



高性能计算与云计算

第二讲 并行计算机体系结构

董守斌

sbdong@scut.edu.cn

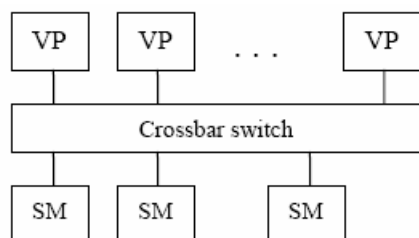
华南理工大学计算机学院
广东省计算机网络重点实验室

Communication & Computer Network Laboratory (CCNL)

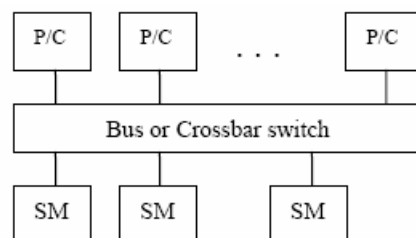
复习：并行计算机体系结构

- **Single Instruction Stream Over Multiple Data Streams (SIMD)**
- **Parallel Vector Processor (PVP)**
- **Symmetric Multiprocessors (SMP)**
- **Massively Parallel Processors (MPP)**
- **Distributed Shared Memory (DSM)**
- **Cluster of Workstation (COW)**

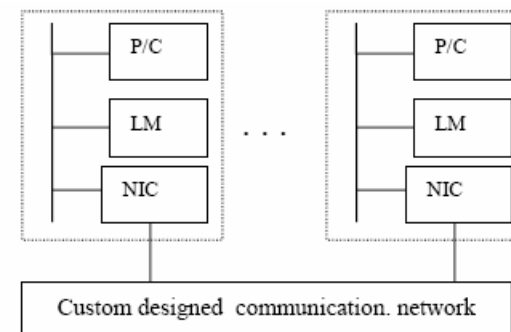
复习：并行计算机体系结构



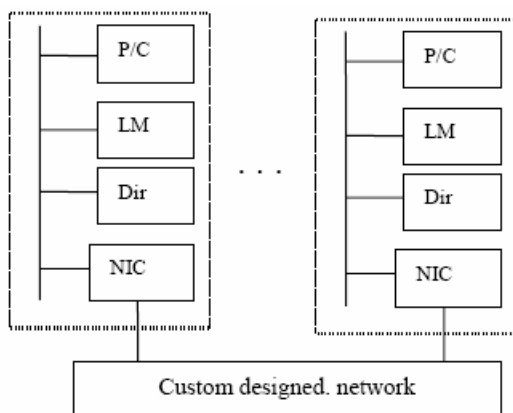
PVP - Parallel Vector Processor



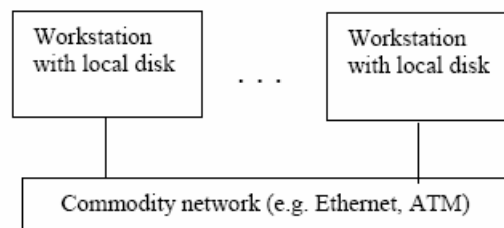
SMP - Symmetric Multiprocessors



MPP - Massively Parallel Processors



DSM - Distributed Shared Memory



COW - Cluster of Workstation

SM - shared memory module
LM - local memory
NIC - network interface circuitry
VP - vector processor
P/C - scalar processor and cache
Dir - address directory/translation

计算机的发展

- 40年代开始的现代计算机发展历程可以分为两个明显的发展时代：串行计算时代、并行计算时代。
- 每一个计算时代都从**体系结构**发展开始，接着是**系统软件**（特别是编译器与操作系统）、**应用软件**，最后随着**问题求解环境**的发展而达到顶峰
- 创建和使用并行计算机的主要原因是因为并行计算机是解决单处理器速度瓶颈的最好方法之一



Cray X1

Latest Cray
Supercomputer

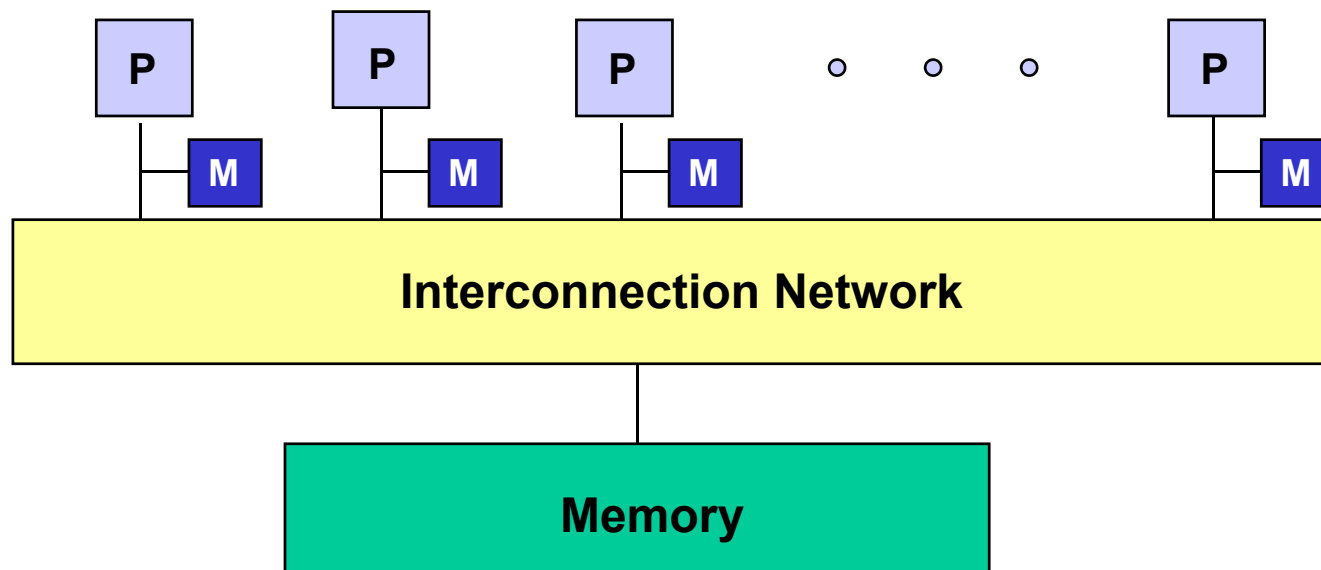
Up to 52.4
teraflops of
peak
computing
power and
65.5 TB of
memory

U.S. list pricing
starts at about
\$2.5 million.

并行结构

- 并行计算机是由一组处理单元组成的，处理单元通过相互之间的通信与协作，以更快的速度共同完成一项大规模的计算任务。因此，并行计算机的两个最主要的组成部分是**计算节点**和**节点间的通信与协作机制**
- 并行计算机体系结构的发展也主要体现在**计算节点性能**的提高以及**节点间通信技术**的改进两方面
- 并行计算机与传统计算机的区别在于其通信架构：
 - 互联网络的特性
 - 处理器的通信、同步方法等

一个通用的并行结构

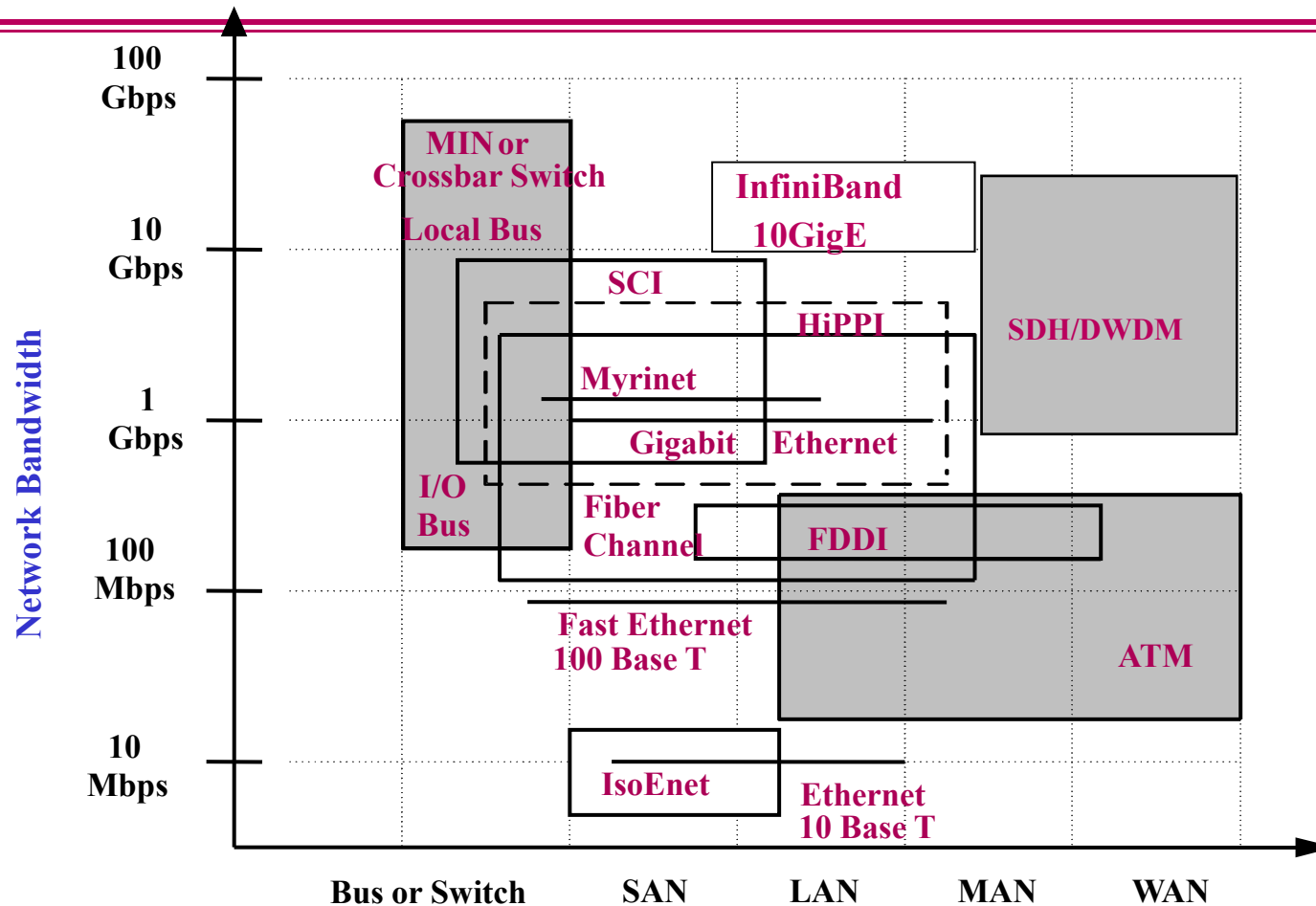


网络拓扑：处理器、内存和I/O的连接方式

主要内容

- 互联网络
- 存储模型

不同带宽与距离的互连技术



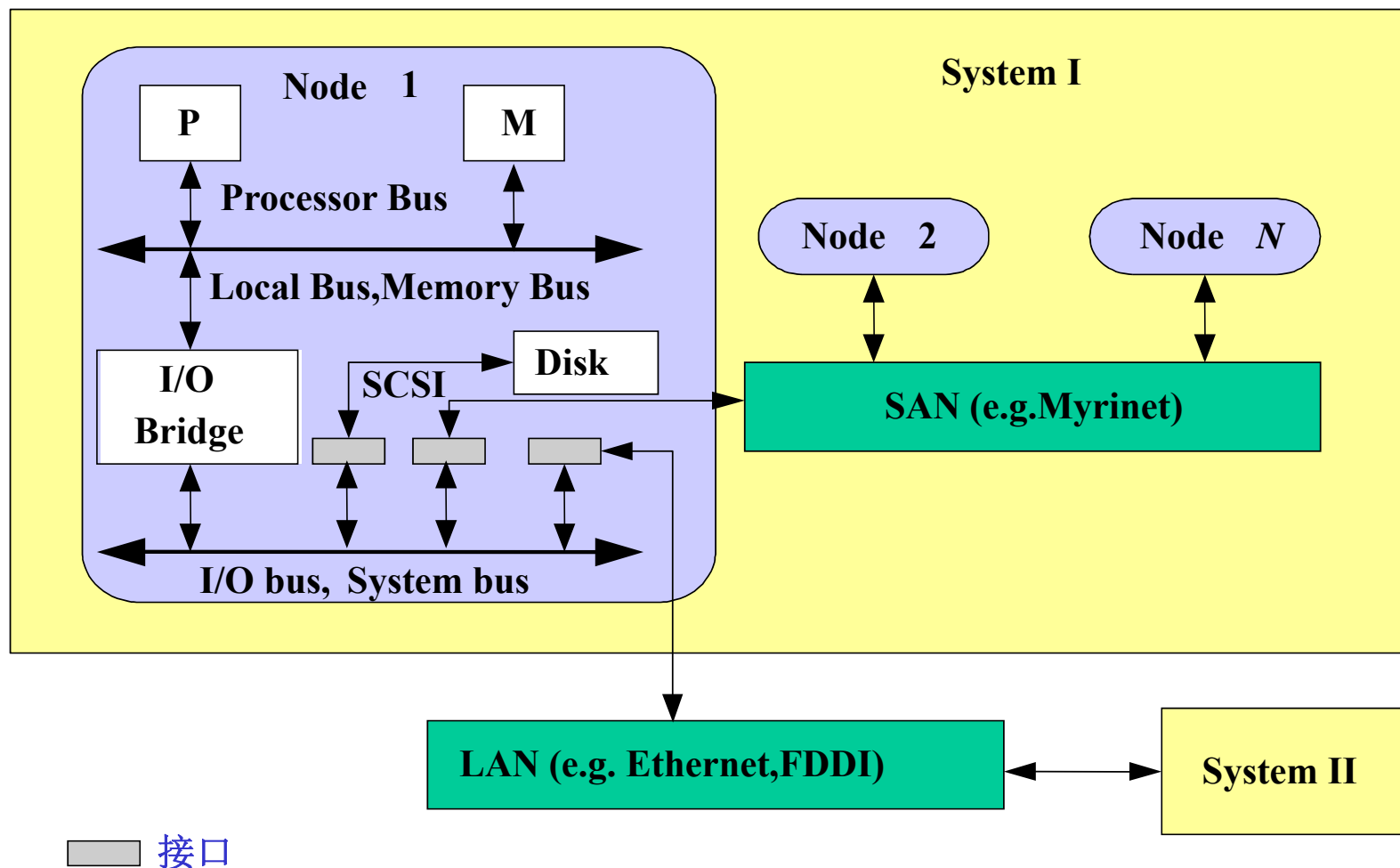
SAN: System Area Network (系统域)

LAN: Local Area Network (局域网)

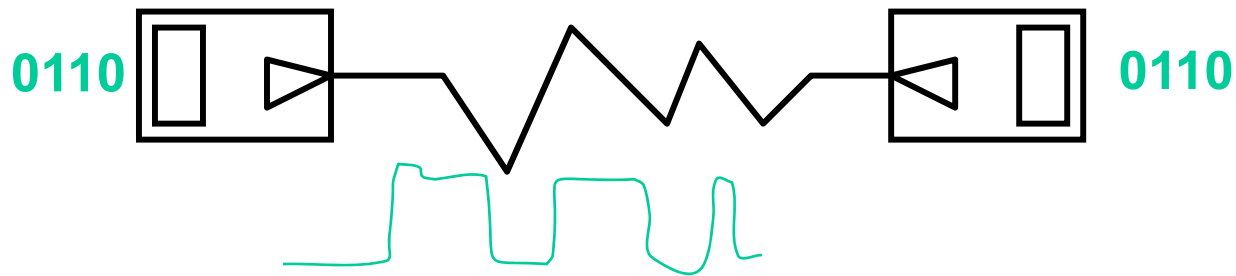
MAN: Metropolitan Area Network (城域网)

WAN: Wide Area Network (广域网)

本地总线, IO总线, SAN 和 LAN

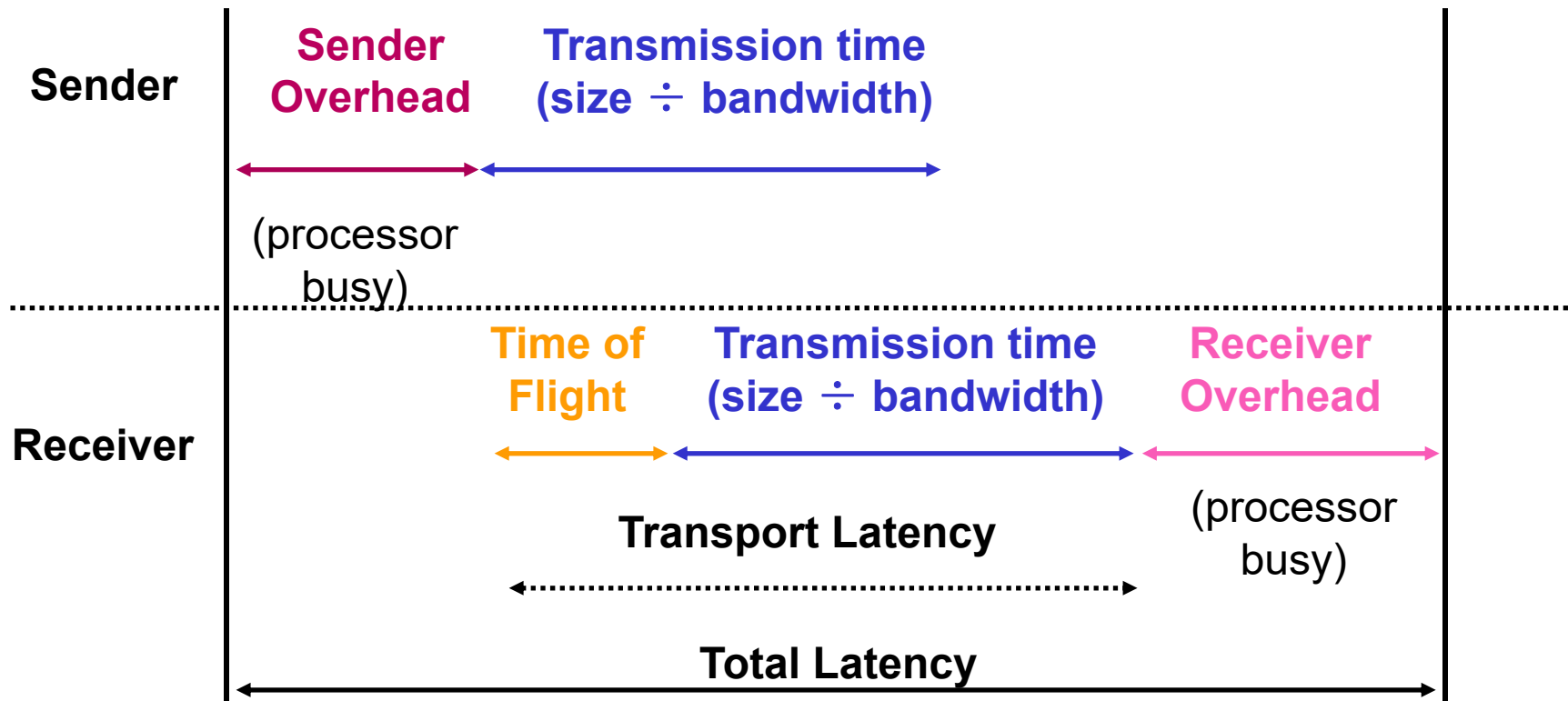


复习：基本网络部件



- 链路（**link**）：传输信息的物理介质
- 交换机/路由器开关（**switch/router**）：用于建立交换网络
- 网络接口电路（**NIC**）：用来连接主机和网络

网络性能指标—时延

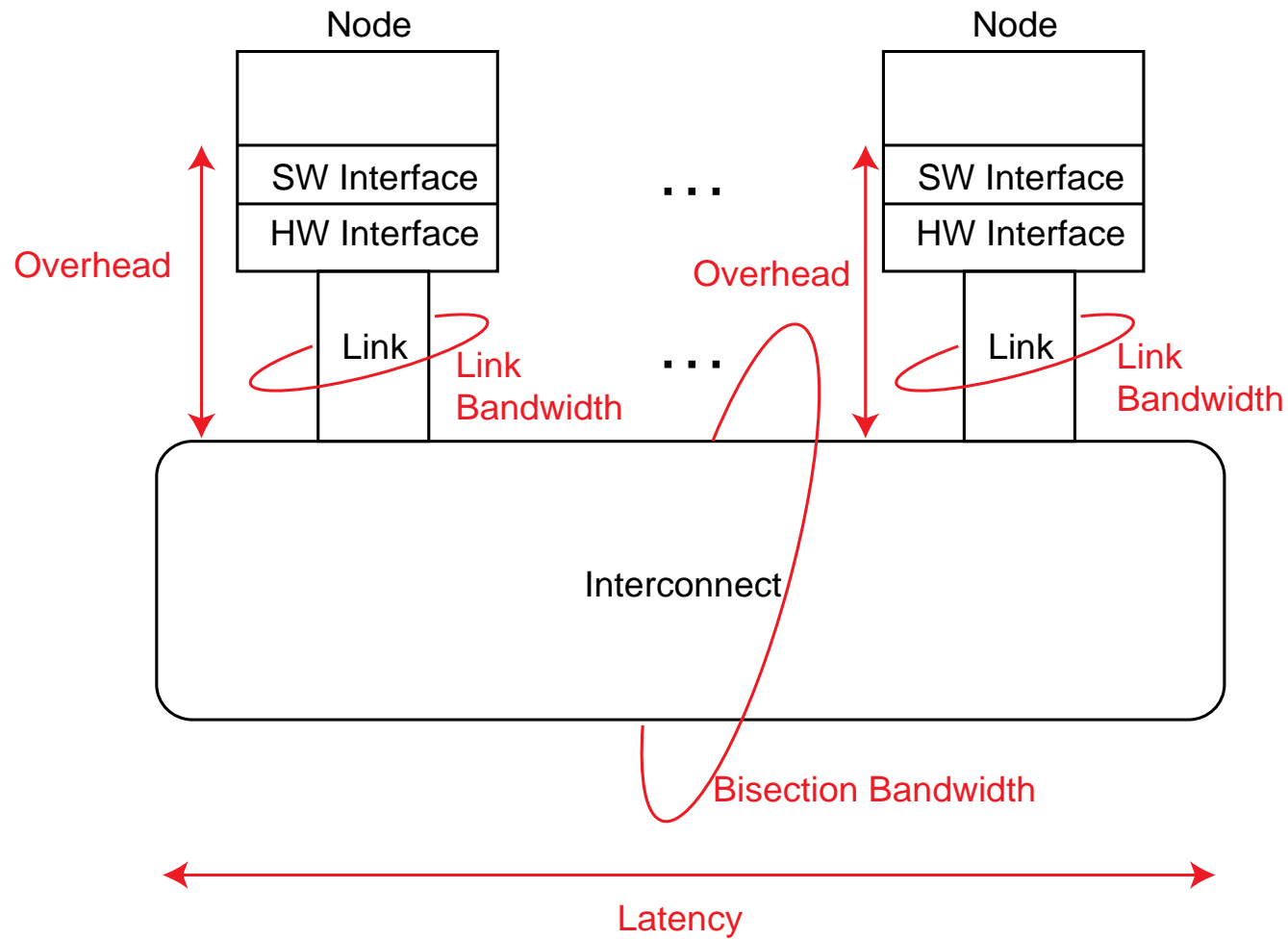


$$\text{Total Latency} = \text{Sender Overhead} + \text{Time of Flight} + \text{Message Size} \div \text{BW} + \text{Receiver Overhead}$$

时延 (Latency)

- **通信时延**：从源节点到目的节点传输一条消息所需的总时间
 - 在网络两端相应收发消息的**软件开销**
 - 由于通道占用导致的**通道时延**，即总的消息长度除以通道带宽
 - 沿选路路径作一系列选路决策期间花费在后续交换开关上的**选路时延**
 - 由于网络传输竞争导致的**竞争时延**
- **软件开销 (overhead)** 主要取决于主机内核，与**竞争时延**均依赖于程序行为
- **传输时延 (transport latency)**：通道时延 (transmission time) 和选路时延之和

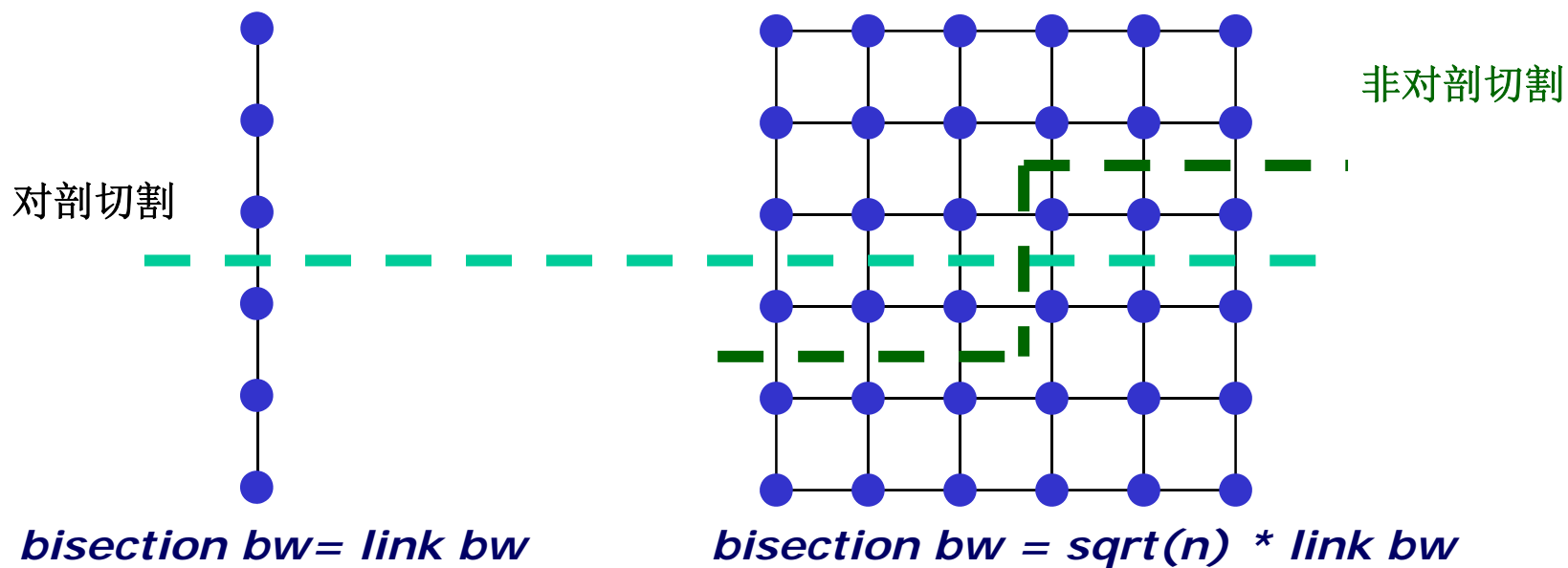
网络性能指标—带宽



带宽（Bandwidth: BW）

- **端口带宽**：从任意端口到另外端口单位时间内传输消息的最大位（或字节）数
 - 如IBM HPS 每端口带宽40MB/s
- **聚集带宽**：从一半节点到另一半节点，单位时间内传输消息的最大位（或字节）数
 - 如IBM HPS端口数最多为512，聚集带宽为 $512 * 40 / 2 = 10.24\text{GB/s}$
- **链路带宽（Link Bandwidth）**：单位时间内链路传输消息的最大位（或字节）数
- **对剖宽度**：将网络分成两个相等部分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。等于对剖宽度与链路带宽之积

对剖带宽与链路带宽



互联网络的评价标准

- 硬件复杂度 (**Cost**)
 - 将N个处理机按一定拓扑结构连成网络所需的开关个数
- 时延 (**Latency**)
 - 发送消息到接收消息所需的时间
- 带宽 (**Bandwidth**)
 - 单位时间内传送的数据量
- 可扩展性 (**Scalability**)
- 容错能力 (**Reliability**)

静态互连网络与动态互连网络

- 静态互连网络：处理单元间有着固定连接的一类网络，在程序执行期间，这种点到点的链接保持不变
 - 一维线性阵列（**1-Linear Array**）
 - 二维网孔（**2-D Mesh**）
 - 树（**Tree**）
 - 超立方（**Hypercube**）
 - 蝶形网络（**Butterfly**）
- 动态网络：用交换开关构成的，可按应用程序的要求动态地改变连接组态
 - 总线（**Bus**）
 - 交叉开关（**Crossbus Switcher**）
 - 多级互连网络（**MIN-Multistage Interconnection Network**）

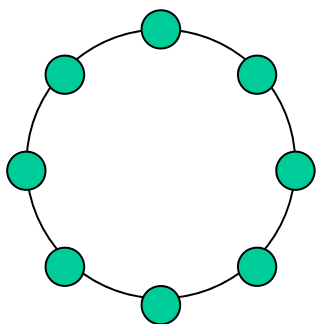
主要内容

- 系统互联
 - 静态互联网络
 - 动态互连网络
 - 标准互联网络
- 存储模型

一维线性阵列 (1-D Linear Array)



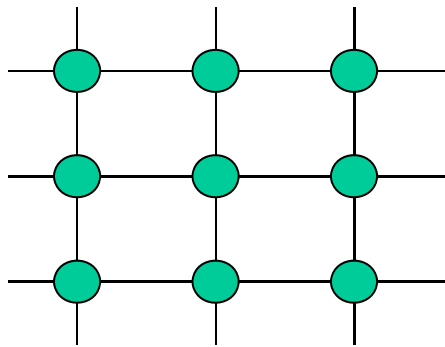
Linear Array



Ring

- 并行机中最简单最基本的互连方式
- 每个节点只与其左、右近邻相连，也叫二近邻连接，
- N 个节点用 $N-1$ 条边串接之，内节点度为2，直径为 $N-1$ ，对剖宽度为1
- 当首、尾节点相连时可构成循环移位器，在拓扑结构上等同于环，环可以是单向的或双向的
 - 节点度恒为2
 - 直径或为 $\lfloor N/2 \rfloor$ （双向环）或为 $N-1$ （单向环）
 - 对剖宽度为2
- 例子：FDDI, SCI

二维网孔（2-D Mesh）



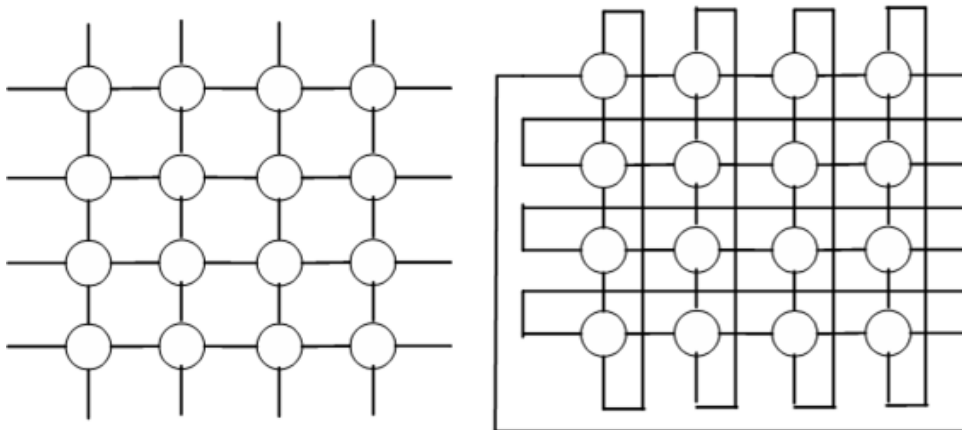
2-D Mesh

- 每个节点只与其上、下、左、右的近邻相连（边界节点除外）
 - 网络规模： N
 - 节点度： 4
 - 网络直径： $2(N^{1/2} - 1)$
 - 对剖宽度： $N^{1/2}$
- 例子： Intel paragon

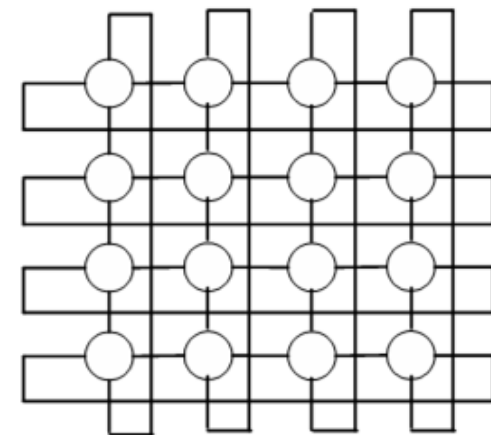
二维网孔 (2-D Mesh)

- 在垂直方向上带环绕，水平方向呈蛇状，就变成Illiac网孔了
 - 节点度恒为4，网络直径为 $N^{1/2} - 1$ ，而对剖宽度为 $2N^{1/2}$
- 垂直和水平方向均带环绕，则变成了2-D环绕 (2-D Torus)
 - 节点度恒为4，网络直径为 $2\lfloor N^{1/2}/2 \rfloor$ ，对剖宽度为 $2N^{1/2}$

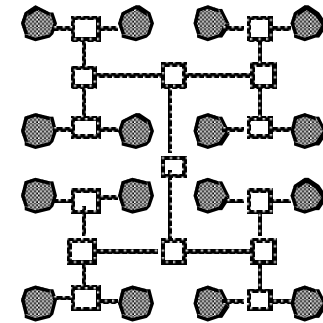
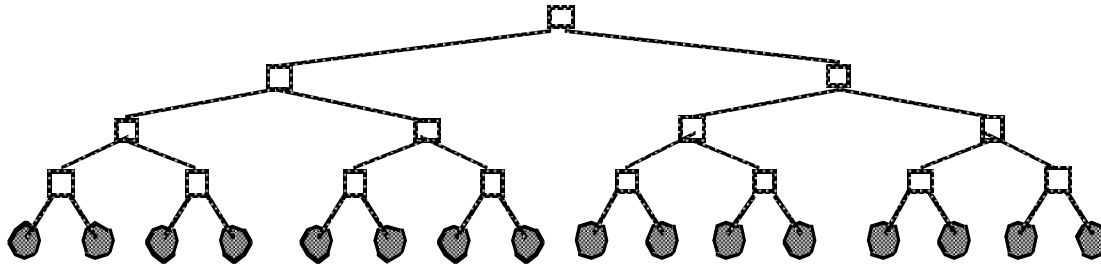
Illiac Mesh



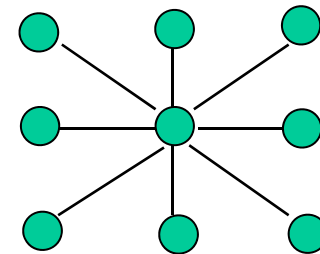
2-D Torus



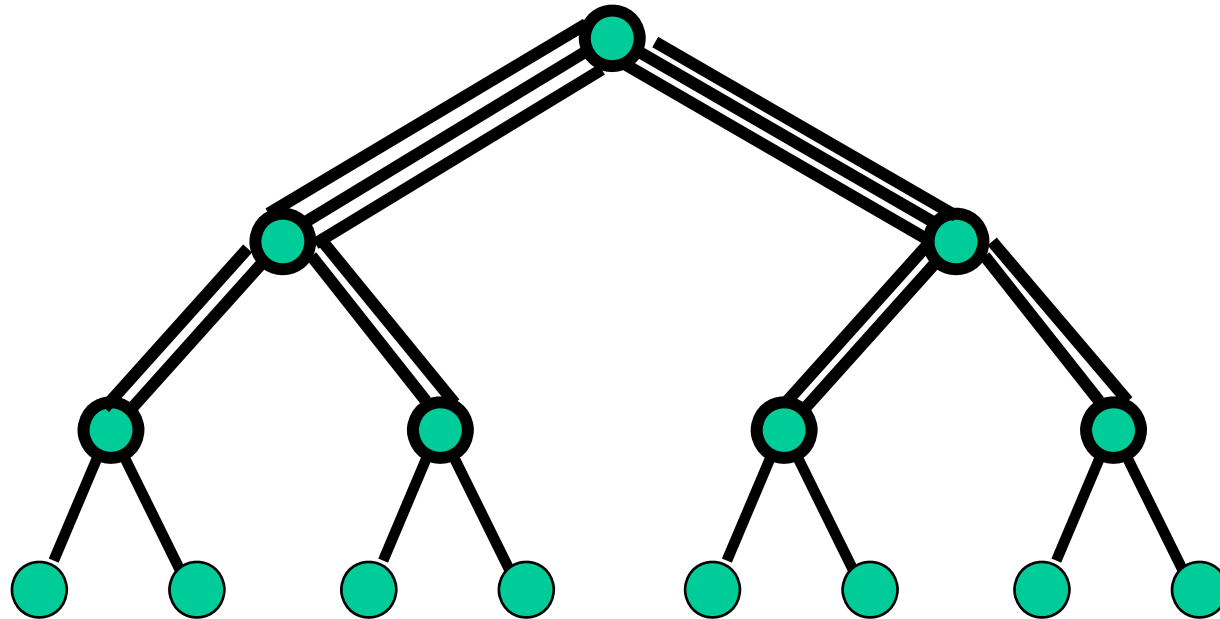
二叉树



- 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（三近邻连接），对剖宽度为1，
 - 树的直径为 $2*k-2=2\lceil\log N\rceil-1$ （k为层数，N为树的总节点数）
- 如果尽量增大节点度为N-1，则直径缩小为2，此时就变成了星形网
 - 对剖宽度为 $\lfloor N/2 \rfloor$

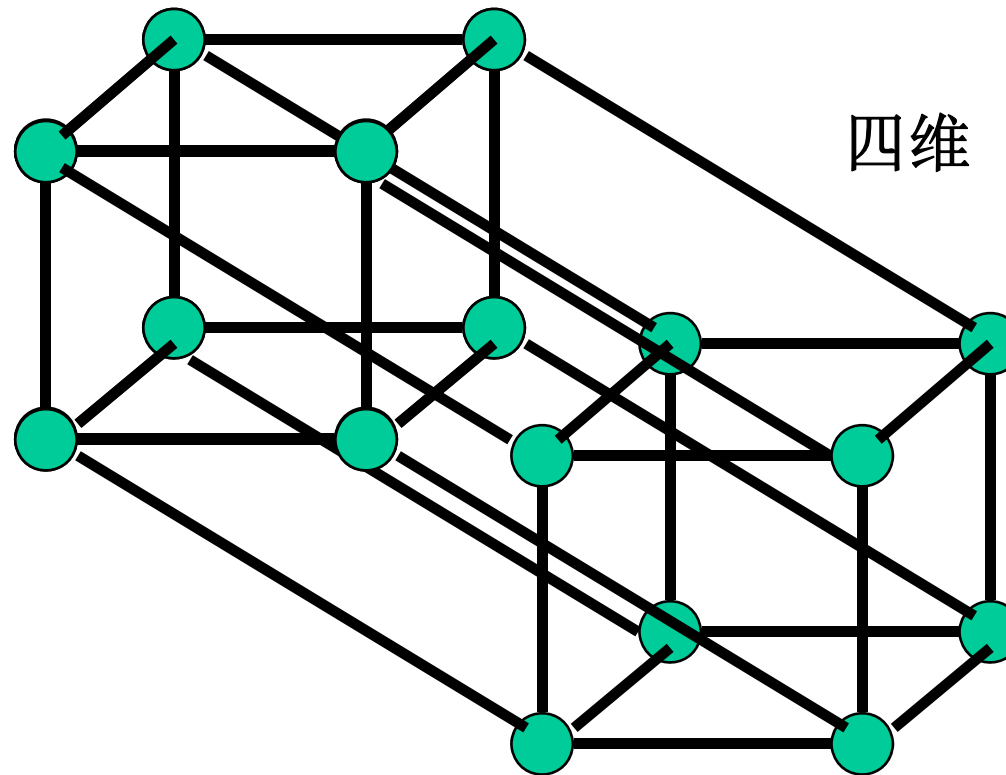


胖树 (Fat-Trees)



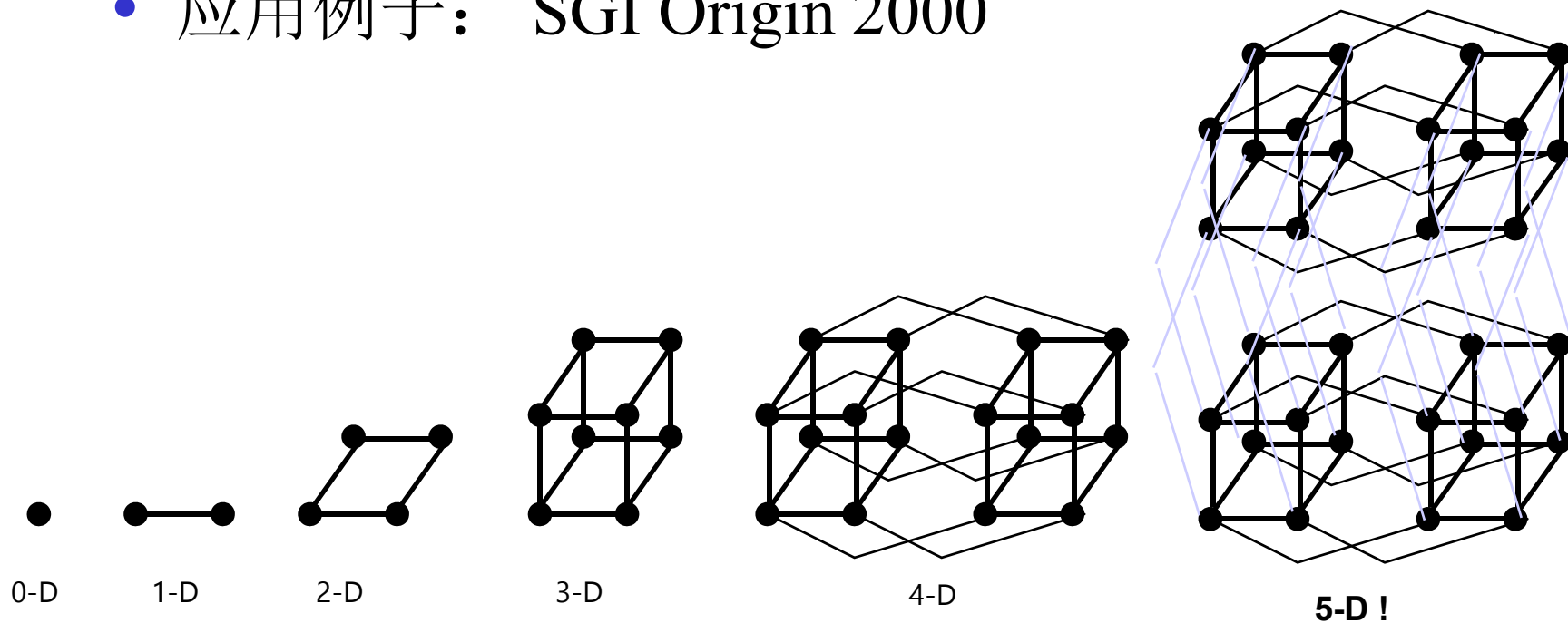
- 传统二叉树的主要问题是根易成为通信瓶颈。胖树节点间的通路自叶向根逐渐变宽
- 对剖带宽随着N的增大而增大
- 例子: Infiniband

超立方 (Hypercube)



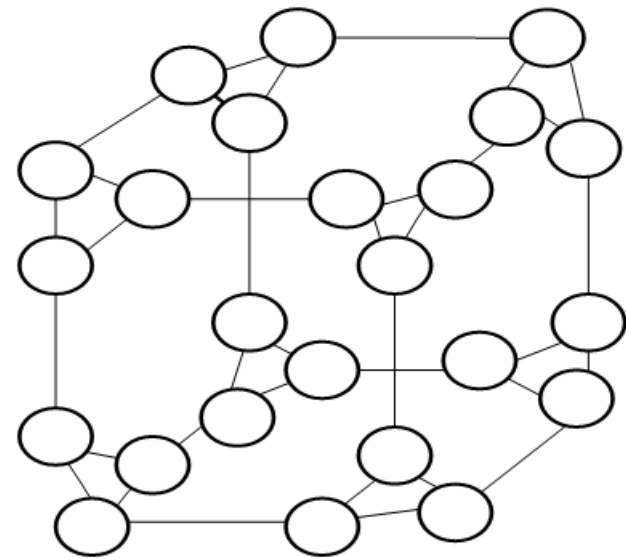
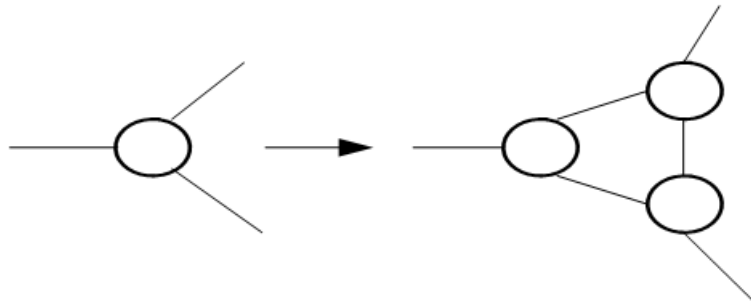
超立方

- 一个n-立方由 $N=2^n$ 个顶点组成
 - n-立方的节点度为n，网络直径也是n，而对剖宽度为 $N/2$
- 应用例子： SGI Origin 2000



3-立方环

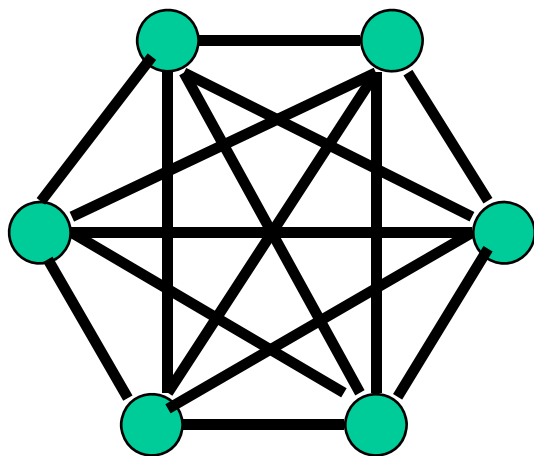
- 如果将3-立方的每个顶点代之以一个环就构成了3-立方环，此时每个顶点的度为3，而不像超立方那样节点度为 n 。



静态互连网络特性比较（表1.3）

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	$\lfloor N/2 \rfloor$	2	是	N
2-D网孔	$N = n^2$	4	$2(n-1)$	n	非	$2(N-n)$
Illiac网孔	$N = n^2$	4	$n-1$	$2n$	非	$2N$
2-D环绕	$N = n^2$	4	$2 \lfloor n/2 \rfloor$	$2n$	是	$2N$
二叉树	N	3	$2(\lceil \log_2 N \rceil - 1)$	1	非	$N-1$
星形	N	$N-1$	2	$\lfloor N/2 \rfloor$	非	$N-1$
超立方	$N=2^n$	$\log_2 N=n$	N	$N/2$	是	$nN/2$
立方环	$N=k \cdot 2^k$	3	$2k-1 + \lfloor k/2 \rfloor$	$N/(2k)$	是	$3N/2$

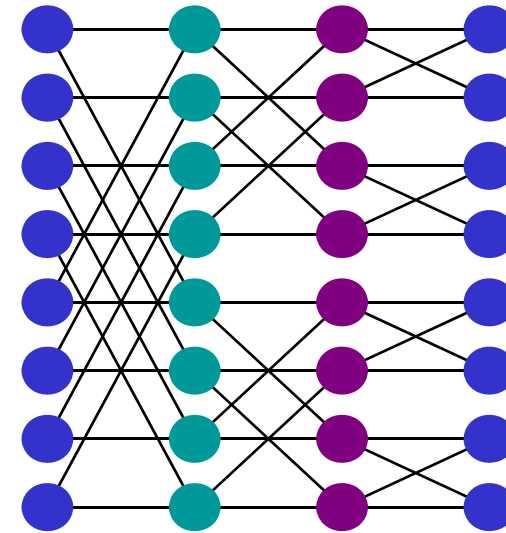
例子：完全连接



- 网络规模: N
- 连接数: $N*(N - 1) / 2$
- 节点度: $N - 1$
- 对剖宽度: $(N / 2)^2$
- 直径 = 1

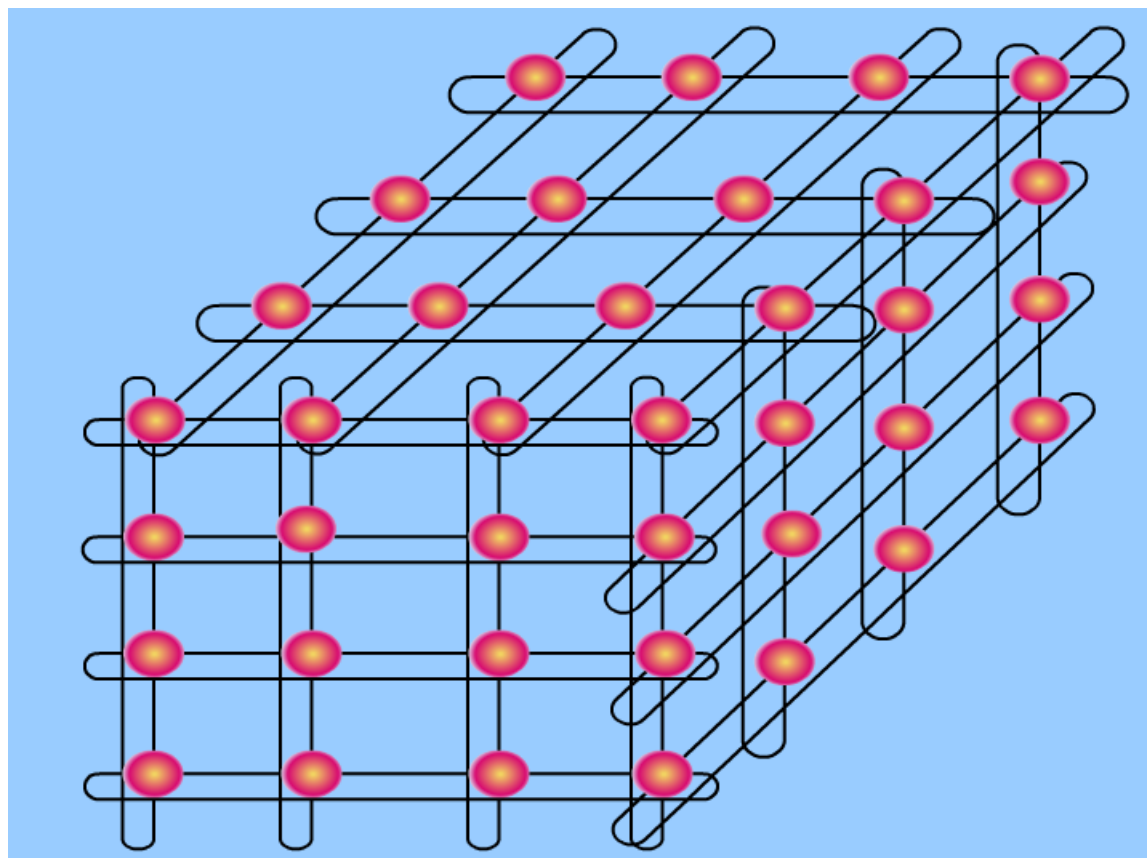
例子：蝶网 (butterfly)

- 节点: $N = (k+1)2^k$
- 直径: $2k$
- 对剖带宽: 2^k



multistage butterfly network

例子：4元3立方体的网络拓扑结构




- k元-n立方体网络中，参数k是基数或者说是沿每个方向的结点数，n是立方体的维数。这两个数与网络中节点数N的关系为 $N=k^n$

静态互联网络小结

- 大多数静态网络的**节点度**都小于4，这是比较理想的。若能实现所有节点的连接，节点度愈小愈好，当然要求相应的网络时延也是愈小愈好
- 节点度愈大，表示连接性愈好，但网络的连接复杂，成本高
- **对剖带宽**愈大，表示网络的带宽就愈大；网络直径愈大，表示通信的时间延迟就愈大。
- 网络的**造价**随节点度和链路数增大而上升。
- 固定网络规模，对剖宽度越大，或网络直径越小，则互联网络质量越高
- **对称性**会影响可扩展性和路由效率
- 环、Mesh、环绕、超立方体、 k 元 n -立方体可用于MPP系统

真实机器中的拓扑结构

 older newer	Red Storm (Opteron + Cray network)	3D Mesh
	IBM Blue Gene/L	3D Torus
	SGI Altix	Fat tree
	Cray X1	4D Hypercube
	Quadrics	Fat tree
	IBM SP	Fat tree (近似)
	SGI Origin	Hypercube
	Intel Paragon	2D Mesh

主要内容

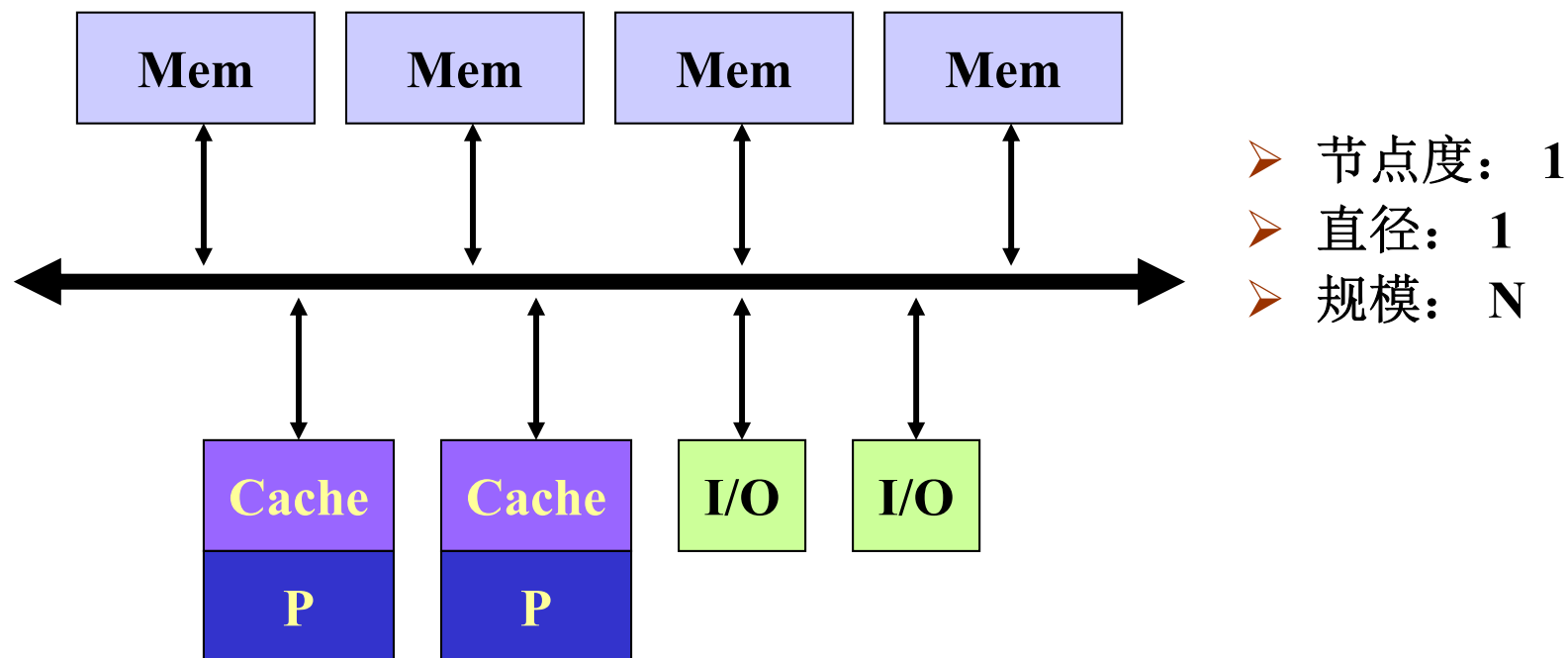
- 互联网络
 - 静态互联网络
 - 动态互连网络
 - 标准互联网络
- 存储模型

动态互联网络

- 通信模式是基于程序的要求
- 连接实在程序执行过程总实时建立
- 基于总线（**Bus-based**）或开关（**Switch-based**）

总线

- 总线：PCI、VME、Multics、Sbus、MicroChannel, IEEE Futurebus



多处理机总线和层次总线，常用来构筑SMP、NUMA和DSM机器

多处理机总线

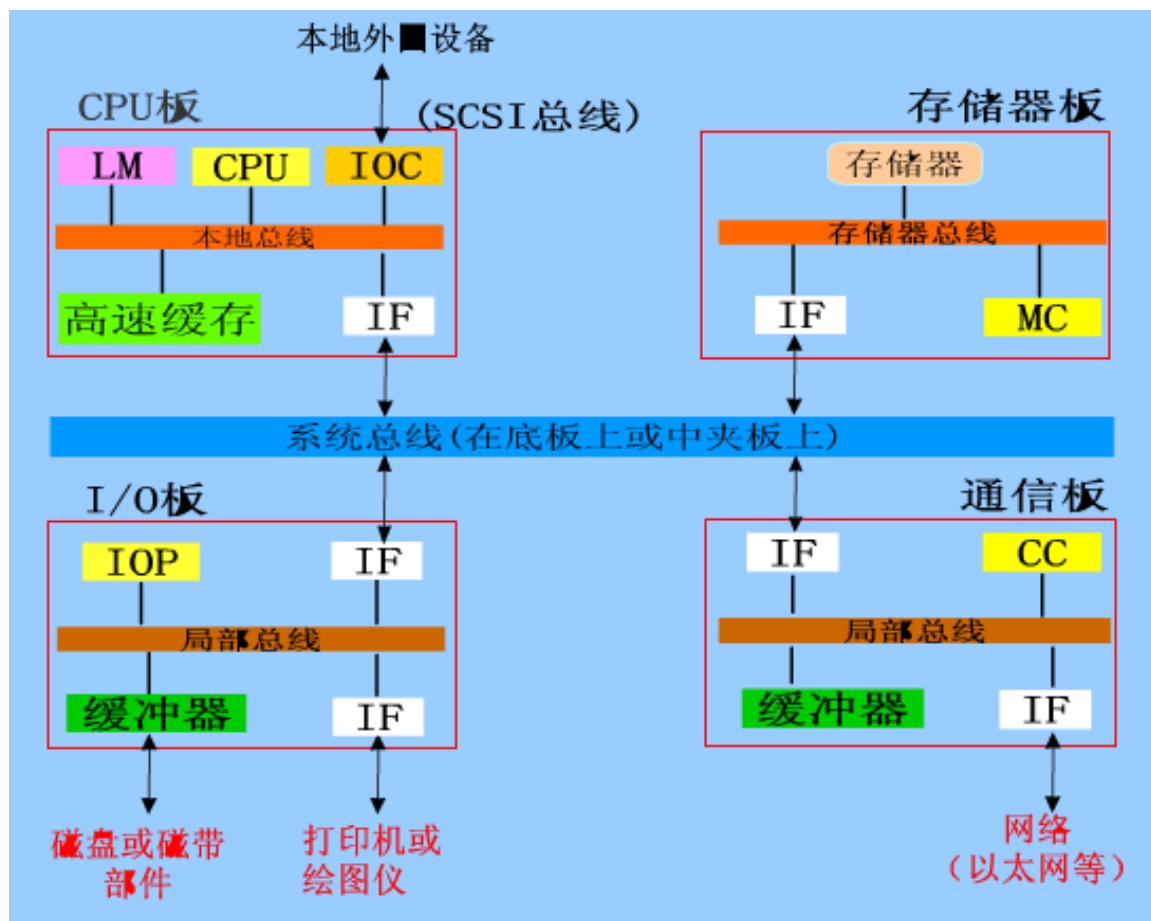
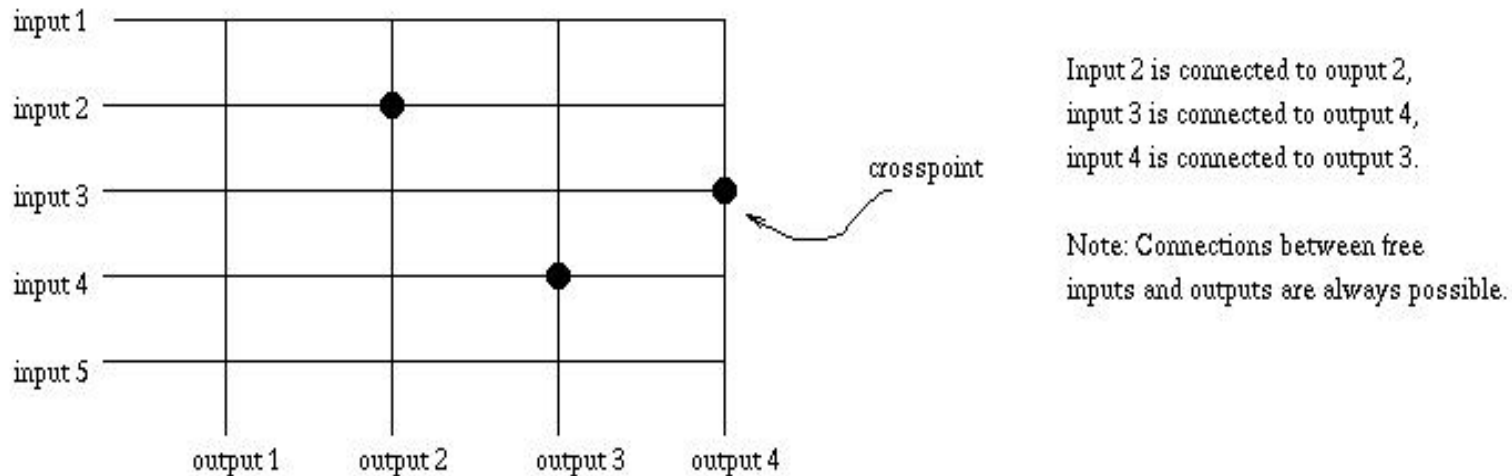


图1.9

总线的特点

- 总线的优点在于成本低，不随处理器数目的增加而增加
- 总线的缺点在于扩展性不好，总线的带宽固定，随着处理器数的增加，每个处理器带宽减少
- 可利用程序中的局部性原理减少对总线带宽的需求

交叉开关

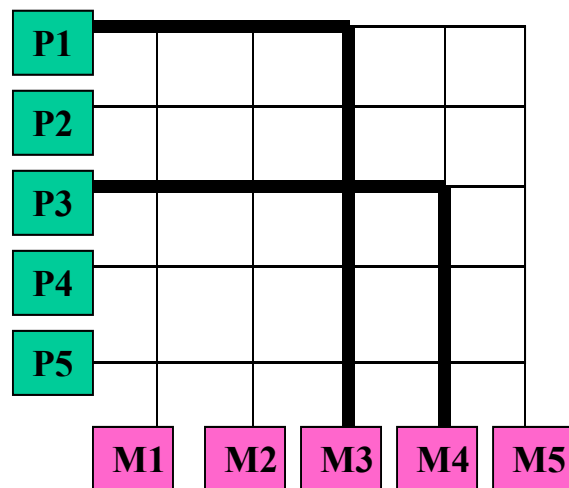


A Small Crossbar Switch

- 单级交换网络，可为每个端口提供更高的带宽。象电话交换机一样，交叉点开关可由程序控制动态设置其处于“开”或“关”状态，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[处理器间的通信](#)；另一种是用于[处理器和存储模块之间的存取](#)。

交叉开关的应用例子

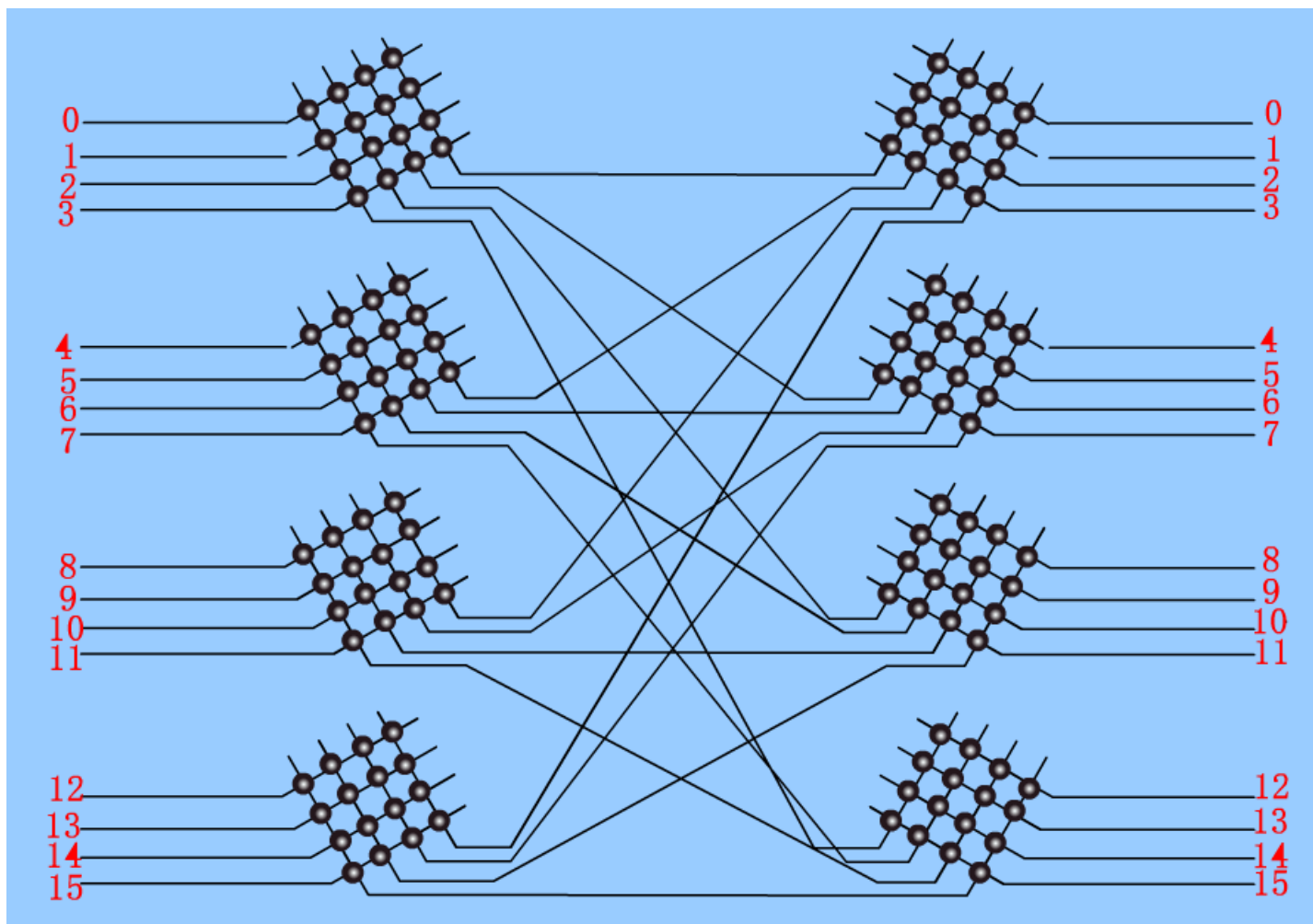
- Sun Microsystems 的 Ultra Enterprise 10000 (SMP, 1997)
 - 400 MHz UltraSparc 2 CPU's, 2 floats/cycle.
 - UMA (Uniform Memory Access)
 - 16 KB data cache (32 byte linesize), 4MB level 2 cache, 64 GB memory per processor
 - 网络: 10 GB/sec (聚合带宽), 600 ns时延



交叉开关的特点

- 交叉开关具有良好的带宽特性
- 非阻塞通信 (**Non-Blocking**)：两个节点之间的通信，不会阻塞其他节点之间的通信。
- 代价不可扩充， $O(n^2)$ (n 是交叉开关中的交叉点数)

多极互联的交叉开关网络



用8个4×4的交叉开关构成二级16×16的交叉开关网络

多级互连网络

- 单级交叉开关级联起来形成多级互连网络MIN (Multistage Interconnection Network)

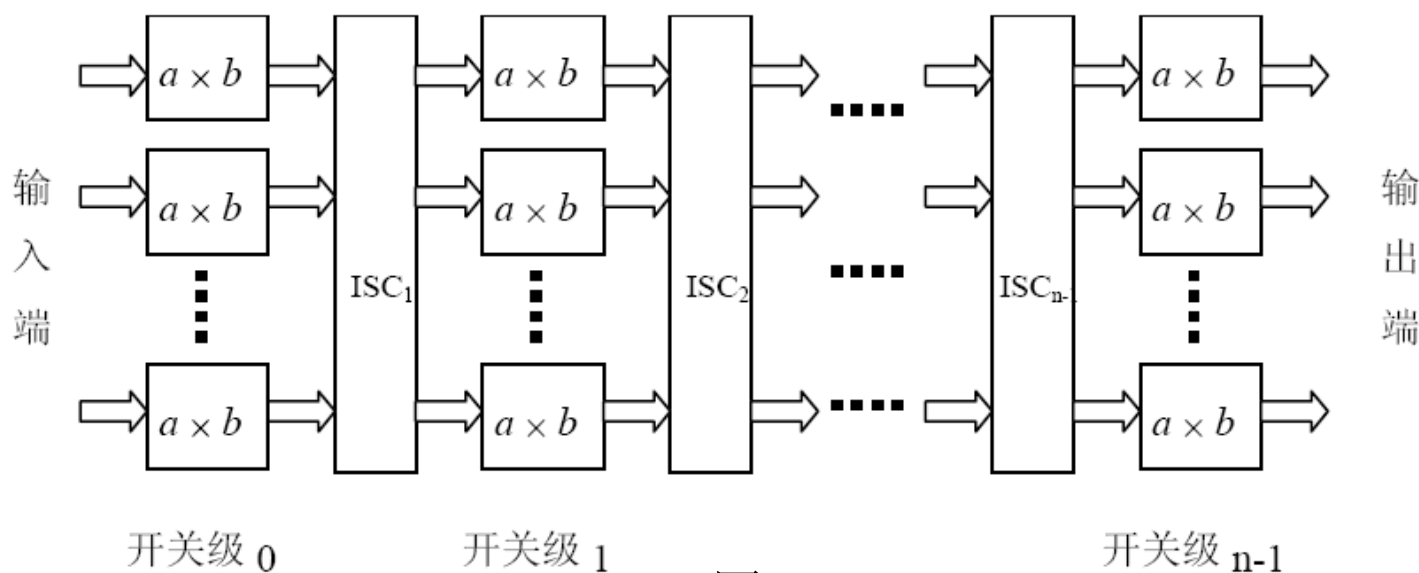


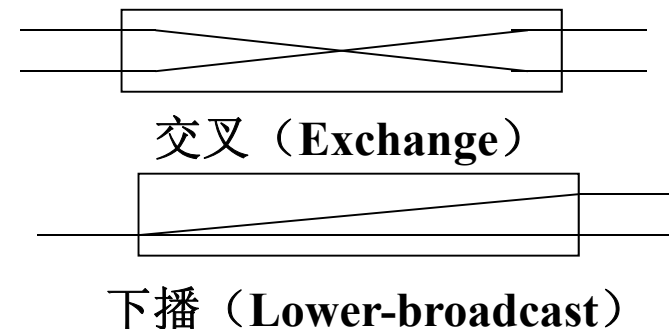
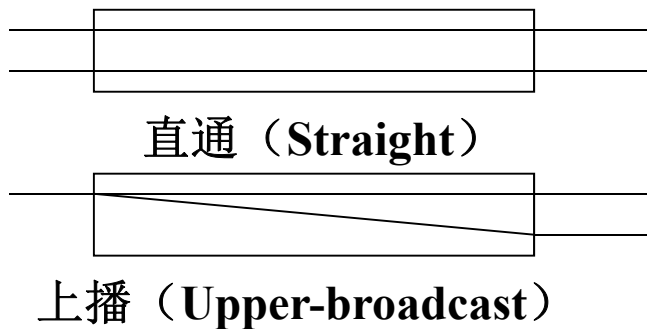
图1.11

- N 个输入, N 个输出
- 度: 1, 直径: $\log N$, 网络规模: $N \log N$

交换开关

- 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，不允许多对一的映射，否则将发生冲突



4种可能的交叉开关 (图1.12)

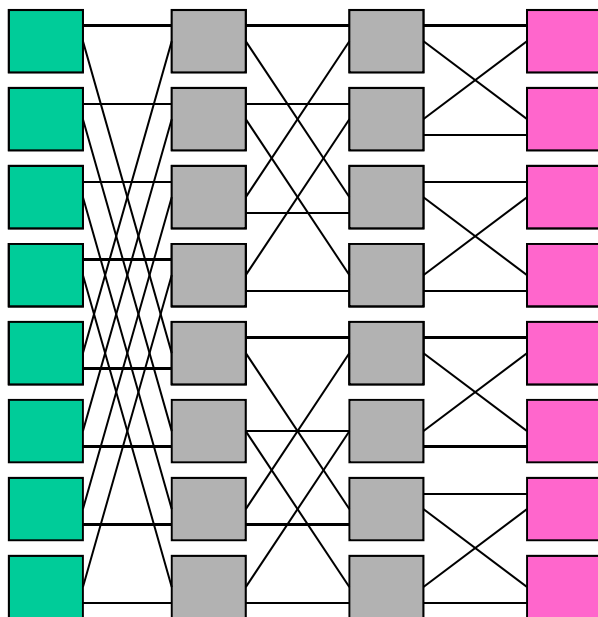
级间互连

(ISC: Inter-Stage Connection)

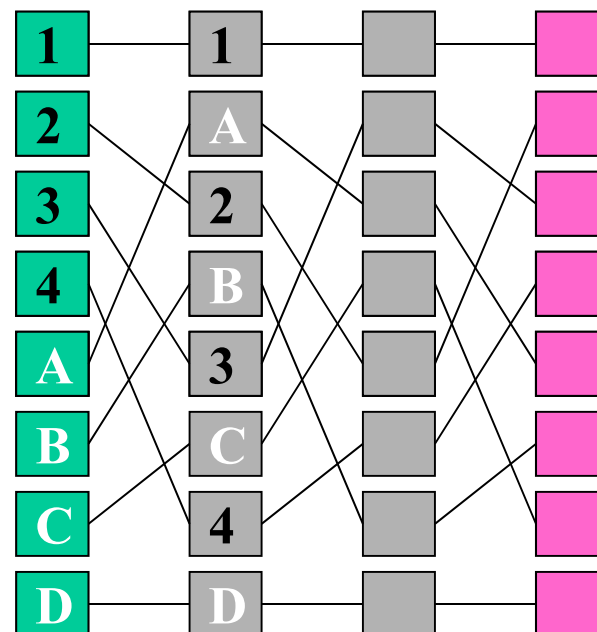
- 级间互连 (ISC: Inter-Stage Connection) :
 - 每一级输入与输出之间连接，输出作为交换开关的输入连到下一级
 - 均匀洗牌 (Perfect-Shuffle)、蝶式 (Butterfly)、交叉开关等

MIN的例子

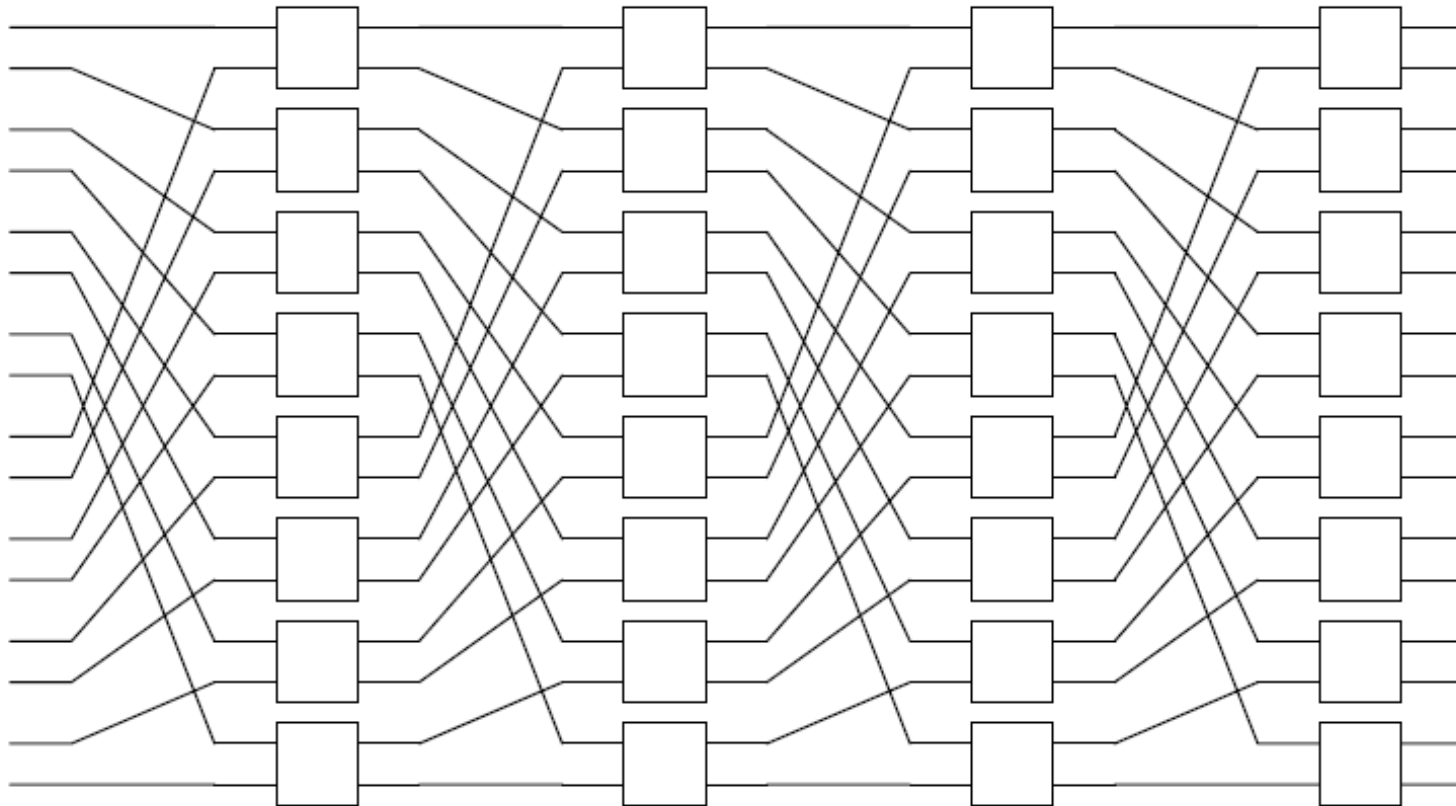
- 蝶式 (Butterfly)



- 均匀洗牌 (Shuffle)



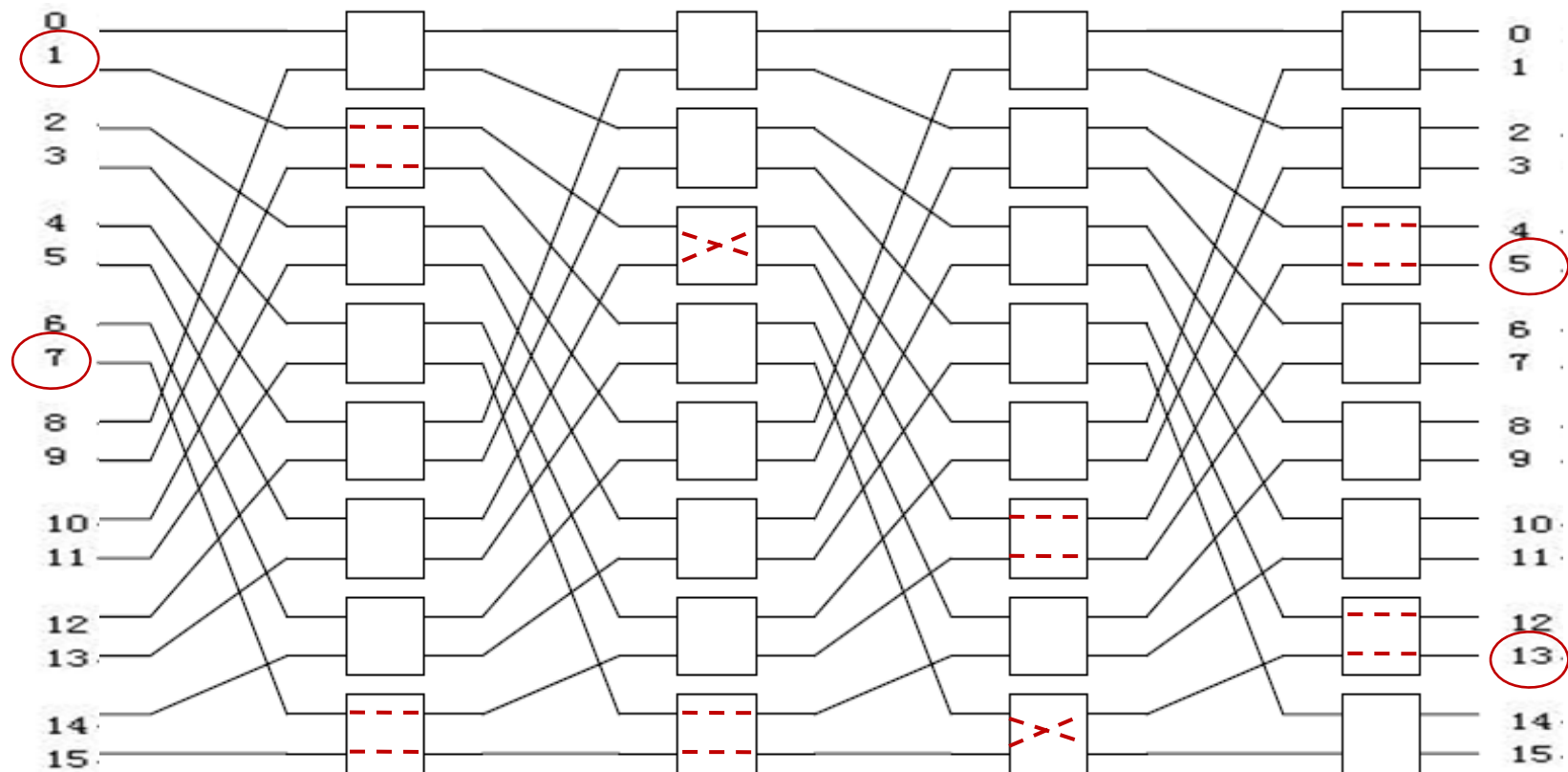
应用例子： Ω 网络



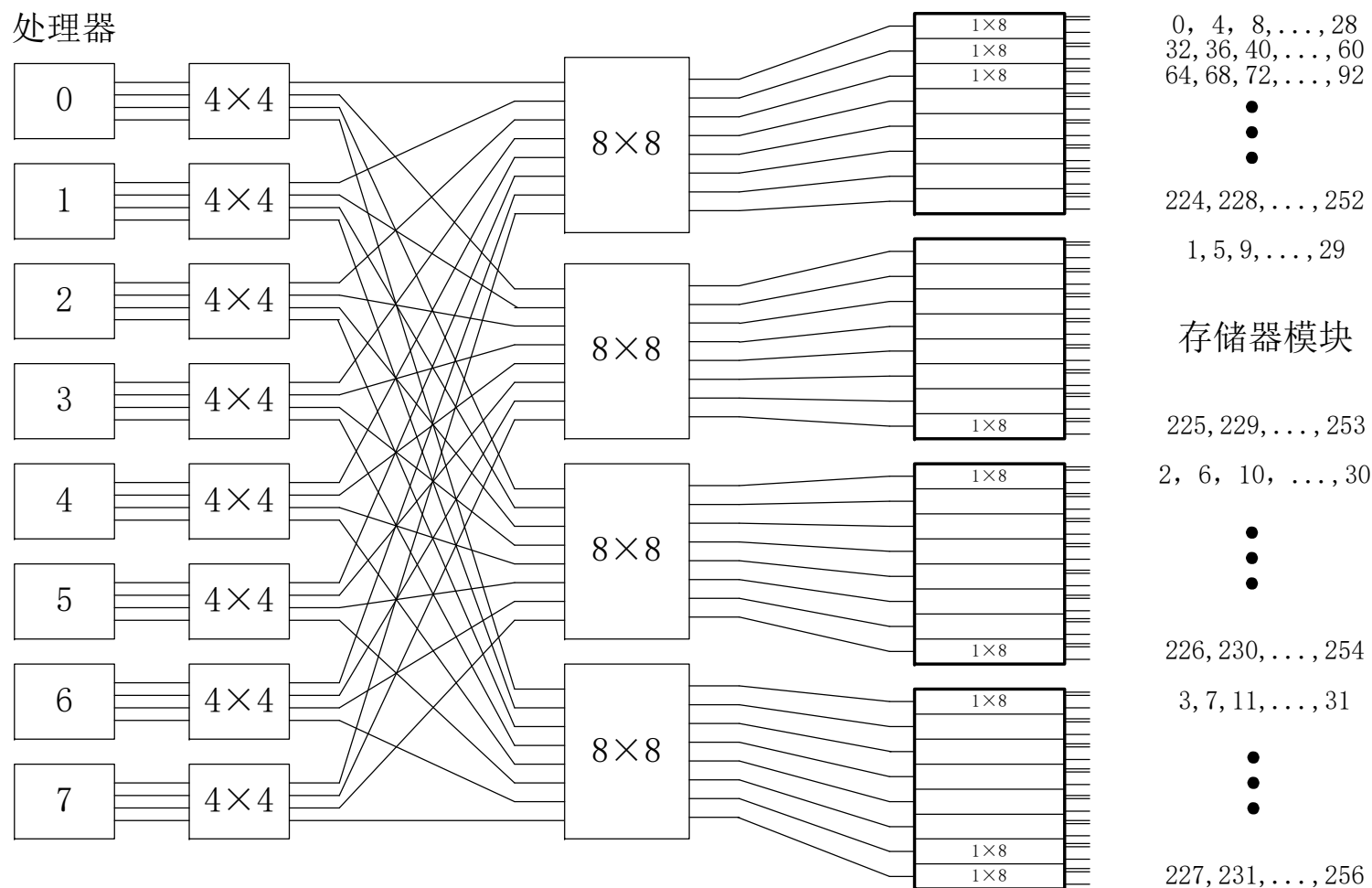
16×16 Ω 网络（图1.12）

应用例子: **IBM SP2, IBM option white, IBM Blue Pacific**

Ω 网络连接示例



应用例子：Cray Y-MP多级网络



Cray Y-MP多级网络

动态互连网络比较（表1.4）

网络特性	每个处理器带宽	硬件复杂度	阻塞	实际机器：聚集带宽
总线	$O(wf/n) \sim O(wf)$	$O(n+w)$	是	SunFire服务器中的 Gigaplane 总线: 2.67GB/s
多级互联	$O(wf)$	$O((n \log_k n)w)$	是	IBM SP2中的512节点的HPS: 10.24GB/s
交叉开关	$O(wf)$	$O(n^2w)$	否	Digital 的千兆开关: 3.4GB/s

n : 节点规模 w : 数据宽度 f : 时钟频率

主要内容

- 互联网络
 - 静态互联网络
 - 动态互连网络
 - 标准互联网络
- 存储模型

标准互联网络

- 开放、高带宽、低延迟、高可靠、扩展性好的交换网络和技术
- 主要应用于集群系统
- 标准互联网络：
 - **HiPPI**
 - **Scalable Coherent Interface (SCI)**
 - **Myrinet**
 - **Ethernet**
 - **Infiniband**

集群中的互连技术 (Myrinet)

- **Myrinet (ANSI/VITA 26-1998)** 是由Myricom公司设计的千兆位包交换网络，用于构建集群
 - 数据链路层具有可变长的包格式，对每条链路施行流控制和错误控制，并使用切通选路法以及定制的可编程的主机接口
 - 充分利用Myrinet在硬件架构上的特性，减轻主机的通信开销，每个数据包括只耗费主机大约 $1\mu\text{s}$ 的处理时间
 - 能适应任意拓扑结构，不必限定为开关网孔或任何规则的节点
- 在2004年7月的TOP 500中有187台 (37.4%) 的系统采用Myrinet
- 例子：曙光4000A



以太网

代别 类型		以太网 10BaseT	快速以太网 100BaseT	千兆位以太网 1GB	万兆位以太网 1GB
引入年代		1982	1994	1997	2002
速度（带宽）		10Mb/s	100Mb/s	1Gb/s	10Gb/s
最大 距离	UTR (非屏蔽双扭对)	100m	100m	25—100m	25—100m
	STP (屏蔽双扭对) 同轴电缆	500m	100m	25—100m	25—100m
	多模光纤	2Km	412m (半双工) 2Km (全双工)	500m	300m
	单模光纤	25Km	20Km	3Km	10—40Km

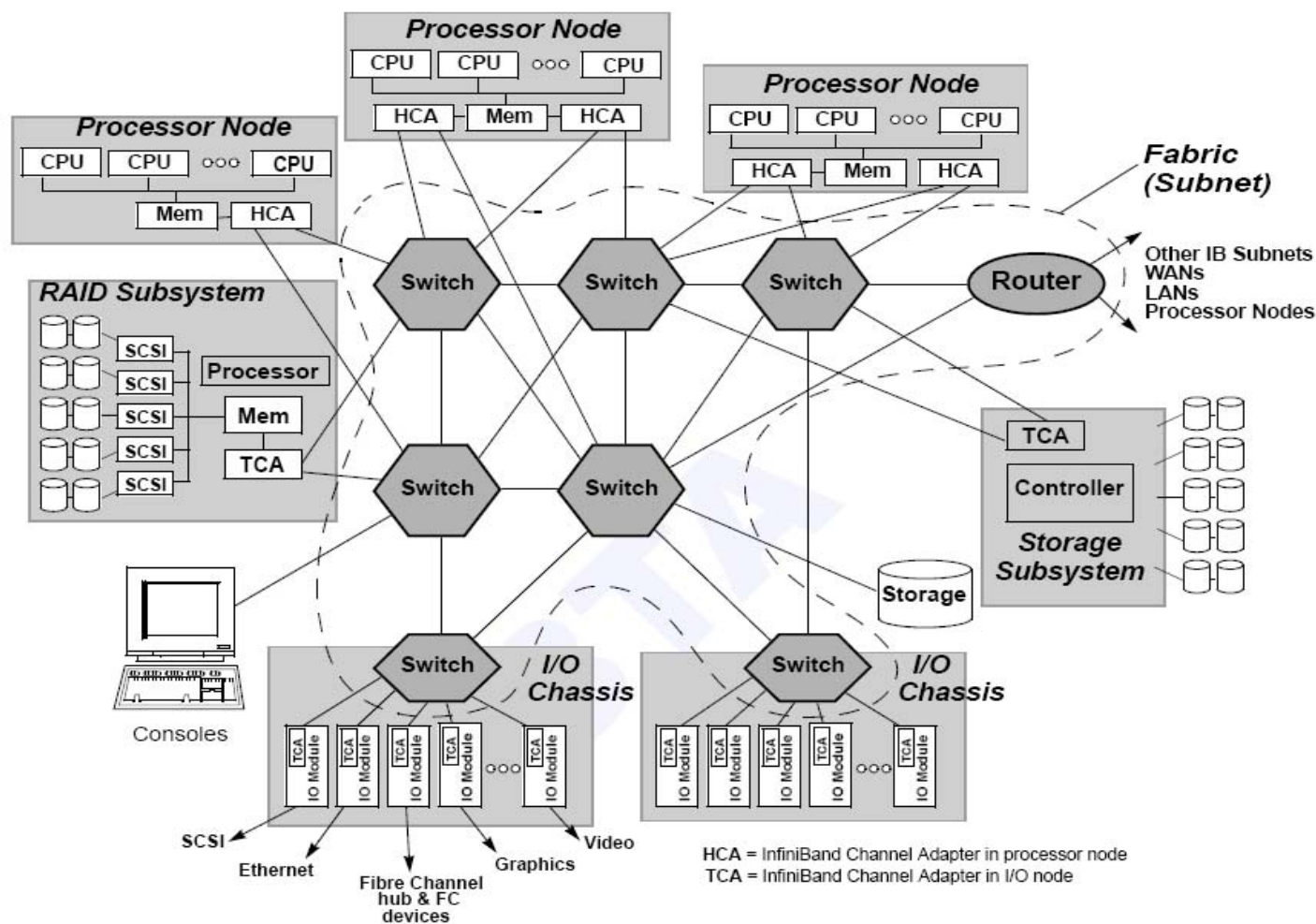
万兆以太网

- IEEE于2002年6月批准万兆以太网(10gige)802.3ae标准
- 万兆以太网是一种只采用全双工与光纤的技术，其技术特色首先表现在物理层面上：
 - 物理层（PHY）和OSI模型的第一层(物理层)一致，它负责建立传输介质（光纤或铜线）和MAC层的连接
 - 把PHY进一步划分为物理介质关联层（PMD）和物理代码子层（PCS）。光学转换器属于PMD层。PCS层由信息的编码方式(如64B/66B)、串行或多路复用等功能组成
- 万兆标准意味着以太网将具有更高的带宽（10G）和更远的传输距离(最长传输距离可达40公里)。
- 万兆以太网技术提供了更多的更新功能，如QoS，能更好的满足网络安全、服务质量、链路保护等多个方面需求。

InfiniBand (IB)

- **IB**是由**IB**行业协会所倡导的，**IBTA (Infiniband Trade Association)**于1999年成立，协会主要成员有：康柏、戴尔、惠普、**IBM**、**Intel**、微软和**Sun**
 - <http://www.infinibandta.org>
- **Infiniband**是一种可满足存储区域网、高端计算集群及局域网需求的全新互连标准
- **InfiniBand**技术通过一种交换式通信组织（**Switched Communications Fabric**）提供了较局部总线技术更高的性能，它通过硬件提供了可靠的传输层级的点到点连接，并在线路上支持消息传递和内存映像技术

InfiniBand体系结构



InfiniBand架构

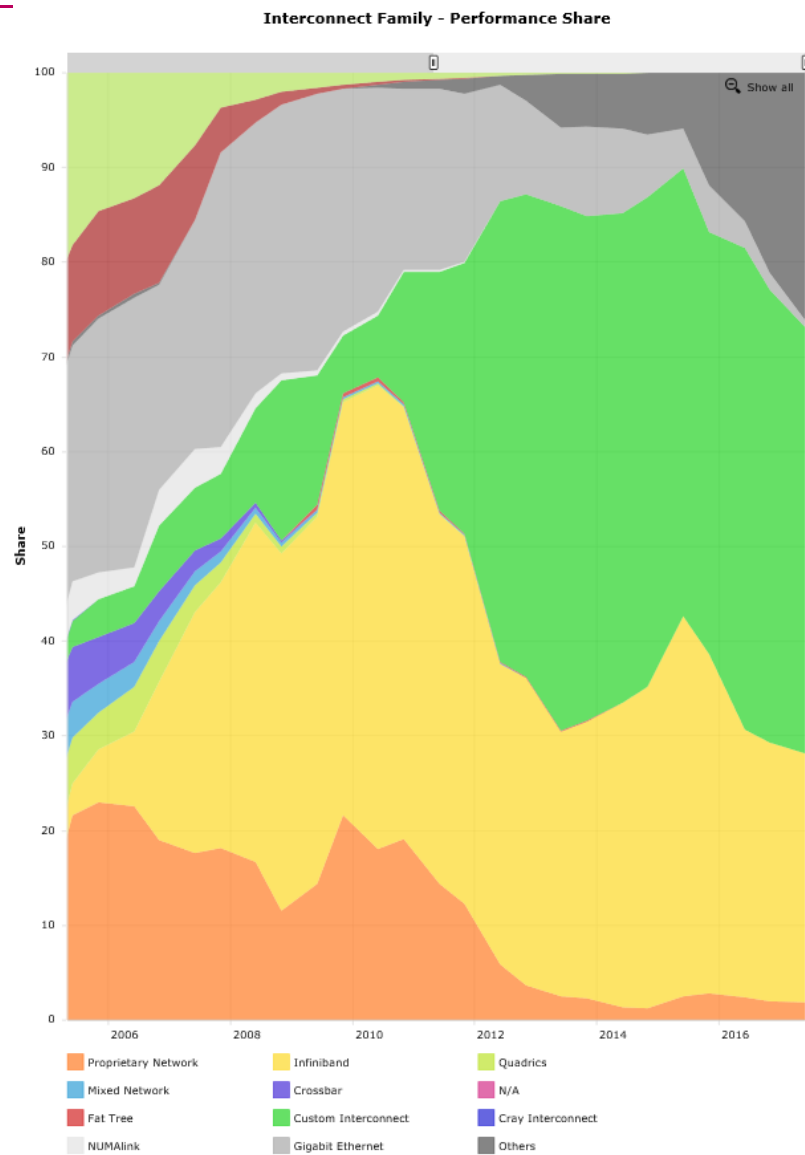
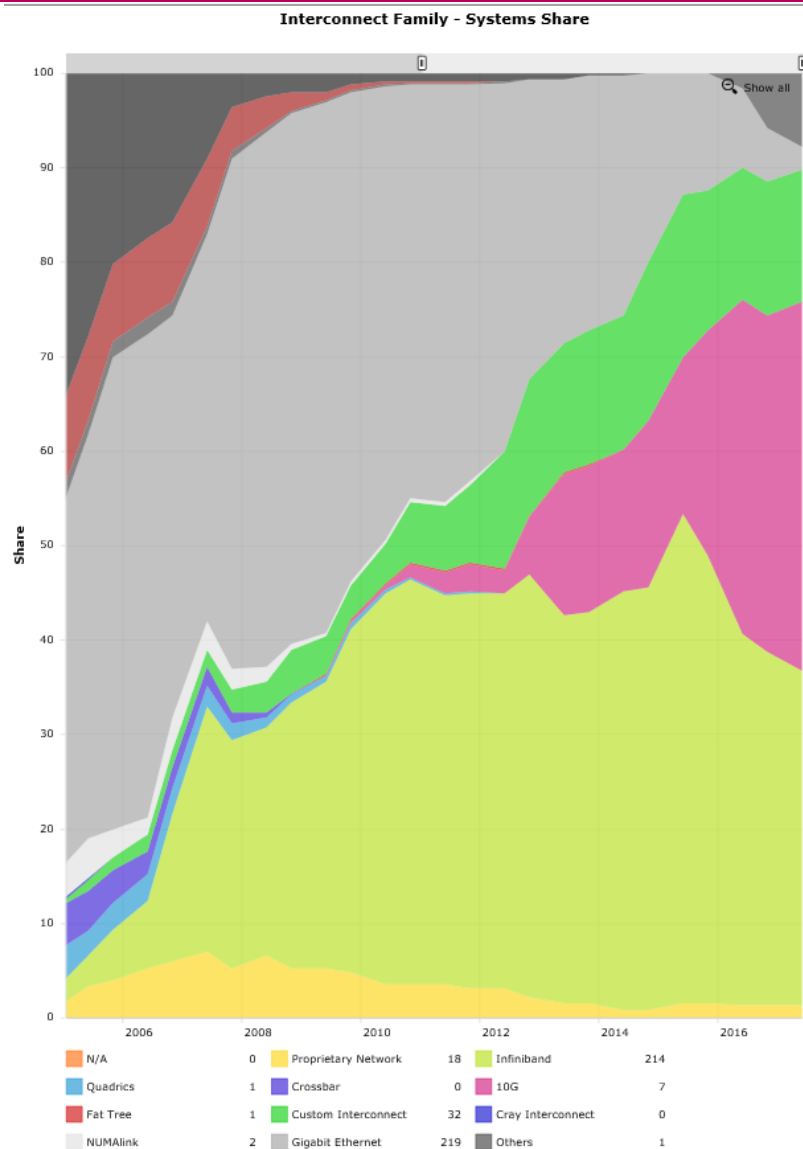
- **InfiniBand架构**
 - 主机通道适配器**HCA**(Host Channel Adapters): 连接内存控制器和TCA
 - 目标通道适配器**TCA**(Target Channel Adapters): 将I/O设备（如网卡、SCSI控制器）的数字信号打包发送给HCA
 - **IBLink**: 连接HCA和TCA的交换机以及路由器
- **拓扑**
 - 不规则
 - 规则: 胖树 (Fat Tree)
- 物理层基于**IEEE 802.3.z**标准, 与**10G**以太网一样
- 三种连接速率(**1x**, **4x**, **12x**), 全双工, 是基本传输速率**2.5Gb/s**的倍数, 即支持速率是**2.5Gb/s**、**10Gb/s**和**30Gb/s**

标准互联网络的比较

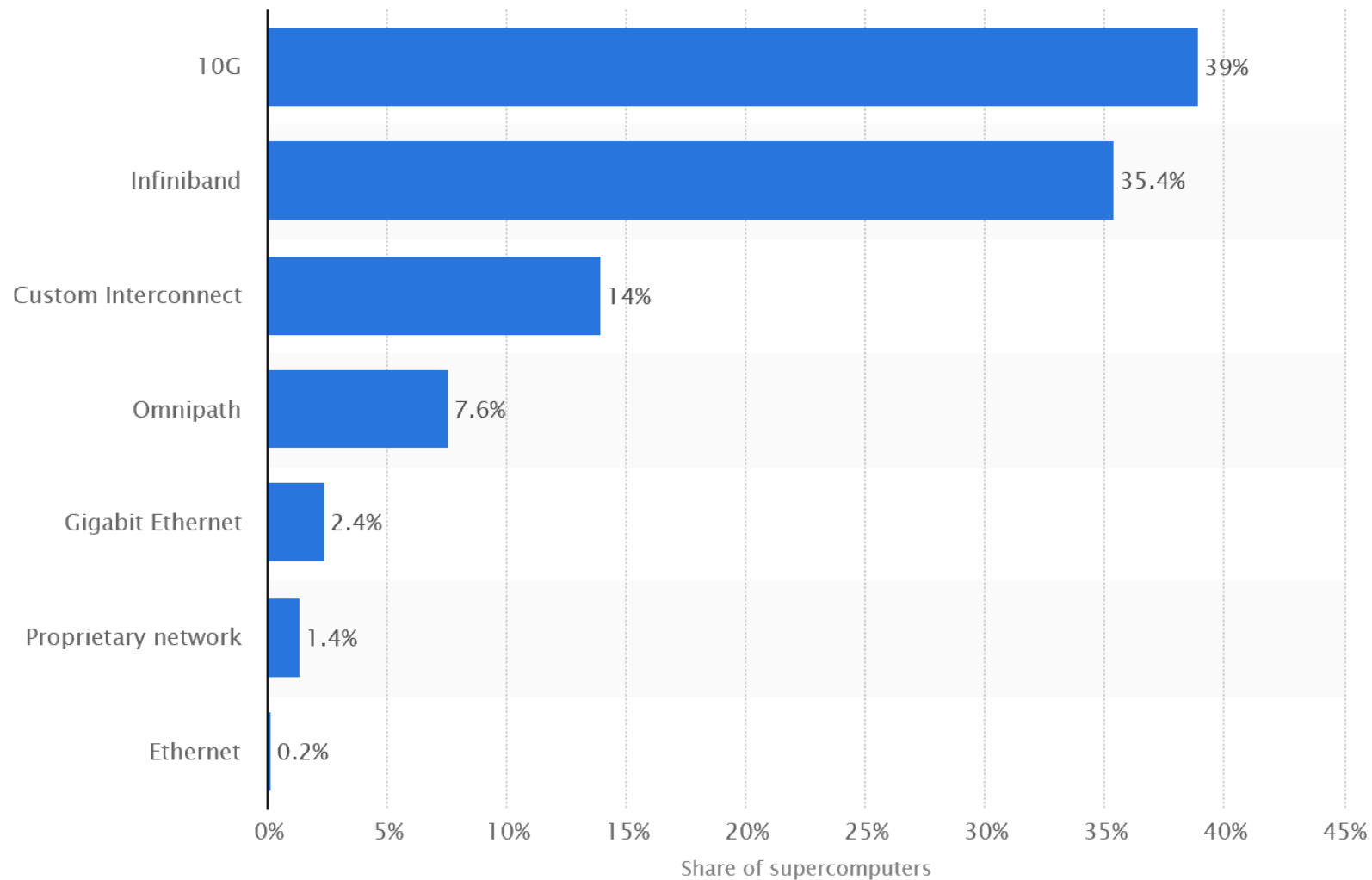
网络	接口界面	带宽 (MB/s)	MPI延时 (μ s)	备注
Gigabit Ethernet	PCI	125	80	
10G Ethernet	PCI Express	1215	2.2	Myri-10G
Myrinet 2000	PCI-X	247	2.6~3.2	Myrinet 2000
InfiniBand	PCI Express	1400	1.29	Qlogic QLE7240

数据来源: myricom, Dolphin, Qlogic 公司网站

互联技术在Top500上的比较（2017.6）



Top 500 (Interconnect Family, June 2017)



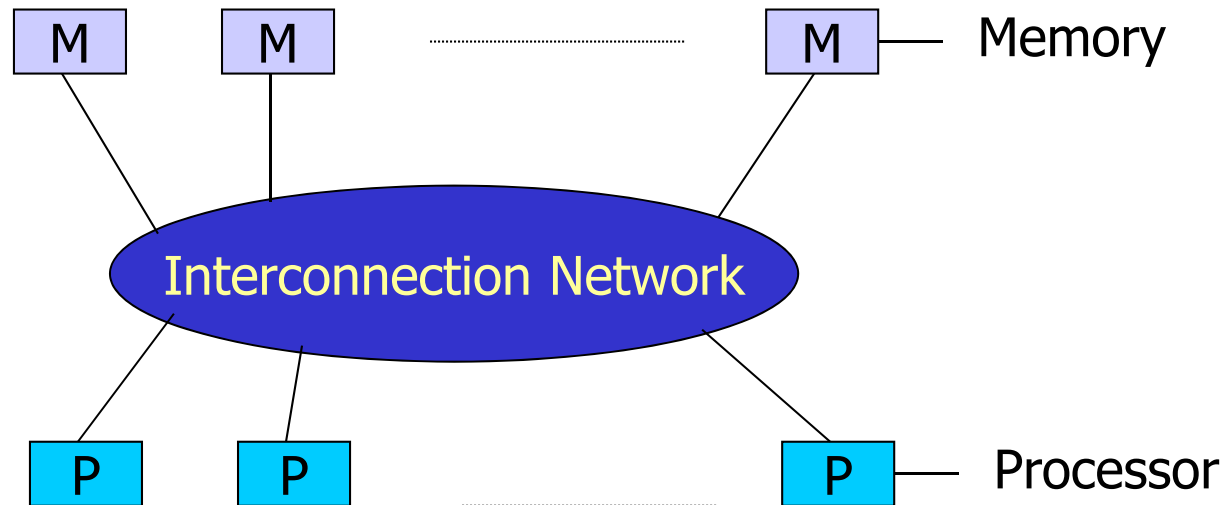
主要内容

- 互联网络
- 存储模型
 - 访存模型
 - 存储组织

MIMD结构

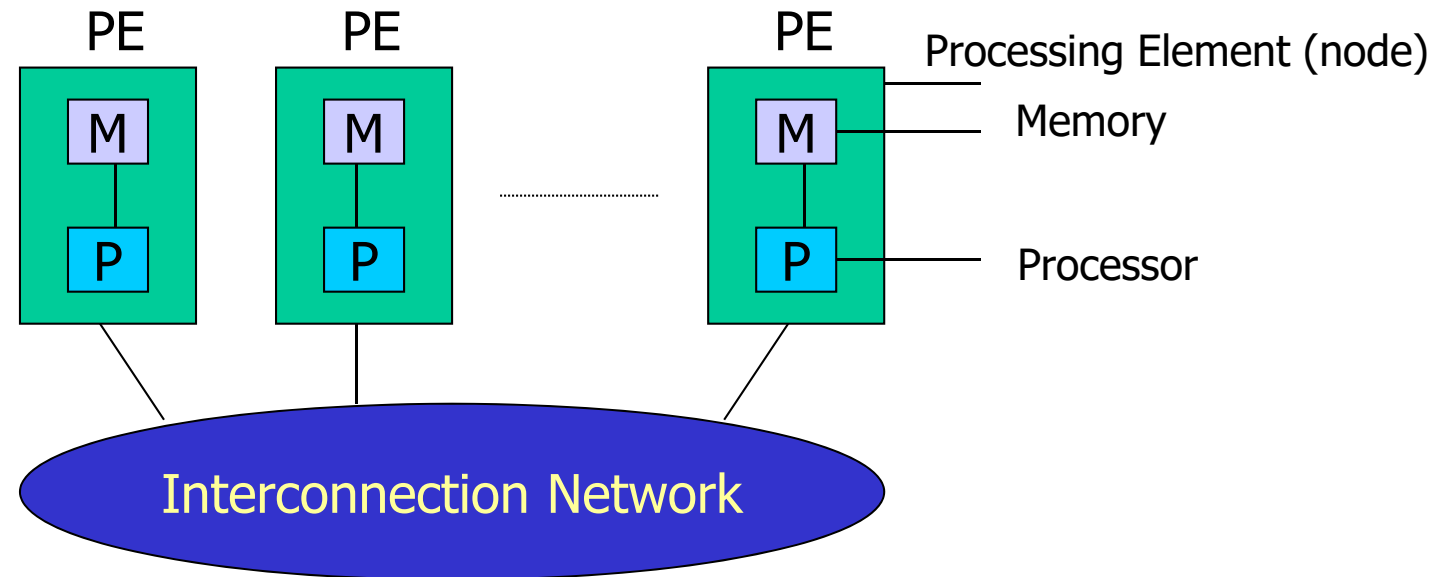
- 共享存储（**Shared Memory**） **MIMD / Multiprocessor**
- 分布式存储（**Distributed Memory**） **MIMD / 消息传递（Message Passing） MIMD / Multicomputer**

共享存储



- 单一地址空间（**Single address space**）：存储模块定义了一个可在处理器间共享的单一地址空间
- 任何处理器可以通过互联网存取任何存储模块
- 例子： **SGI Origin, Sun E10000**

分布式存储

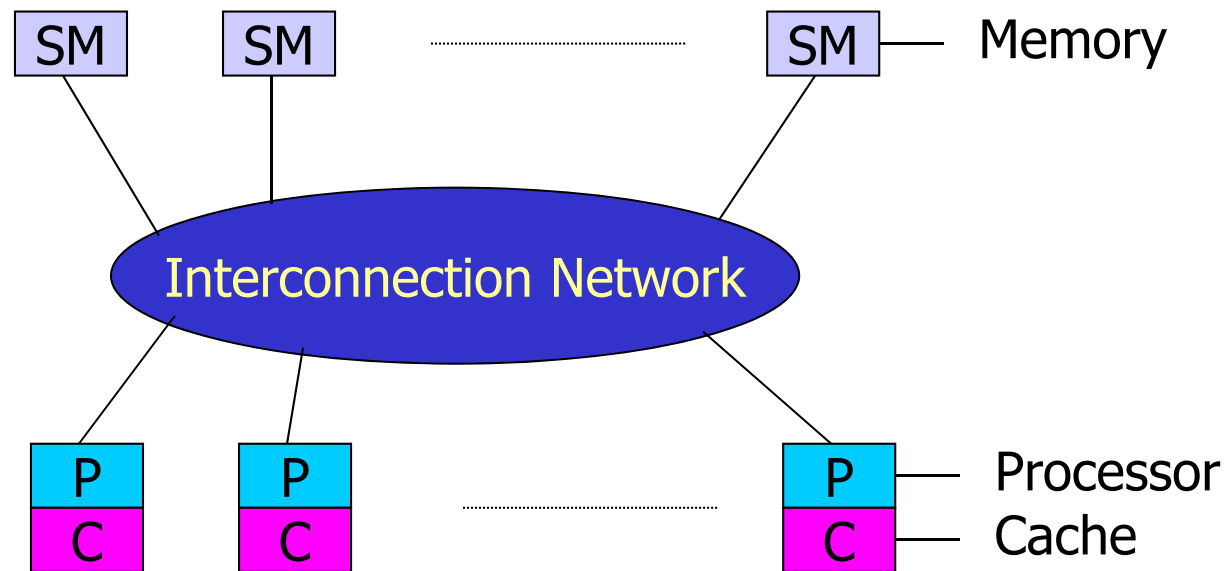


- 处理器单元（**PE**）独立工作，每个处理器有自己本地存储
- 通过消息传递（**message passing**）来交互。**PE**不能直接存取其他**PE**的内存，必须通过消息传递来交换处理器之间的数据
- 例子： **CRAY T3E**, **IBM SP**, 集群（**Cluster**）

存储器结构分类

- 集中式存储器
 - **UMA (Uniform Memory Access)**
- 分布式存储器
 - **NUMA (Non-Uniform Memory Access)**
 - ✓ **NCC-NUMA (Non-Cache Coherent NUMA)**
 - ✓ **COMA (Cache Only Memory Architecture)**
 - ✓ **CC-NUMA (Cache Coherent NUMA)**
 - **NORMA (No-Remote memory Access)**

UMA



- **UMA (Uniform Memory Access)**
- **例子: Pentium Pro Quad, Sun Starfire**

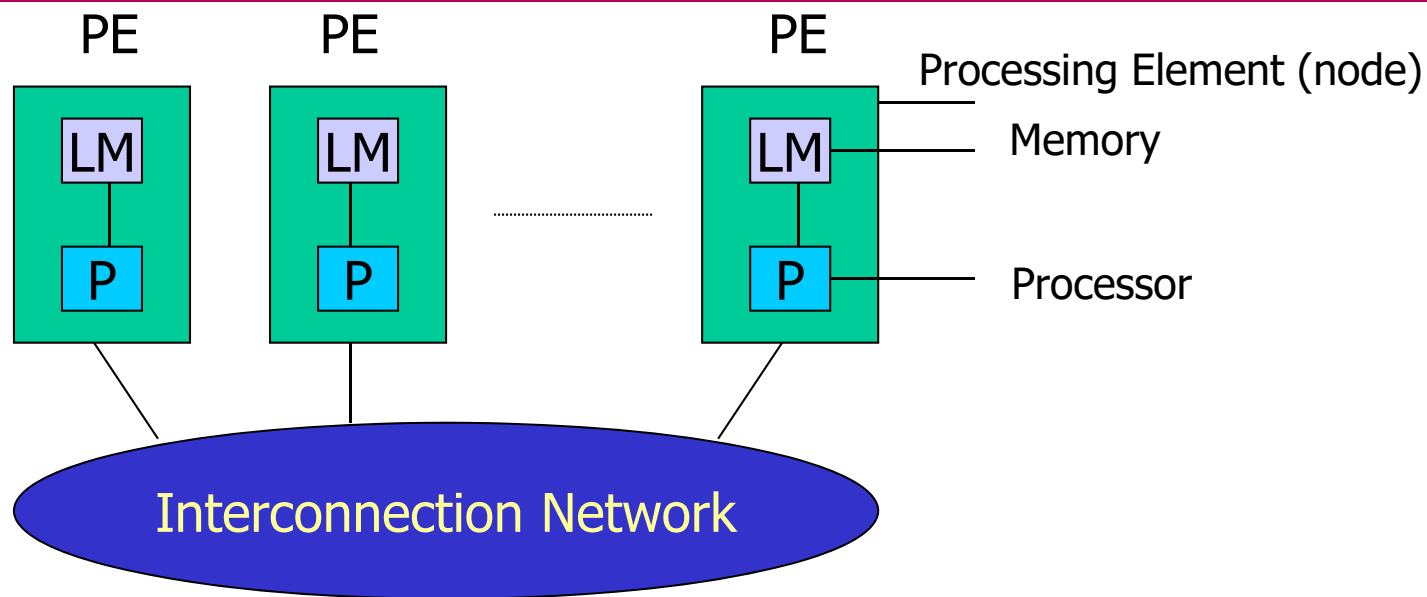
UMA

- **UMA模型是均匀存储访问模型的简称。其特点是：**
 - 物理存储器被所有处理器**均匀共享**
 - 所有处理器访问任何存储字取相同的时间
 - 每台处理器可带私有高速缓存
 - 外围设备也可以一定形式共享

可扩展的共享存储

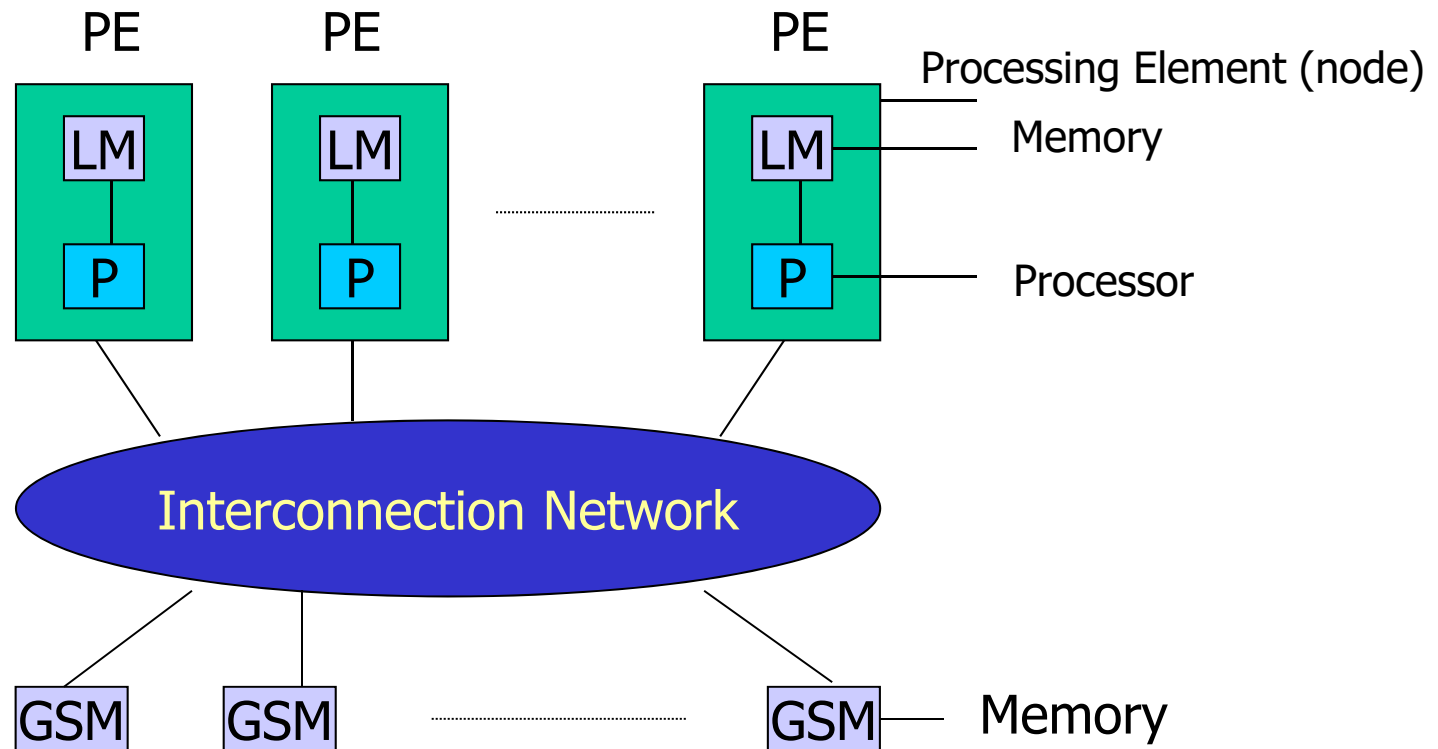
- 通过高吞吐率（**High-throughput**），低时延（**low-latency**）的互联网络
- 每个节点有一个本地缓存或存储
 - 存储缓存一致性问题（**Cache coherence problem**）
- 逻辑共享的存储可以通过一系列本地存储来实现
 - 分布式共享存储**MIMD: NUMA**

分布式共享存储（DSM）



- **分布式共享存储（Distributed Shared Memory）**和分布式存储（**Distributed Memory**）：物理结构是一样的
- 分布式共享存储：本地存储是全局地址空间的组成部分，任何处理器可直接存取其他处理器的本地存储
- 分布式存储：本地存储有独立的地址空间，不能直接存取远程处理器的存储空间

NUMA

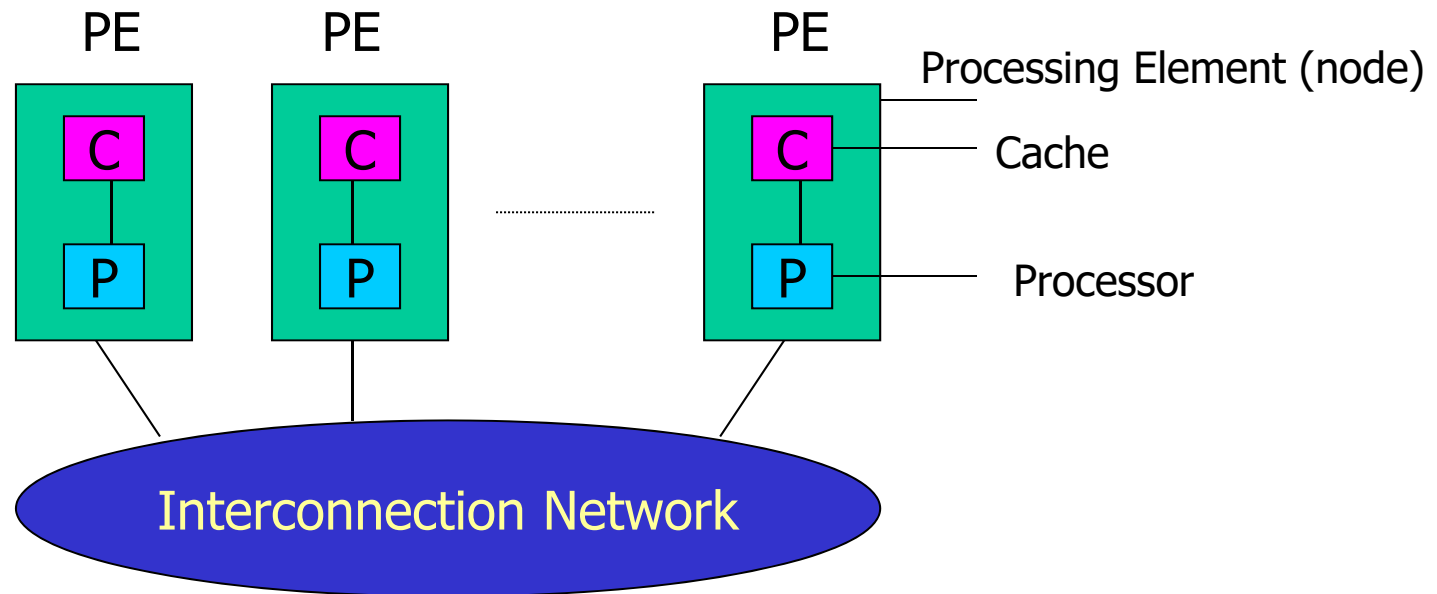


- **NUMA (Non-Uniform Memory Access)**
- **例子: Cray T3E, SGI Origin**

NUMA

- NUMA（NCC-NUMA）模型是**非均匀存储访问**模型的简称
 - 被共享的存储器在**物理上**是**分布**在所有的处理器中的，其所有本地存储器的集合就组成了全局地址空间
 - 处理器**访问**存储器的**时间是不一样的**；访问本地存储器（LM）较快，而访问外地的存储器或全局共享存储器（GSM）较慢（此即非均匀存储访问名称的由来）
 - 每台处理器可带私有高速缓存，外设也可以某种形式共享

COMA

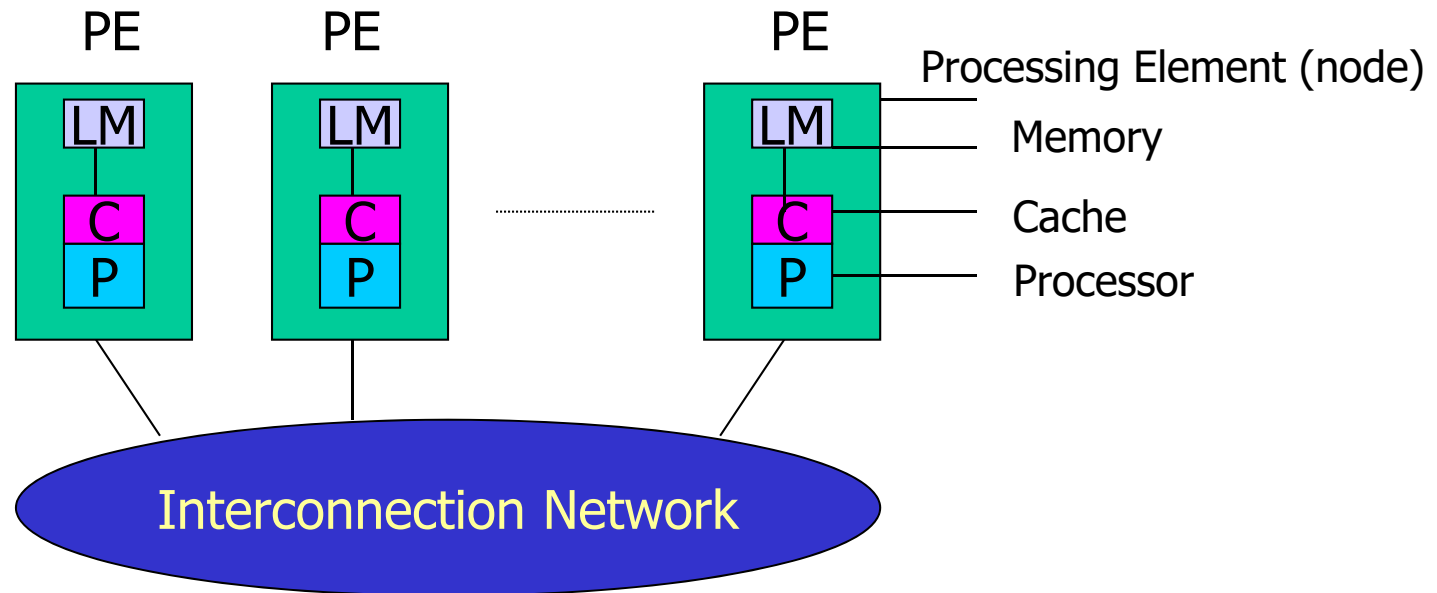


- **COMA (Cache Only Memory Architecture)**
- **例子: Kendall Square. Research公司的KSR - 1**

COMA

- COMA模型是高速缓存存储访问模型的简称，其特点是：
 - 各处理器节点中没有存储层次结构，全部高速缓存组成了全局地址空间
 - 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都大于2级高速缓存容量
 - 使用COMA时，数据开始时可任意分配，因为COMA中没有物理地址，数据可动态迁移
 - ✓ 经过“预热”，数据将被“吸引”到处理节点附近

CC - NUMA



- **CC-NUMA (Cache Coherent Non Uniform Memory Access)**
- **例子: SGI Origin, Stanford DASH, Sequent NUMA-Q**

CC-NUMA

- **CC-NUMA模型是高速缓存一致性非均匀存储访问模型的简称，其特点是：**
 - 大多数使用基于目录的**高速缓存一致性**协议；
 - 保留**SMP**结构易于编程的优点，也改善常规**SMP**的可扩展性；
 - **CC-NUMA**实际上是一个分布共享存储的**DSM**多处理机系统；
 - 对高速缓存一致性提供硬件支持
 - 它最显著的优点是程序员无需明确地在节点上分配数据，系统的硬件和软件开始时自动在各节点分配数据，在运行期间，高速缓存一致性硬件会自动地将数据迁移至要用到它的地方。

NORMA(No-Remote memory Access)

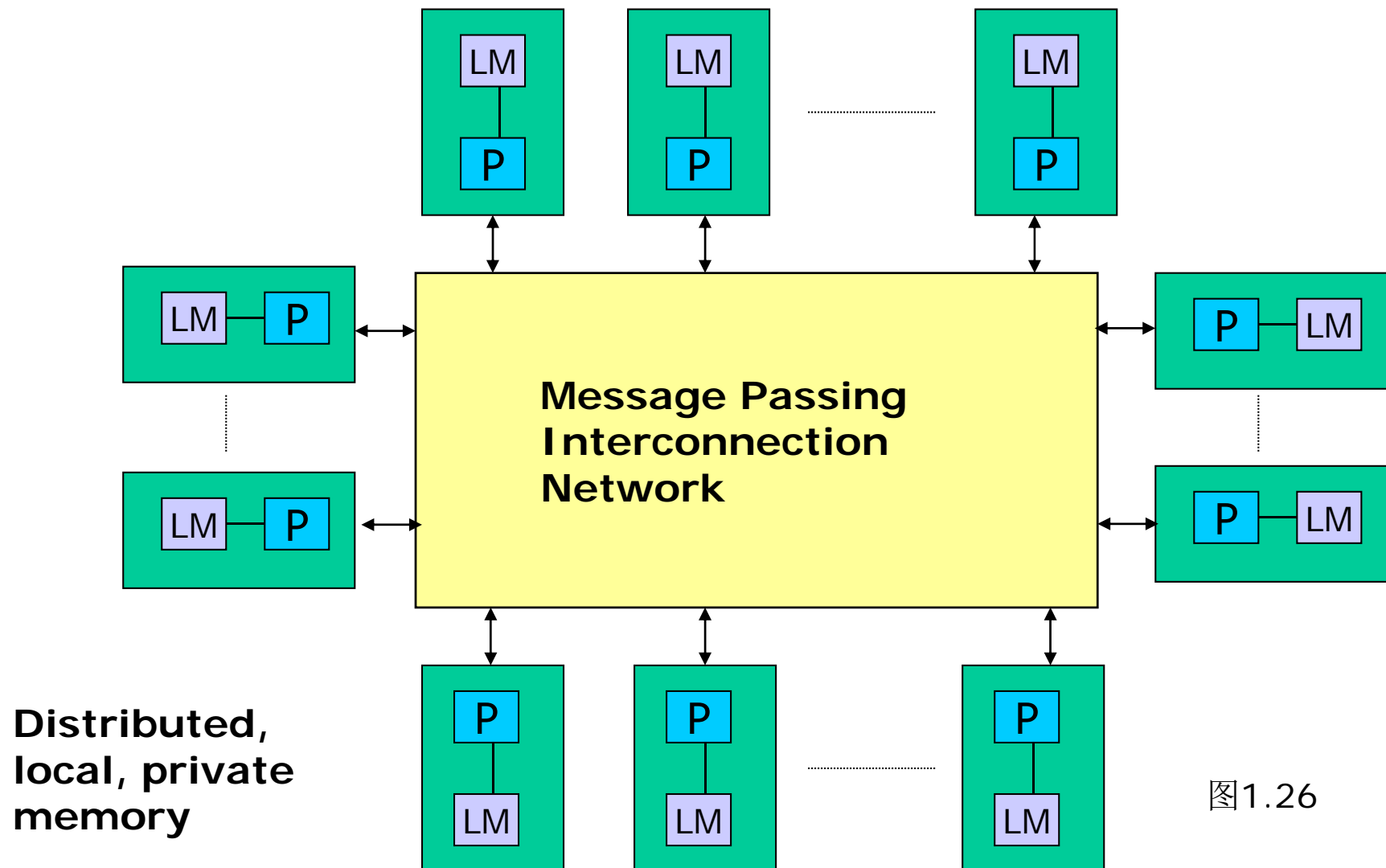
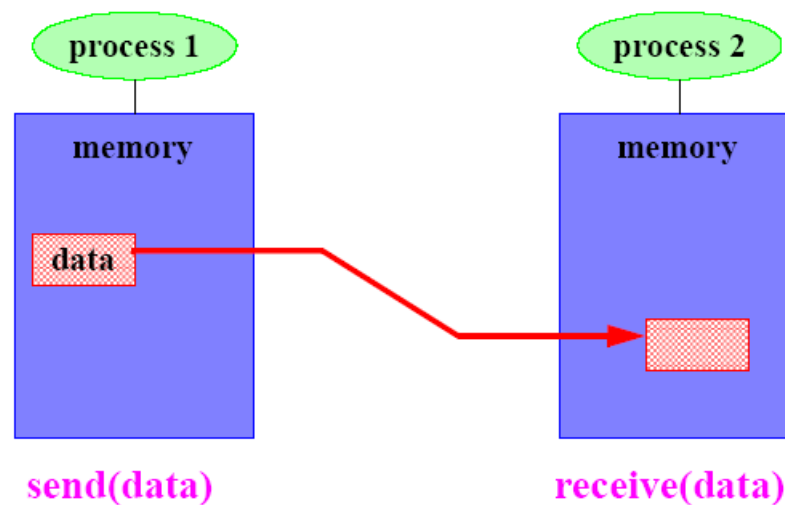


图 1.26

NORMA

- NORMA模型是**非远程存储访问**模型的简称，其特点是：
 - 所有存储器是私有的
 - 节点不能访问远程存储器，而必须通过**消息传递**方式



构筑并行机系统的不同存储结构

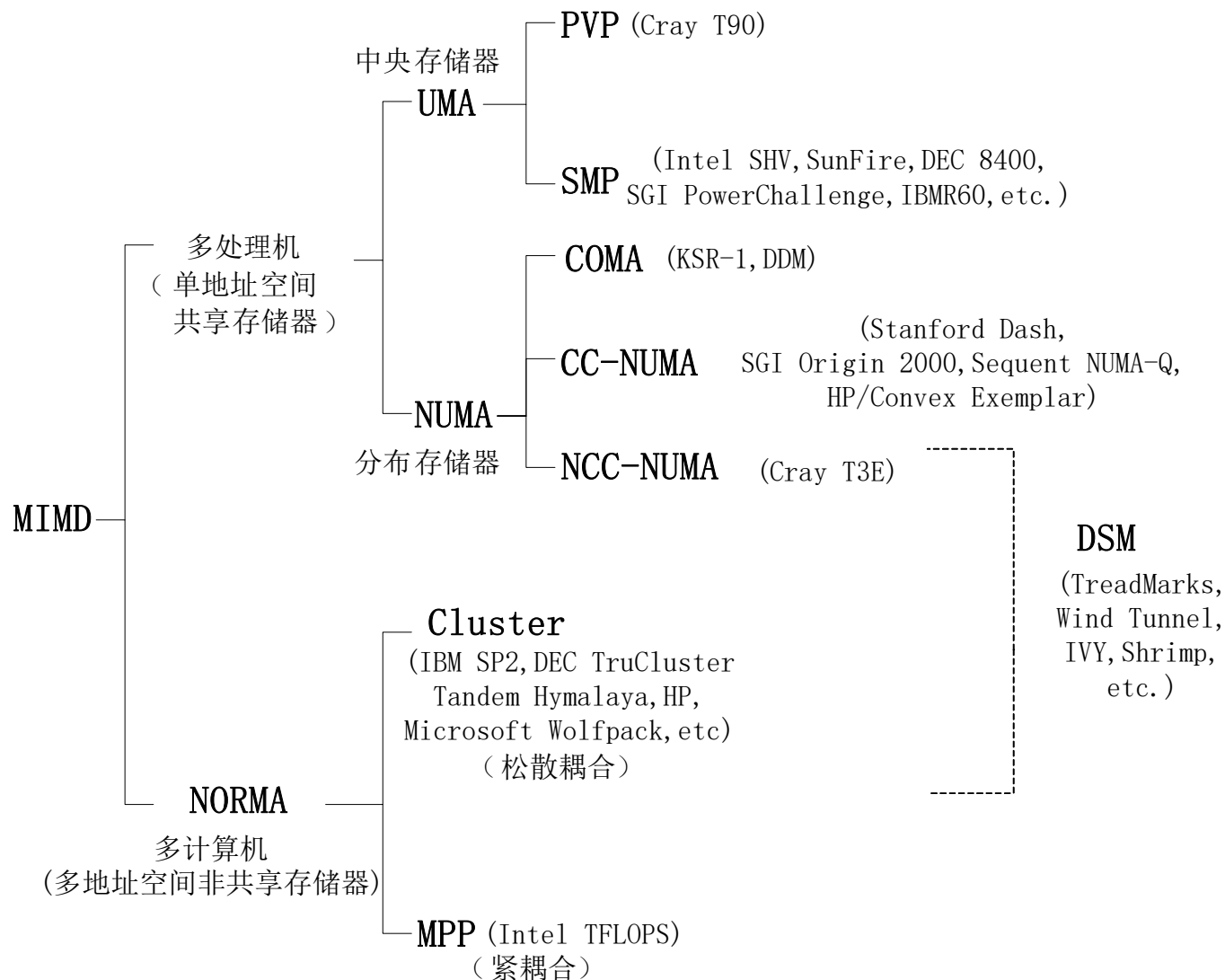


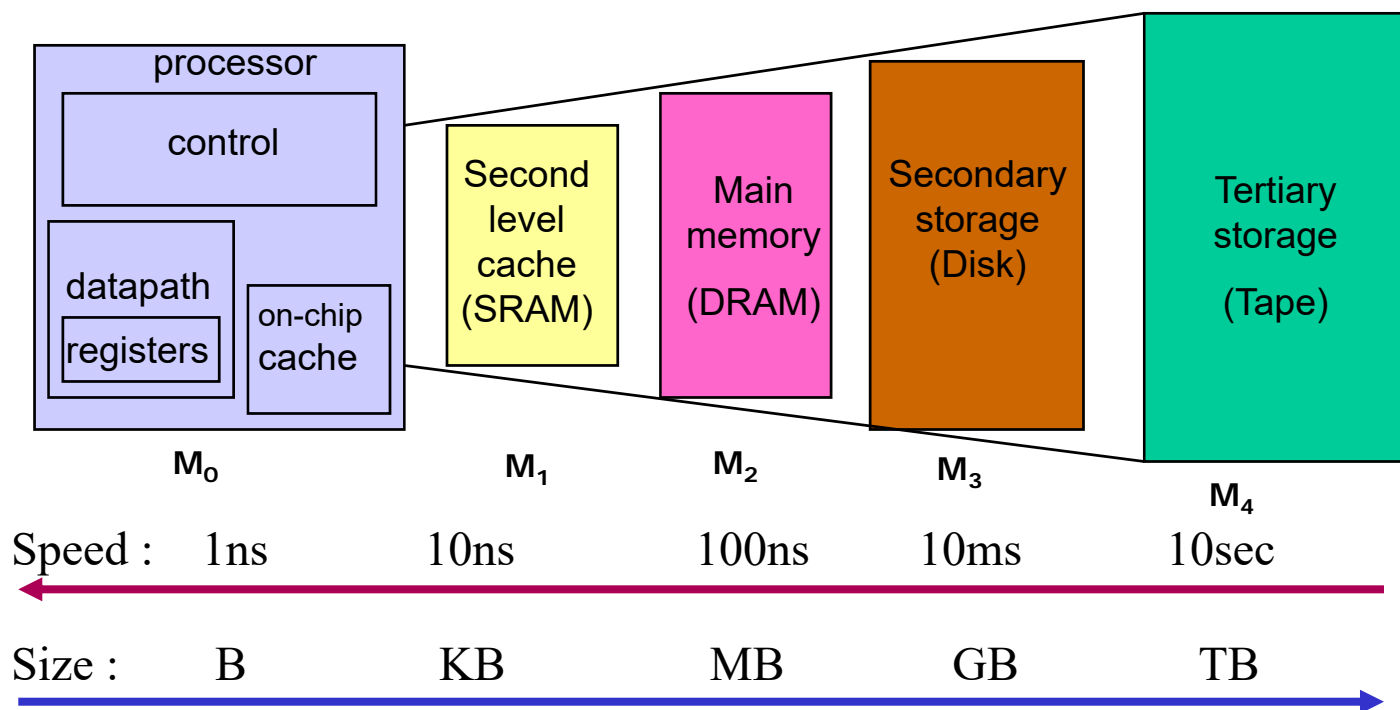
图1.27

主要内容

- 互联网络
- 存储模型
 - 访存模型
 - 存储组织

存储层次结构

- 弥补CPU与主存间的速度差异（不平衡）
- 各个层次间的访问速度和容量差别



存储层次结构的参数

- 参数:

- 容量 (Capacity) : C
- 时延 (Delay) : L
- 带宽 (Bandwidth) : B

远程存储 (Remote memory) : 通过互联网访问

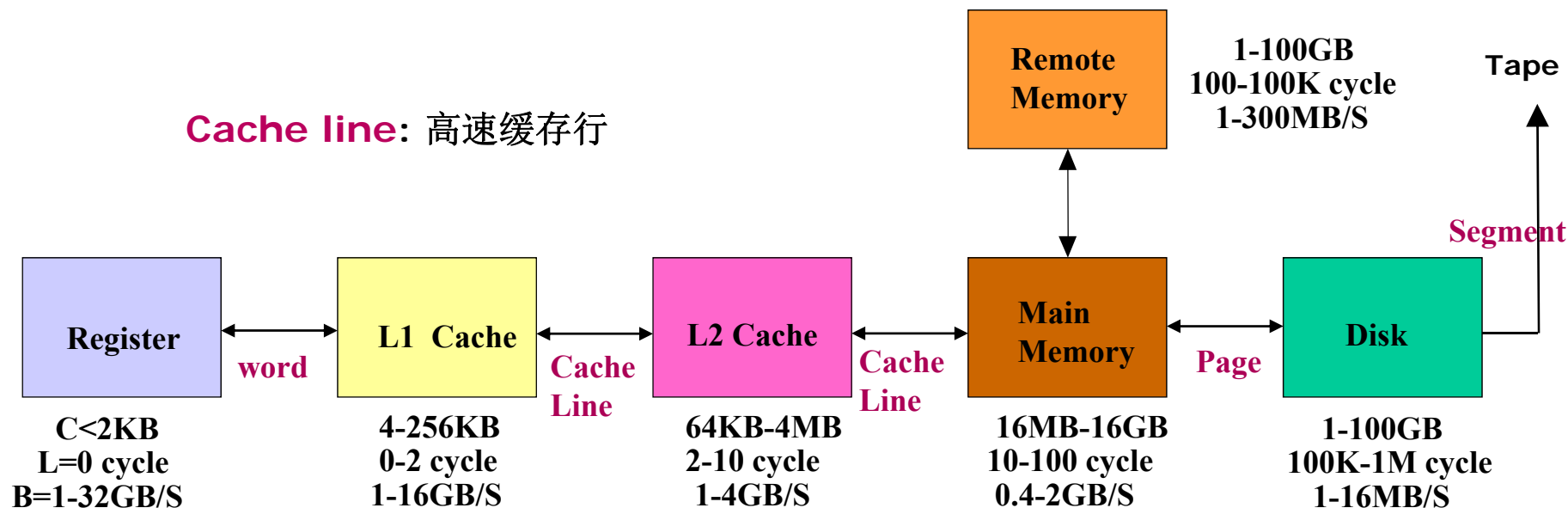


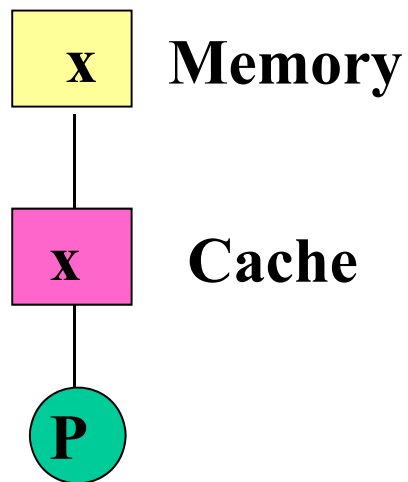
图3.1

例子：Intel Sandy Bridge Core i3 i5 i7

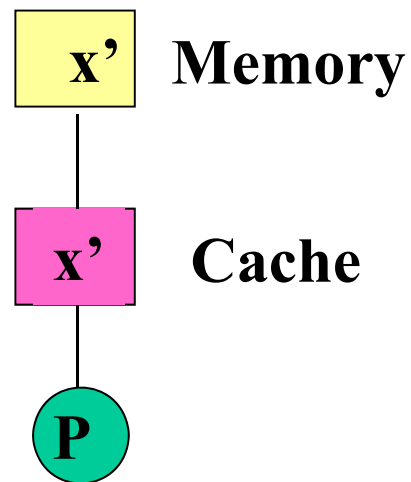
- 32KB L1缓存， 256KB L2缓存和8MB L3共享缓存，可以提供高达25.6GB/s的双通道数据传输带宽



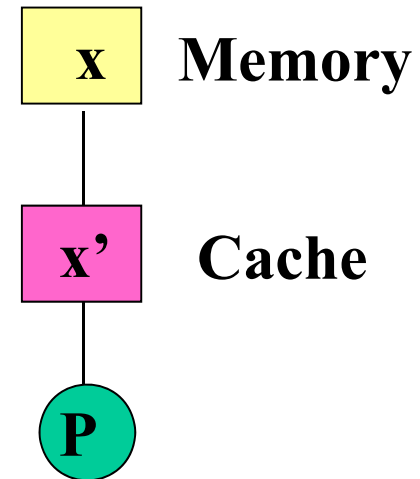
高速缓存一致性 (Cache Coherence)



之前



写直达WT
(Write Through)




写回WB
(Write Back)

商业系统通常采用write-back.

缓存不一致的问题

Time	Event	Cache for A	Cache for B	Memory for x
0				1
1	CPU A reads X	1		1
2	CPU B reads X	1	1	1
3	CPU A writes 0 to X	0	1	0

invalid value

- 出现**Cache**不一致的原因主要有以下三个方面
 - 共享可写数据
 - 多处理机的进程迁移（图1.38）
 - 绕过**Cache**的I/O操作（图1.39）

解决缓存不一致的方法

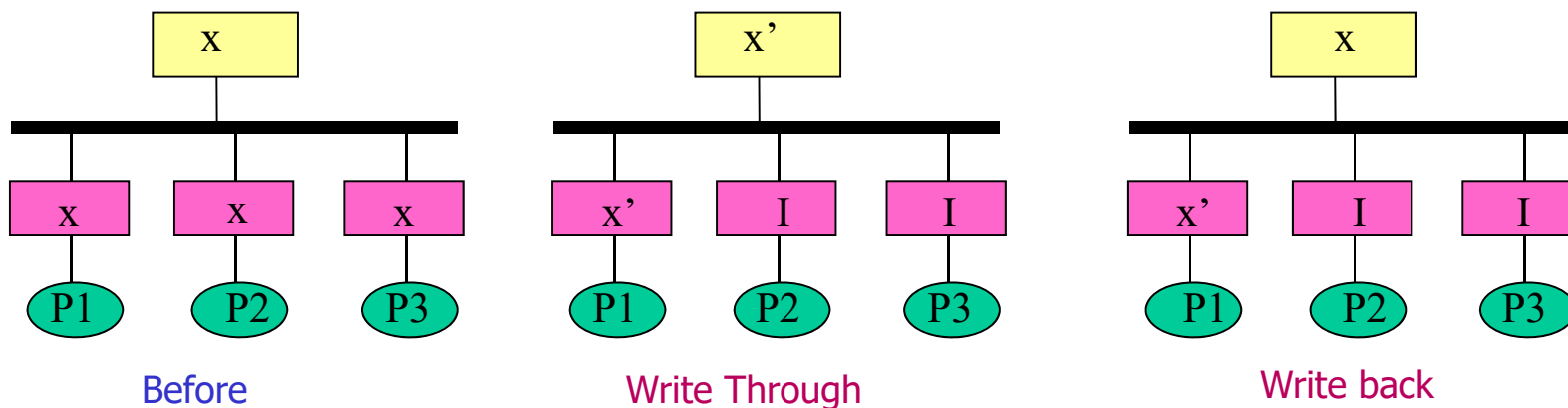
- 总线侦听 (**Bus snooping**)
 - 所有处理器监听共享总线获知写操作，以修改本地缓存
- 基于目录 (**Directory-based**) 的协议
 - 在主存中设置一个目录表，记录所有高速缓存的位置和状态
 - 发送消息使得远程缓存无效或者被更新

监听协议

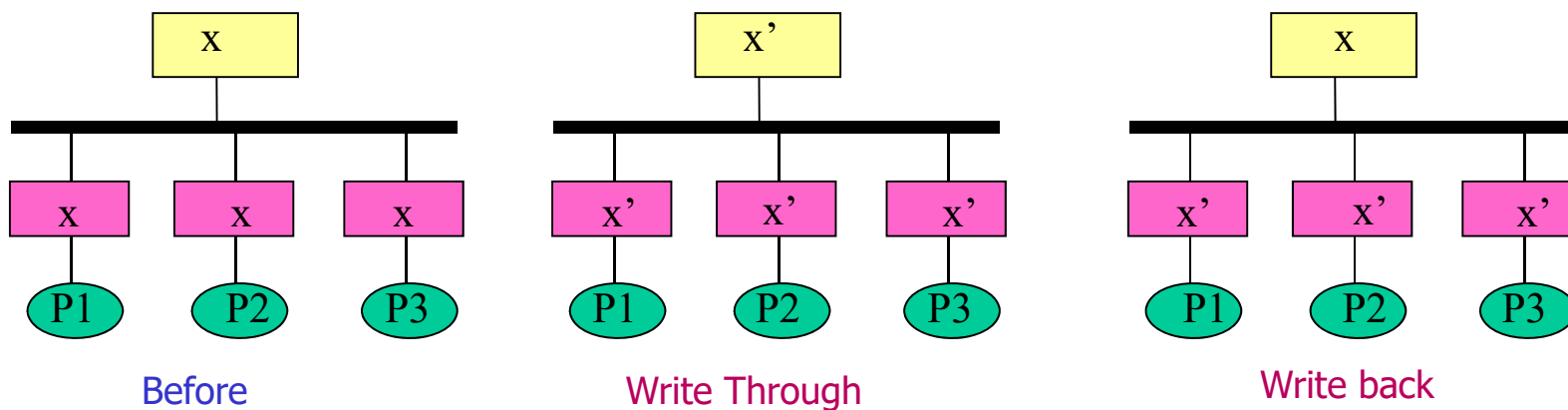
- 基于总线的方法。缓存控制器（**cache controllers**）监听总线的操作，以更新或无效缓存块。
- 两种更新策略
 - 写无效（**Write invalidate**）
 - 写更新（**Write-update**）
- 商业系统通常采用 **write-invalidate**
 - 来节省带宽

两种类型的写协议

- 写无效（Write-invalidate）

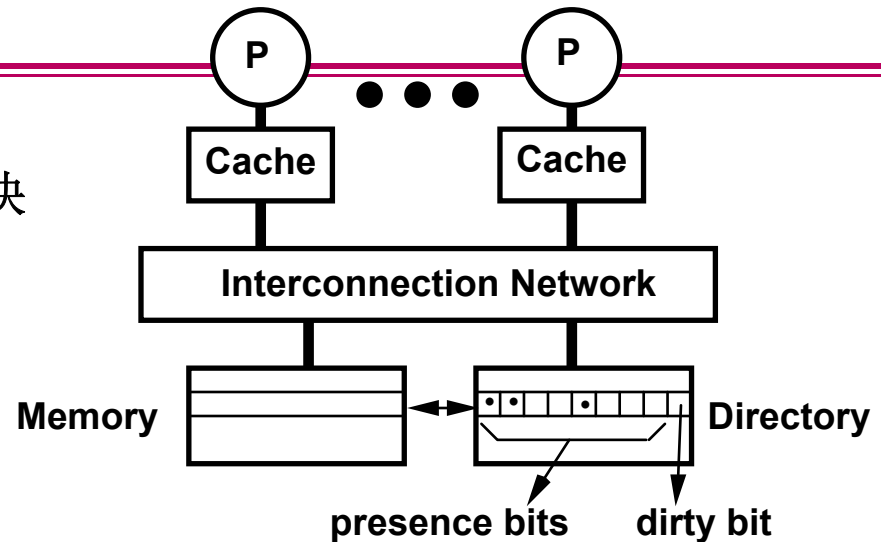


- 写更新（Write-update）



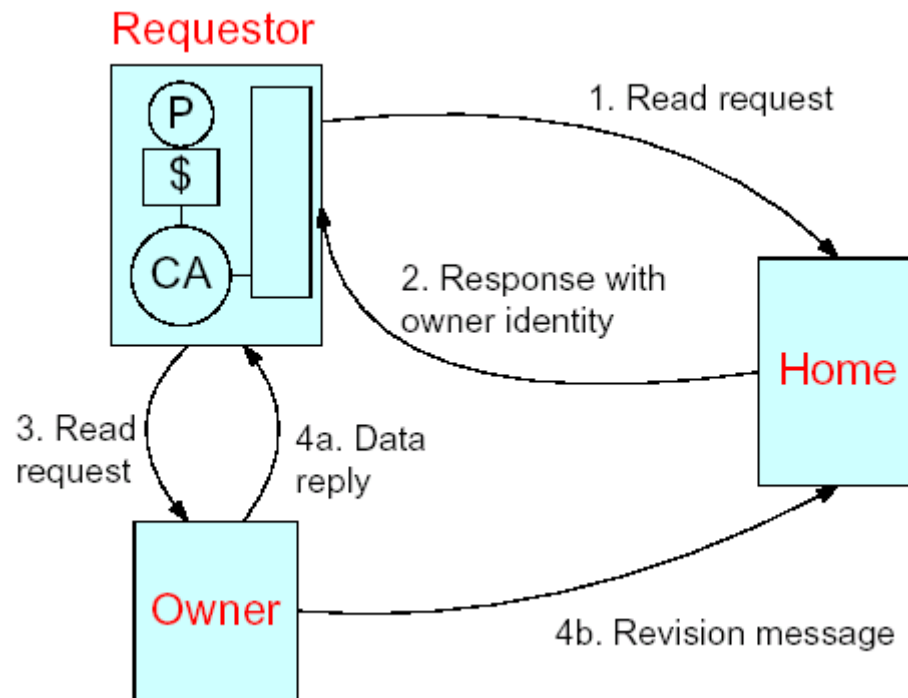
基于目录的协议

- k 个处理器
- 在主存中维护一个目录表，对每个缓存块，有 k 个**presence-bits**（是否存在），1个**dirty-bit**

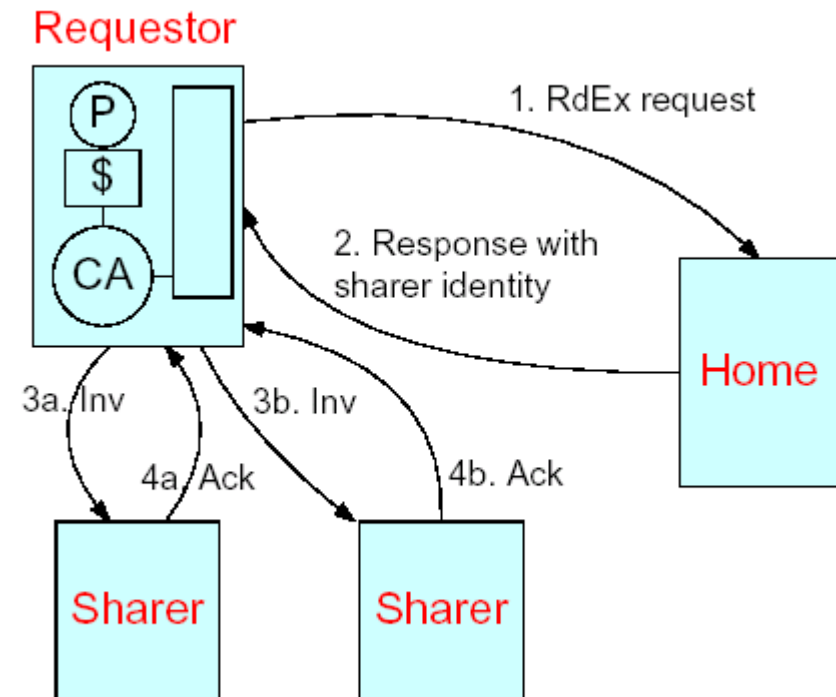


- 第 i 个处理器从主存读取数据：
 - 如果主存的 **dirty-bit**是**OFF**，则直接从主存读取，并令 $p[i]$ 为**ON**;
 - 如果主存的 **dirty-bit**是**ON**，则 只有一个处理器的存在位为“1”，从该处理器读取数据到主存，将**dirty-bit**设为**OFF**;并令 $p[i]$ 为**ON**;将数据返回给处理器 i ;
- 第 i 个处理器写数据到主存：
 - 如果主存的 **dirty-bit**是**OFF**，则使得包含该数据的所有缓存块为无效，清除所有 $p[k]$;将**dirty-bit**设为**ON**;并令 $p[i]$ 为**ON...**;
 - 如果主存的 **dirty-bit**是**ON**?

基本目录事务



Read Miss



Write Miss

- 主节点 (**Home node**) : 包含主存储区的节点
- 所有者节点 (**Owner node**) : 要提供数据的节点

讨论

- 多处理器（**Multiprocessor**）和多计算机（**Muticomputer**）
 - 体系结构
 - 存储
 - 通讯
- 请列出以下存储模型和体系结构的独特之处
 - **Parallel system architecture (PVP,SMP,MPP,DSM,COW)**
 - **Memory models (UMA,NUMA,CC-NUMA,NCC-NUMA,COMA,NORMA)**

总结 (1)

Features	PVP	SMP	MPP	DSM	COW
Architecture	MIMD	MIMD	MIMD	MIMD	MIMD
Processor Type	Customer-Designed	Commercial	Commercial	Commercial	Commercial
Interconnection Network	Customer-Designed Crossbar Switcher	Bus, Crossbar Switcher	Customer-Designed	Customer-Designed	Commercial Network (eg. Ethernet)
Comunication	Shared Variable	Shared Variable	Message Passing	Shared Variable	Message Passing
Address Space	Single	Single	Multiple	Single	Multiple
Memory Access	Shared	Shared	Distributed	Distributed Shared	Distributed
Memory Model	UMA	UMA	NORMA	NUMA	NORMA
Example Machine	Cray C-90, Cray T-90, 银河1号	IBM R50, SGI Power, Sun Starfire, 曙光1号	Intel Paragon, IBM Option White, 曙光 1000/2000	Standford DASH, Cray T3D	Berkeley NOW

总结（2）

- **SMP、MPP、DSM和COW并行结构渐趋一致**
 - 大量的节点通过高速网络互连起来
 - 节点遵循Shell结构：用专门定制的Shell电路将商用微处理器和节点的有关部分（包括板级Cache、局存、NIC和DISK）连接起来。优点是CPU升级只需要更换Shell。

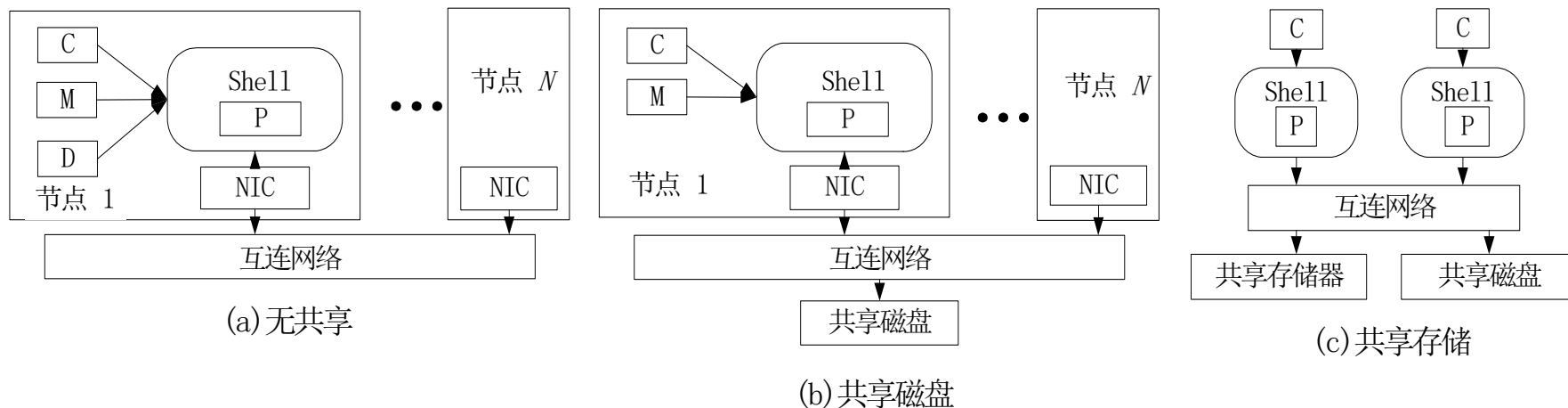


图1.21

课程小结

- 网络环境
 - Intra-node Interconnections(Buses , Switches)
 - Inter-node Interconnections (SAN)
 - Inter-system Interconnections(LAN , MAN , WAN)
- 互联网络结构
 - Static-Connection Networks(LA,RC,MC,TC,HC,CCC)
 - Dynamic-Connection Networks (Buses, Crossbar, MIN)
 - Standard Networks (Myrinet, Ethernet, Infiniband)
- Parallel Computer Memory Access Models
 - UMA : Uniform Memory Access
 - NUMA : Non-Uniform Memory Access
 - COMA : Cache-Only Memory Access
 - NORMA : NO-Remote Memory Access

推荐网站和读物

- 《并行计算—结构、算法、编程》
 - 第1章：并行计算机系统及其结构模型
- 陈国良等，《并行计算机体系结构》，高等教育出版社，2002

下一讲

- 并行计算机系统及性能评测
 - 《并行计算—结构、算法、编程》第2，3章