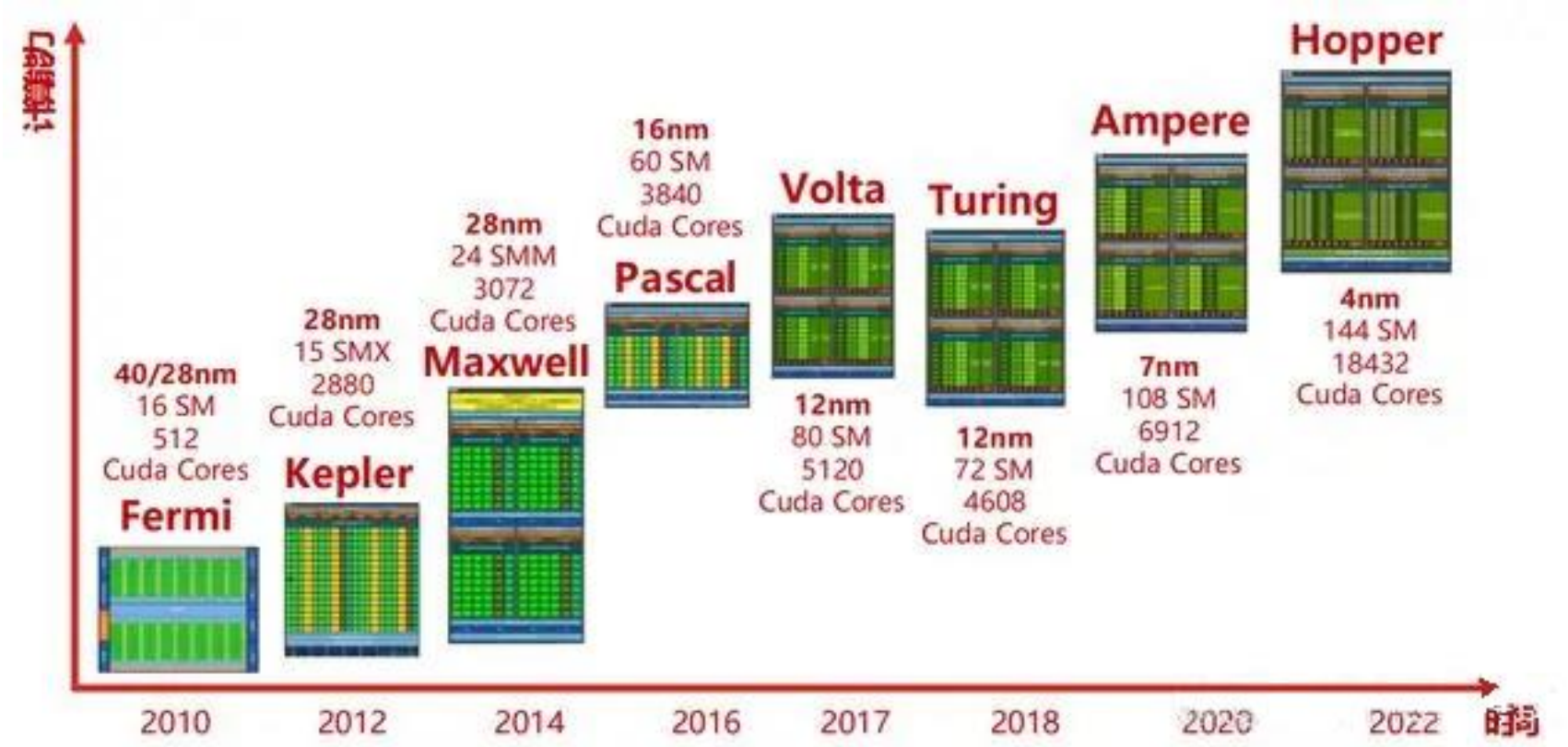
异构系统架构

- **NVIDIA GPU发展**

Micro-architecture	Launch	GPU
Tesla	2008年	S1070
Fermi	2010年	M2090、S2070
Kepler	2012年	K40/K80
Maxwell	2014年	GeForce GTX 750Ti、GeForce GTX TITAN X
Pascal	2016年	Tesla P40、GTX 1080TI/Titan XP、Quadro GP100/P6000/P5000
Voltal	2017年	Tesla V100、GeForceTITan V
Turing	2018年	RTX 2080Ti /2080/2070 QuadroRTX6000
Ampere	2020年	A100、RTX 3080Ti/3080/3070/3060、RTX 3090
Hopper	2022年	H100
Ada Lovelace	2022年	L40、 L4、 RTX 40 系列

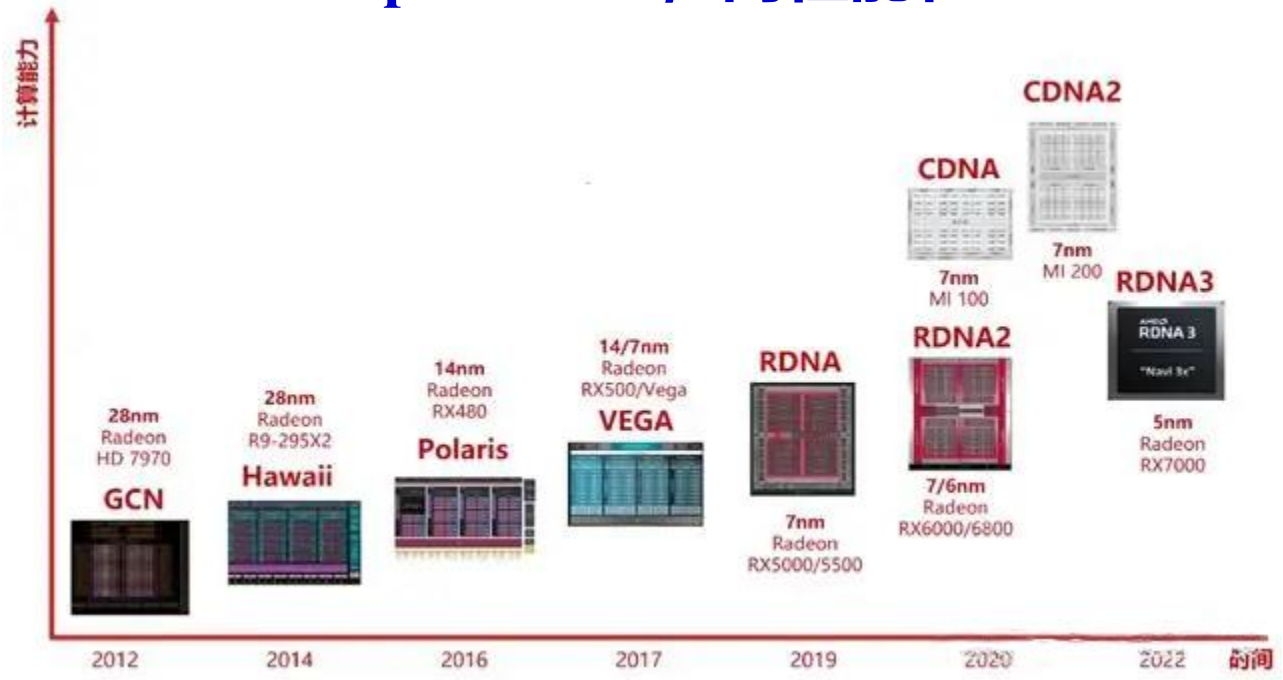
异构系统架构

- NVIDIA GPU发展



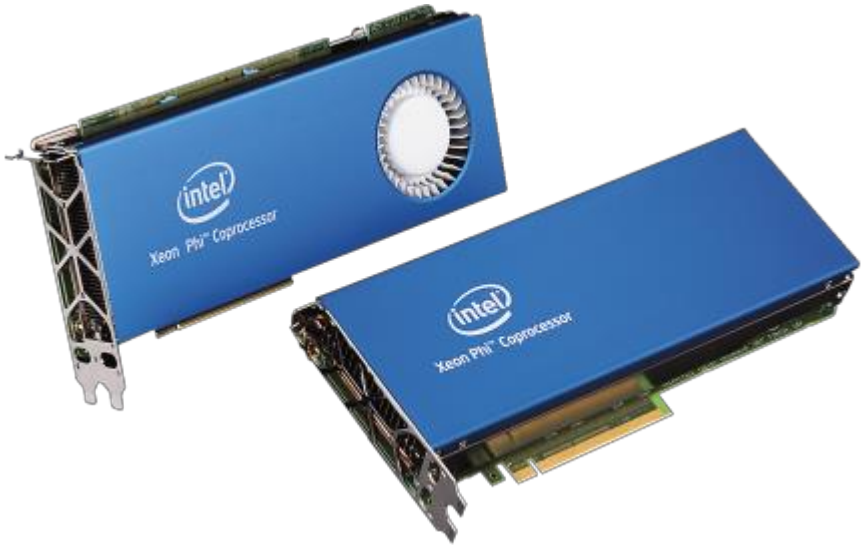
异构系统架构

- AMD GPU 架构:
  - TeraScale
  - GCN： Graphics Core Next
  - RDNA：Radeon DNA， 游戏
  - CDNA：Compute DNA，高性能和AI



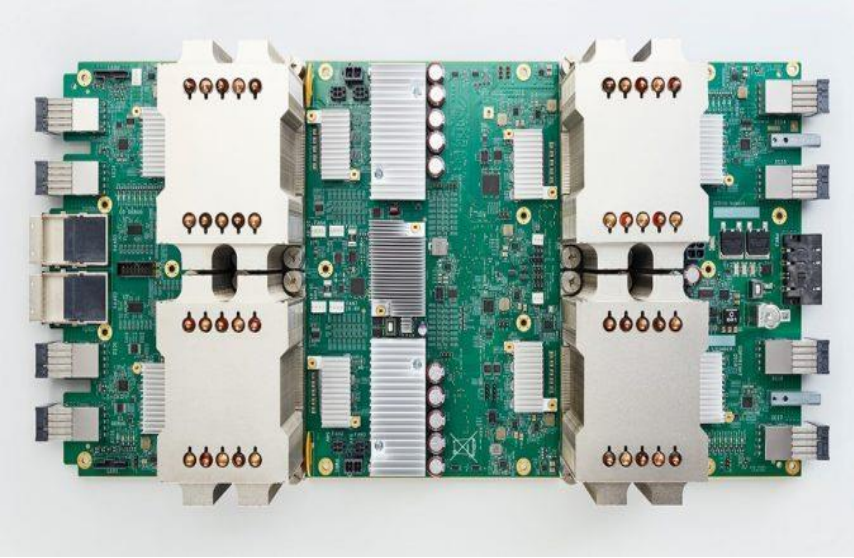
异构系统架构

- Intel MIC （Many Integrated Core）
  - 由许多x86核集成在一起的加速设备
  - 核心设计与奔腾处理器相似
  - 添加64位硬件支持，更多的硬件线程（每四个核4个硬件线程），512位 的SIMD向量支持



异构系统架构

- TPU （Tensor Processing Unit）
  - 人工智能加速器专用集成电路（ASIC）
  - 谷歌开发
  - 主要任务是矩阵处理，乘法和累加运算的组合
  - TPU 包含数千个乘法累加器，这些累加器彼此直接连接以形成大型物理矩阵



异构系统架构

- TPU 架构

	TPUv1	TPUv2	TPUv3	TPUv4
Date Introduced	2016	2017	2018	2021
Process Node	8 nm	16 nm	16 nm	7 nm
Die Size (mm2)	331	< 625	< 700	< 400
On chip memory (MiB)	28	32	32	144
Clock Speed (MHz)	700	700	940	1050
Memory (GB)	8GB DDR3	16GB HBM	32GB HBM	8GB
TDP(W)	75	280	450	175

异构系统架构

- FPGA （Field Programmable Gate Array）
  - 专用集成电路中的一种半定制电路，是可编程的逻辑列阵
  - 低功耗和高能量效率
  - 重新编程以加速不同的应用
  - 采用硬件的方式来实现逻辑和算法，不需发射和解析指令
  - 针对需求设计多种计算部件来同时实现数据并行和流水线并行
  - 实例：微软 Catapult

异构系统架构

- CPU+FPGA异构架构种类：
  - FPGA 作为外部独立的计算模块，通过网络、数据总线、I/O 接口等机制与处理器进行连接；
  - FPGA 作为共享内存的计算模块，被用于可重构计算器件置于Cache高速缓存和内存之间；
  - FPGA 作为协处理器，与 CPU共享缓存；
  - FPGA 集成处理器架构，将处理器高度嵌入到FPGA可编程器件中，实现了CPU与FPGA的紧耦合。随着FPGA处理单元与CPU之间耦合度的增加，通信代价逐渐降低，系统设计的复杂度也相应提高。

目录

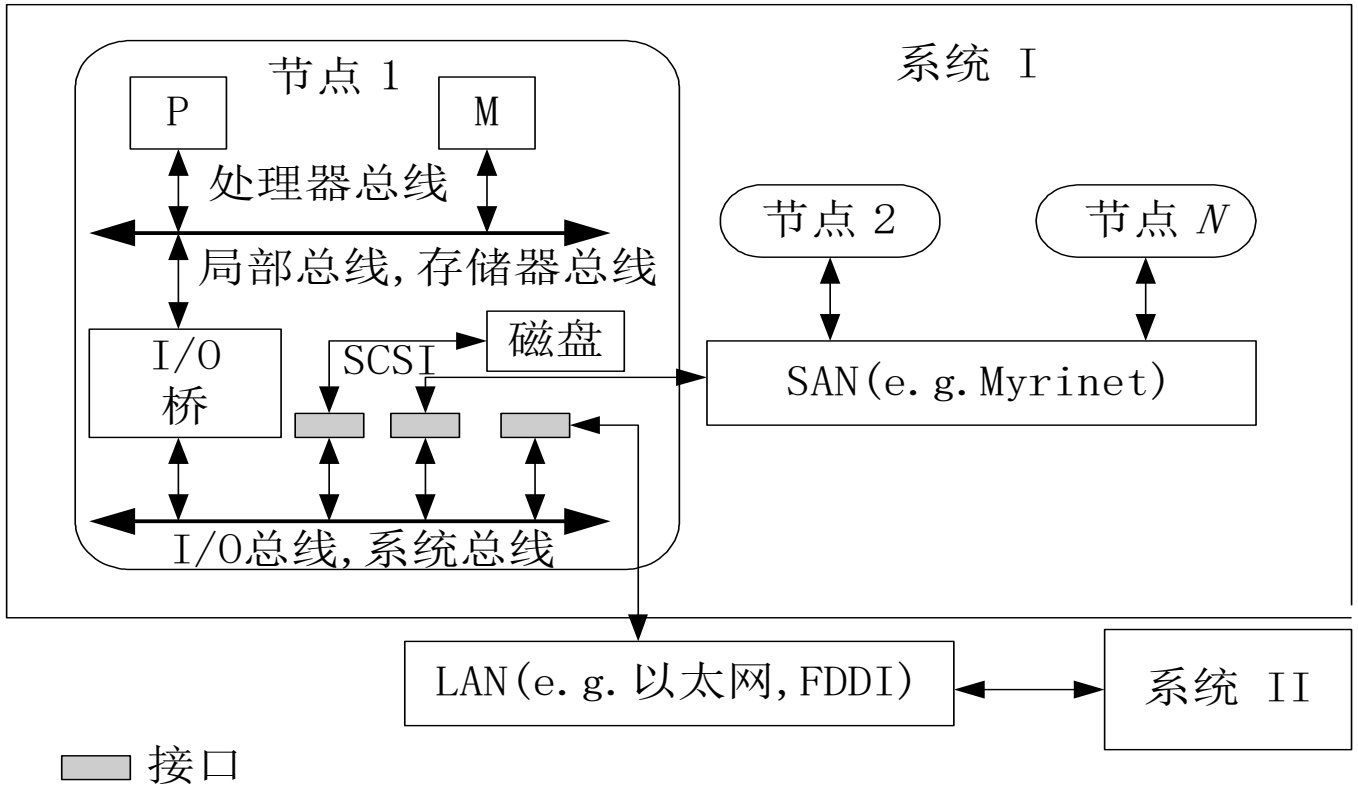
- 并行系统分类
- 共享内存系统
- 分布式内存系统
- 异构系统架构
- **互连网络**

互连网络概述

- 并行计算机的通信体系结构是系统核心
  - 两个层次：底层的互联网络；上层的语言、软件工具包、编译器、操作系统等提供的通信支持
- 互连网络是并行计算机系统内部的互连网络
  - 定义：由开关元件按一定拓扑结构和控制方式构成的网络以实现计算机系统内部多个处理机或多个功能部件间的相互连接
  - 与计算机网络在工作原理、概念以及术语上有许多相同或相似之处；并且某些并行计算机系统中的互连网络就是高速以太网和ATM网络
- 互连网络一般由以下五个部分组成
  - CPU、内存模块、接口、链路和交换节点

互连网络

- **局部总线、I/O总线、SAN和LAN**



互连网络拓扑结构

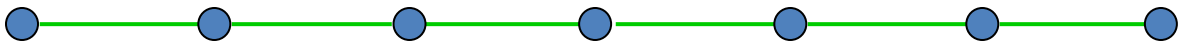
- 互连网络的拓扑结构描述了链路和交换节点是如何组织安排的。拓扑结构可以用图来表示，链路用边表示，交换节点用节点表示
- 静态网络
  - 静态网络(Static Networks)是指节点间有着固定连接通路且在程序执行期间，这种连接保持不变的网络
- 动态网络
  - 动态网络(Dynamic Networks)由开关单元构成，可按应用程序的要求动态地改变连接状态。如总线、交叉开关，多级交换网络等

互连网络参数

- **节点度**（Node Degree）：与节点相连接的边数，表示节点所需要的端口数，根据链路到节点的方向，节点度可以进一步表示为：**节点度 = 入度 + 出度**，其中**入度**是进入节点的链路数，**出度**是从节点出来的链路数
- **网络直径**（Network Diameter）：网络中任何两个节点之间的最长距离，即最大路径数
- **对剖宽度**（Bisection Width）：对分网络各半所必须移去的最少边数
- **对剖带宽**（Bisection Bandwidth）:每秒钟内，在最小的对剖平面上通过所有连线的最大信息位（或字节）数
- 如果从任一节点观看网络都一样，则称网络为**对称的**（Symmetry）

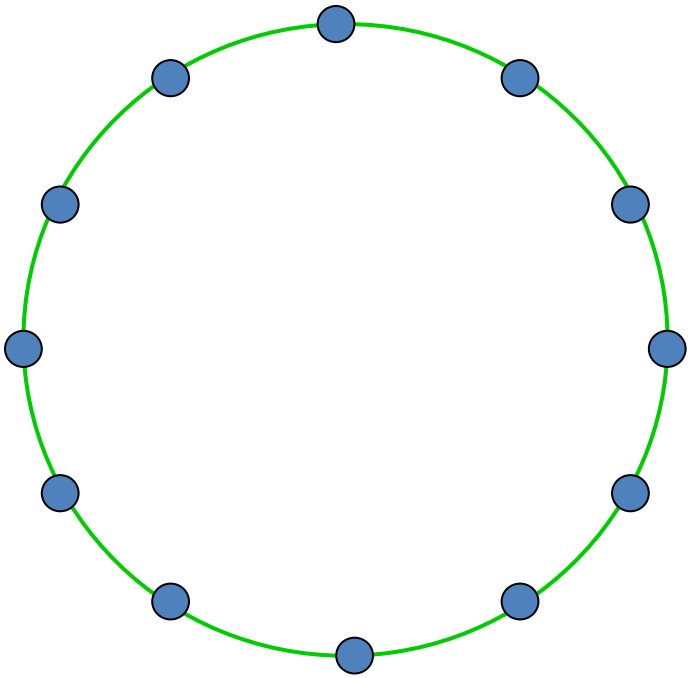
静态互连网络

- **一维线性阵列（1-D Linear Array）**
  - 并行机中最简单、最基本的互连方式
  - 每个节点仅与其左右近邻相连，也叫二近邻连接
  - 对*N*个结点的线性阵列，有*N*-1条链路，直径为*N*-1（任意两点之间距离的最大值）度为2不对称，对剖宽度为1。*N*很大时，通信效率很低



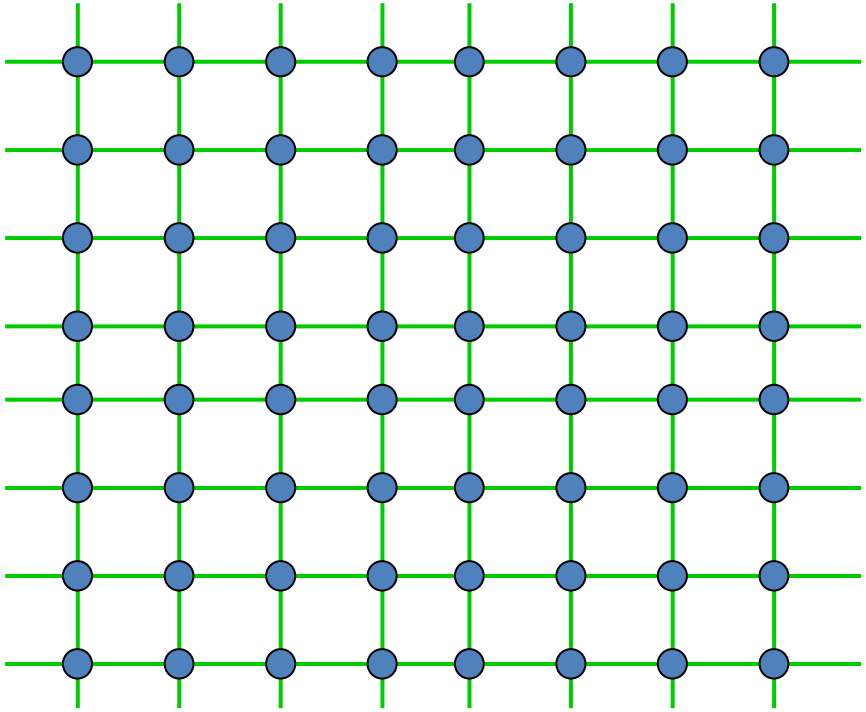
静态互连网络

- **环形**
  - 对N个节点的环，考虑相邻节点数据传送方向：
  - 双向环：链路数为N，直径$\lfloor N/2 \rfloor$，度为2，对称，对剖宽度为2
  - 单向环：链路数为N，直径N-1，度为2，对称，对剖宽度为2



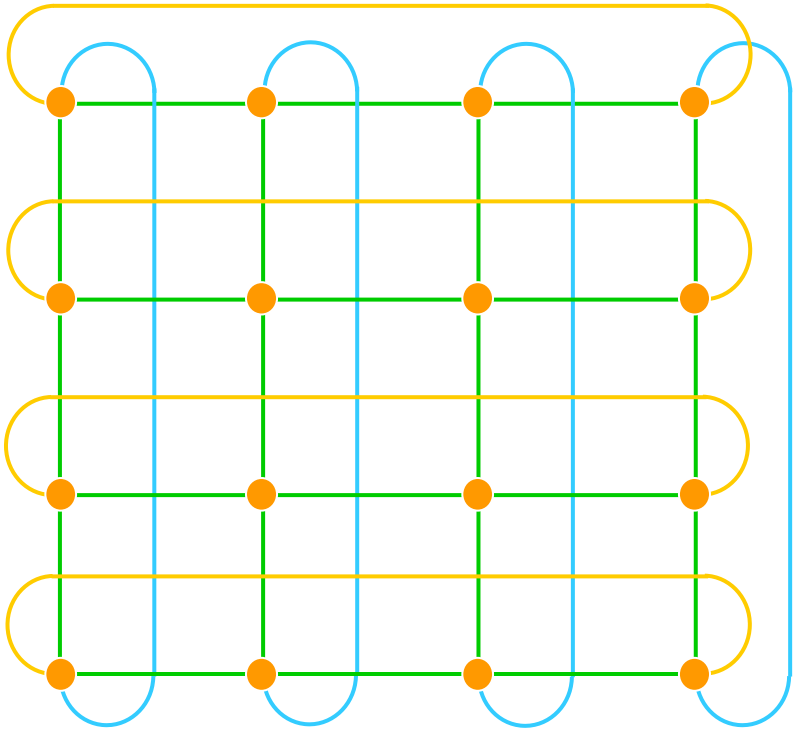
静态互连网络

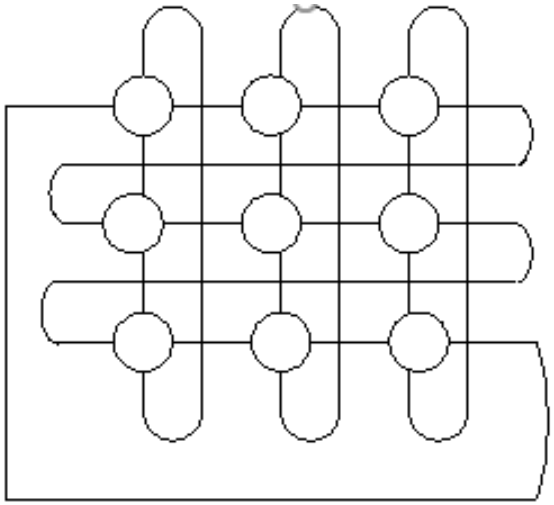
- **二维网孔（2-D Mesh）**
  - N个节点的 $\sqrt{N} \times \sqrt{N}$ 网格
  - 每个节点只与其上、下、左、右的近邻相连（边界节点除外）
  - 链路数量为 $2\left(N-\sqrt{N}\right)$，节点度为4，网络直径为 $2\left(\sqrt{N}-1\right)$，非对称，对剖宽度为$\sqrt{N}$



静态互连网络

- **二维环绕（2-D Torus）**
  - N个节点的 $\sqrt{N} \times \sqrt{N}$ 网格
  - 垂直和水平方向带环绕
  - 链路数量为$2N$，节点度恒为4，网络直径 $2\lfloor \sqrt{N}/2 \rfloor$，对称，对剖宽度为$2\sqrt{N}$



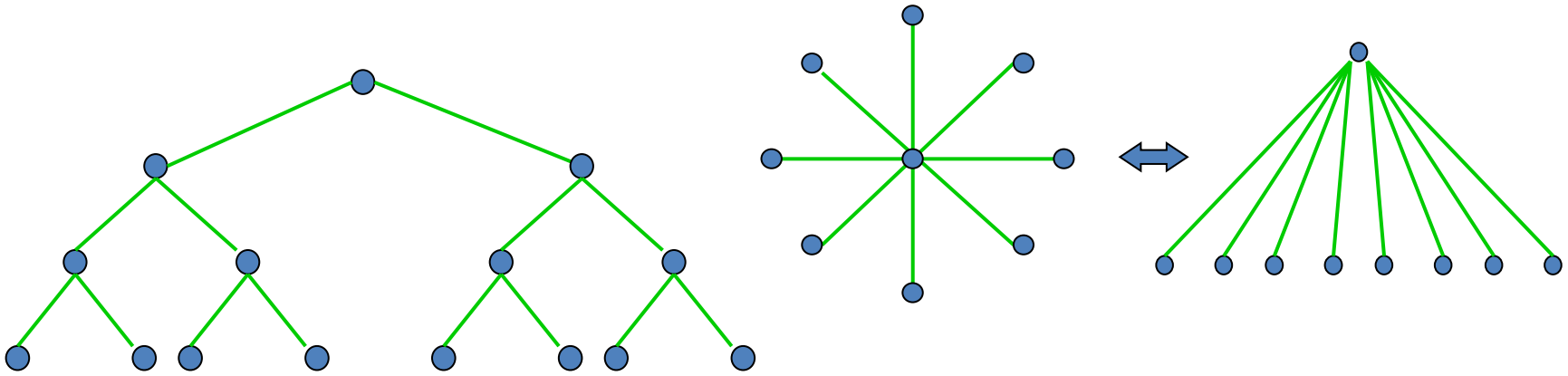


垂直方向带环绕，水平方向呈蛇状，网络直径 $\sqrt{N}-1$

Illiac网孔

静态互连网络

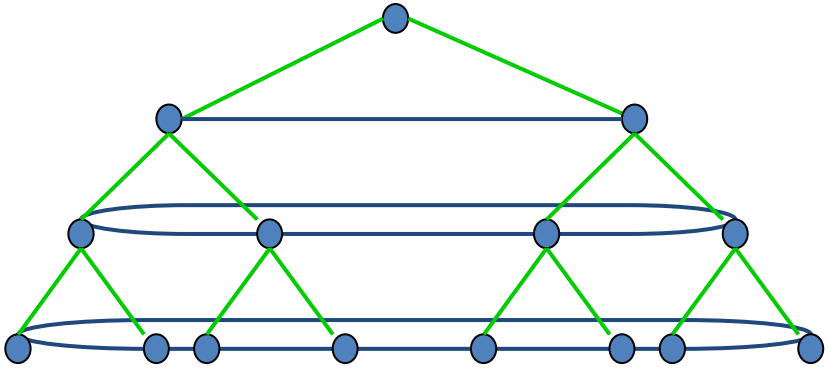
- **二叉树**
  - 除了根、叶节点，每个内节点只与其父节点和两个子节点相连
  - 节点度为3，对剖宽度为1，直径为 $2(\lceil \log N \rceil - 1)$
  - 尽量增大节点度为N-1，直径缩小为2，此时就变成星形网络，其对剖宽度为 $\lfloor N/2 \rfloor$



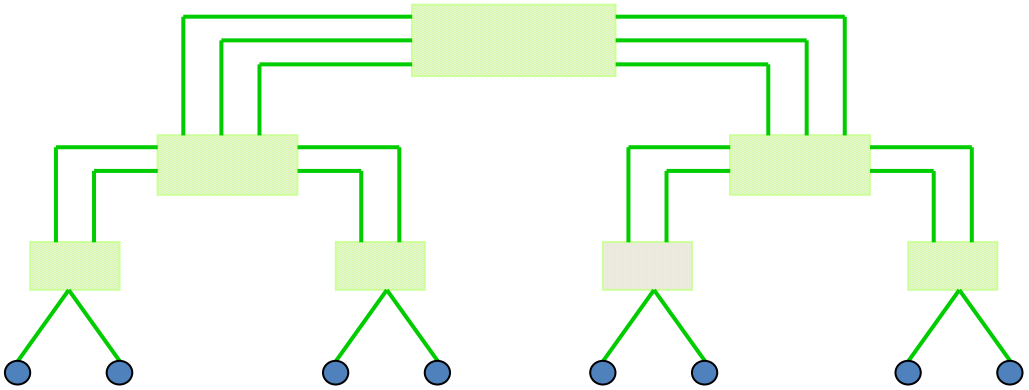
静态互连网络

- **二叉树**
  - **传统二叉树的主要问题是，根节点易成为通信瓶颈**
  - **这两种结构都可以缓解根节点的瓶颈问题**

带环树

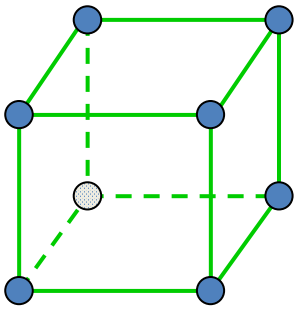


二叉胖树

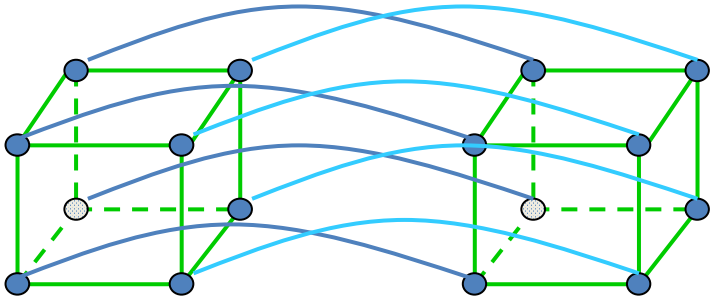


静态互连网络

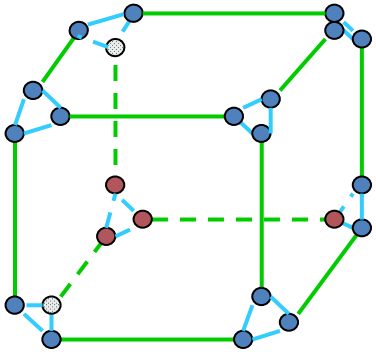
- **超立方**
  - 一个n-立方由 $N = 2^n$ 个顶点组成，4-立方是由3-立方的对应顶点连接而成
  - n-立方的节点度为n，网络直径为n，对剖宽度为$N/2$
  - 如果将n-立方的每个顶点代之以一个n个节点的环，节点总数 $n2^n$，就构成了n-立方环，每个顶点度为3



3-立方体



4-立方体



3-立方环

静态互连网络

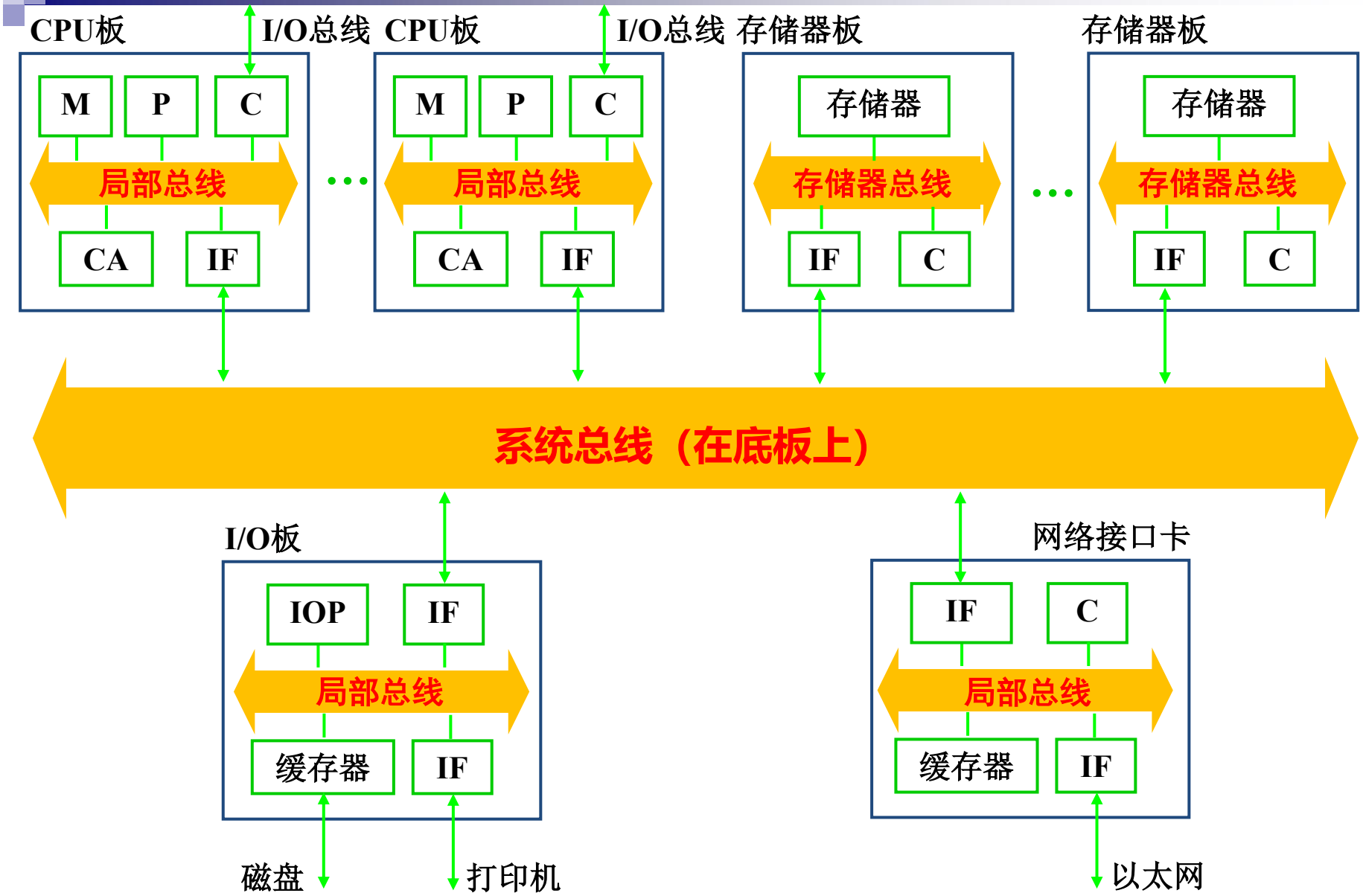
网络名称	网络规模	节点度	网络直径	对剖宽度	对称	链路数
线性阵列	N	2	$N-1$	1	非	$N-1$
环形	N	2	$\lfloor N/2 \rfloor$(双向)	2	是	N
2-D 网孔	$(\sqrt{N} \times \sqrt{N})$	4	$2(\sqrt{N}-1)$	$\sqrt{N}$	非	$2(N-\sqrt{N})$
Illiac 网孔	$(\sqrt{N} \times \sqrt{N})$	4	$(\sqrt{N}-1)$	$2\sqrt{N}$	非	$2N$
2-D 环绕	$(\sqrt{N} \times \sqrt{N})$	4	$2\lfloor \sqrt{N}/2 \rfloor$	$2\sqrt{N}$	是	$2N$
二叉树	N	3	$2(\lceil \log N \rceil - 1)$	1	非	$N-1$
星形	N	N-1	2	$\lfloor N/2 \rfloor$	非	$N-1$
超立方	$N = 2^n$	n	n	$N/2$	是	$n \cdot N/2$
立方环	$N = k \cdot 2^k$	3	$2k-1+\lfloor k/2 \rfloor$	$N/(2k)$	是	$3N/2$

动态互连网络

- **网络特点**
  - 动态网络中的连接不固定，在程序执行过程中可根据需要改变
  - 网络的开关元件，链路可通过设置这些开关的状态来重构
  - 动态网络主要有**总线、交叉开关、多级交换网络**

动态互连网络

- **总线**
  - 总线实际上是连接处理器、存储器和I/O等外围设备的一组导线和插座
  - 它在某一时刻只能用于一对源和目的之间传输数据。
  - 当有多对源和目的请求使用总线时，要进行总线仲裁。当CPU数目较多时对总线争用严重（<=32个）



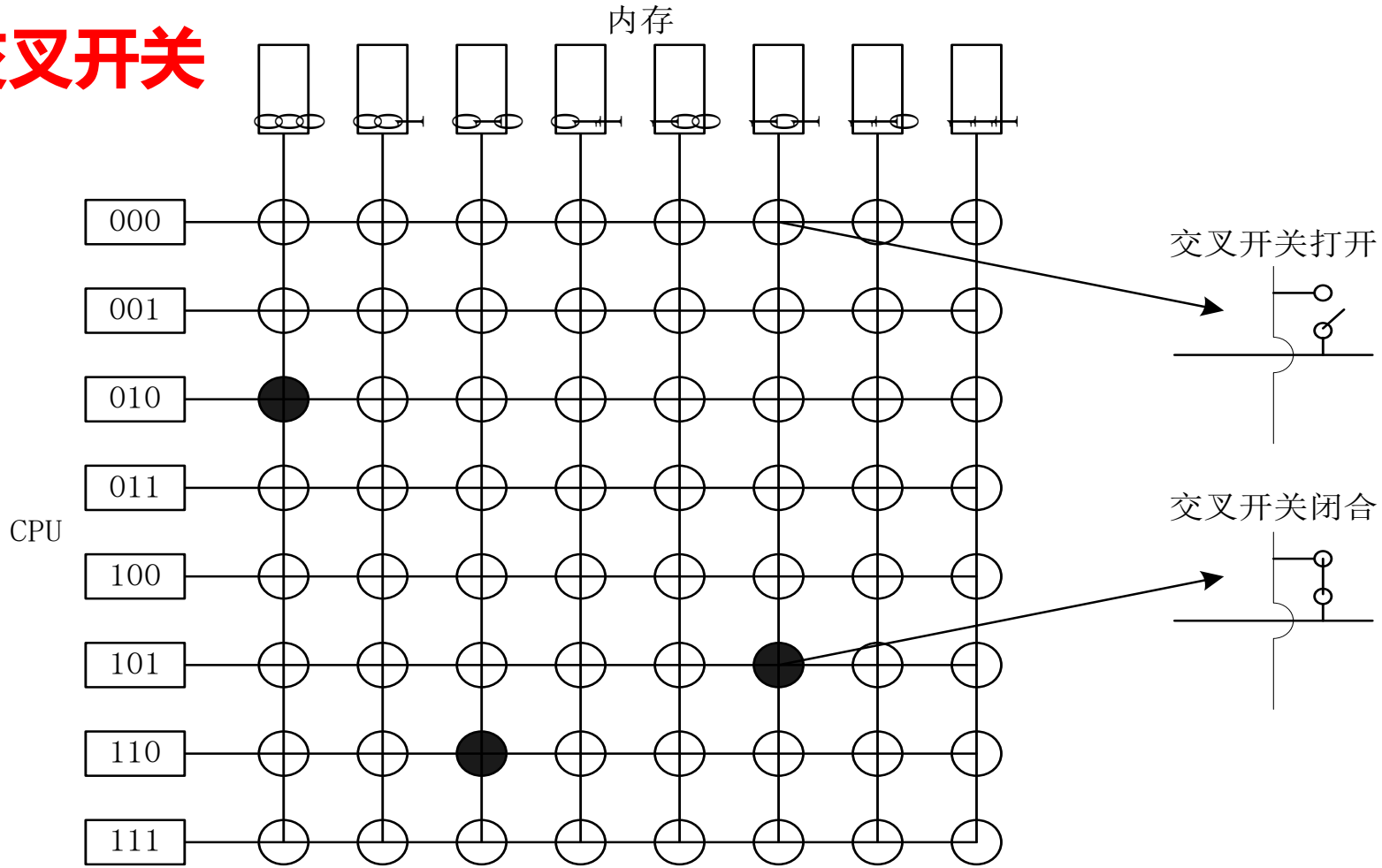
IF:专用逻辑接口 C:专用控制器 P:处理器 M:局部存储器 CA:高速缓存 IOP:I/O处理器

动态互连网络

- **交叉开关**
  - 单级交换网络，可为每个端口提供更高的带宽。像电话交换机一样，交叉点开关可由程序控制动态设置其处于“开”或“关”状态，从而能提供所有（源、目的）对之间的动态连接
  - 交叉开关一般有两种使用方式：一种是用于对称的多处理机或多计算机机群中的处理器间的通信；另一种是用于SMP服务器或向量超级计算机中处理器和存储器之间的存取

动态互连网络

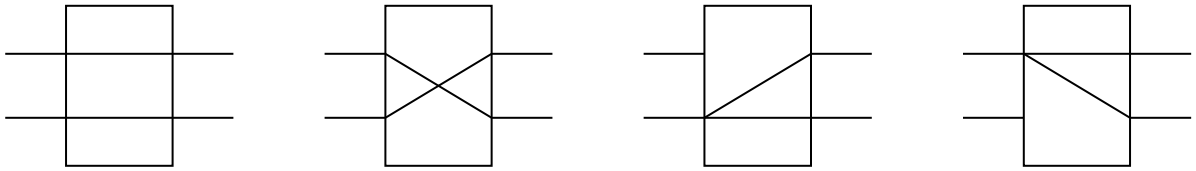
- **交叉开关**



8×8的交叉开关

动态互连网络

- **多级交换网络**
  - MIN（Multistage Interconnection Network）：单级交叉开关级联起来形成多级互连网络
  - 交换开关模块：一个交换开关模块有n个输入和n个输出，每个输入可连接到任意输出端口，但只允许一对一或一对多的映射
  - 级间互联模式：均匀洗牌、蝶式、多路洗牌、交叉开关及立方体连结等



动态互连网络

- **多级交换网络**