

计算机系统结构

第1章 计算机系统结构的基础知识

目录

- 1.1 [计算机系统结构的基本概念](#)
- 1.2 [计算机系统的设计](#)
- 1.3 [计算机系统的性能评测](#)
- 1.4 [计算机系统结构的发展](#)
- 1.5 [计算机系统结构中并行性的发展](#)

1.3 计算机系统的性能评测

1. 执行时间和吞吐率

如何评测一台计算机的性能，与测试者看问题的角度有关。

➤ 用户关心的是：

单个程序的**执行时间**（Execution Time）。

响应时间（Response Time）--完成一个任务的全部时间。

➤ 数据处理中心的管理员关心的是：

吞吐率（Throughput）-- 单位时间内完成的任务数。

1.3 计算机系统的性能评测

假设两台计算机为X和Y，X比Y快的意思是：

对于给定任务，X的执行时间比Y的执行时间少。

X的性能是Y的n倍：

$$\frac{\text{执行时间Y}}{\text{执行时间X}} = n$$

执行时间与性能成反比：

$$n = \frac{\text{执行时间Y}}{\text{执行时间X}} = \frac{\frac{1}{\text{性能Y}}}{\frac{1}{\text{性能X}}} = \frac{\text{性能X}}{\text{性能Y}}$$

1.3 计算机系统的性能评测

- 执行时间可以有多种定义：
 - ❑ 计算机完成某一任务所花费的**全部时间**，包括磁盘访问、存储器访问、输入/输出、操作系统开销等。
 - ❑ **CPU时间**：CPU执行给定的程序所花费的时间，不包含I/O等待时间以及运行其它程序的时间。
 - **用户CPU时间**：用户程序所耗费的CPU时间。
 - **系统CPU时间**：用户程序运行期间操作系统耗费的CPU时间。

1.3 计算机系统的性能评测



其他性能评价指标

$$MIPS = \frac{\text{指令条数}}{\text{执行时间} \times 10^6} = \frac{f}{CPI \times 10^6}$$

MIPS

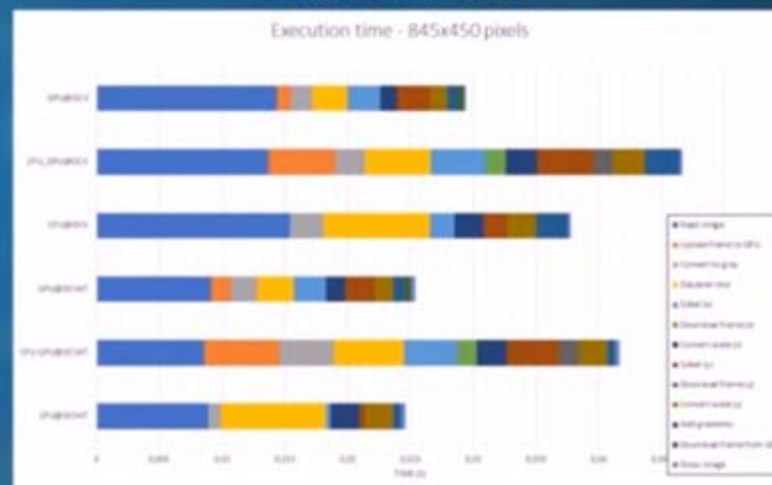


1.3 计算机系统的性能评测

其他性能评价指标

$$\text{程序执行时间} T_e = \frac{\text{指令条数}}{MIPS \times 10^6}$$

程序执行时间



The only consistent and reliable measure of performance is the execution time of real programs!---D.A.Patterson
对性能而言,唯一一致和可靠的测量是真实程序的执行时间!

1.3 计算机系统的性能评测

其他性能评价指标

$$MFLOPS = \frac{\text{程序中的浮点操作数}}{\text{执行时间} \times 10^6}$$



MFLOPS

1.3 计算机系统的性能评测

2. 基准测试程序（Benchmark）

- 用于测试和比较性能的基准测试程序的最佳选择是**真实应用程序**。
- 以前常采用简化了的程序，例如：
 - ❑ **核心测试程序**：从真实程序中选出的关键代码段构成的小程序。
 - ❑ **小测试程序**：简单的只有几十行的小程序。
 - ❑ **合成的测试程序**：人工合成出来的程序。

Whetstone与Dhrystone是最流行的合成测试程序。

从测试性能的角度来看，上述测试程序不可信了：

- ❑ 这些程序比较小，具有片面性；
- ❑ 系统结构设计者和编译器的设计者可以“合谋”把他们的机器面向这些测试程序进行优化设计，使得该机器显得性能更高。

1.3 计算机系统的性能评测

- 性能测试的结果除了和采用什么测试程序有关以外，还和**在什么条件下进行测试**有关。
 - 基准测试程序设计者对制造商的要求
 - 采用同一种语言编译器；
 - 对同一种语言的程序都采用相同的一组编译标志。
 - **问题：**是否允许修改测试程序的源程序
- 三种不同的处理方法：
- 不允许修改；
 - 允许修改，但因测试程序很复杂或者很大，几乎是无法修改。
 - 允许修改，只要保证最后输出的结果相同。

1.3 计算机系统的性能评测

- **基准测试程序套件：**由各种不同的真实应用程序构成。
（能比较全面地反映计算机在各个方面的处理性能）
- **SPEC系列：**最成功和最常见的测试程序套件
（美国的标准性能测试公司创建）
 - 桌面计算机的基准测试程序套件可以分为两大类：
处理器性能测试程序，图形性能测试程序
 - **SPEC89：**用于测试处理器性能。10个程序（4个整数程序，6个浮点程序）
 - 演化出了4个版本
 - SPEC92：**20个程序
 - SPEC95：**18个程序
 - SPEC2000：**26个程序
 - SPEC CPU2006：**29个程序

1.3 计算机系统的性能评测

- ❑ SPEC CPU2006

整数程序12个（CINT2006）

9个是用C写的，3个是用C++写的

浮点程序17个（CFP2006）

6个是用FORTRAN写的，4个是用C++写的，3个是用C写的，4个是用C和FORTRAN混合编写的。

- SPEC测试程序套件中的其它一系列测试程序组件

- ❑ **SPECSFS**: 用于NFS（网络文件系统）文件服务器的测试程序。它不仅测试处理器的性能，而且测试I/O系统的性能。它重点测试吞吐率。
- ❑ **SPECWeb**: Web服务器测试程序。
- ❑ **SPECviewperf**: 用于测试图形系统支持OpenGL库的性能。
- ❑ **SPECapc**: 用于测试图形密集型应用的性能。

1.3 计算机系统的性能评测

3. 性能比较

两个程序在A、B、C三台机器上的执行时间

	机器A	机器B	机器C	W (1)	W (2)	W (3)
程序1	1.00	10.00	20.00	0.50	0.909	0.999
程序2	1000.00	10.00	20.00	0.50	0.091	0.001

如何比较这三台机器的性能呢？从该表可以得出：

执行程序1：

A机的速度是B机的10倍

A机的速度是C机的20倍

B机的速度是C机的2倍

执行程序2：

B机的速度是A机的100倍

C机的速度是A机的50倍

B机的速度是C机的2倍

1.3 计算机系统的性能评测

- **总执行时间：** 机器执行所有测试程序的总时间
 - B机执行程序1和程序2的速度是A机的50.05倍
 - C机执行程序1和程序2的速度是A机的25.025倍
 - B机执行程序1和程序2的速度是C机的2倍
- **平均执行时间：** 各测试程序执行时间的算术平均值

$$S_m = \frac{1}{n} \sum_{i=1}^n T_i$$

其中： T_i ： 第*i*个测试程序的执行时间

n ： 测试程序组中程序的个数

1.3 计算机系统的性能评测

- **加权执行时间：** 各测试程序执行时间的加权平均值

$$A_m = \sum_{i=1}^n W_i \cdot T_i$$

其中， W_i ：第*i*个测试程序在测试程序组中所占的比重

$$\sum_{i=1}^n W_i = 1$$

T_i ：该程序的执行时间

1.3 计算机系统的性能评测

3. 性能比较

两个程序在A、B、C三台机器上的执行时间

	机器A	机器B	机器C	W (1)	W (2)	W (3)
程序1	1.00	10.00	20.00	0.50	0.909	0.999
程序2	1000.00	10.00	20.00	0.50	0.091	0.001
加权算术 平均值A _m (1)	500.50	10.00	20.00	$A_m = \sum_{i=1}^n W_i T_i$		
加权算术 平均值A _m (2)	91.91	10.00	20.00			
加权算术 平均值A _m (3)	2.00	10.00	20.00			

1.3 计算机系统的性能评测

➤ 调和平均值法

如果性能是用速度（如MFLOPS）表示的，则可以用调和平均值法比较。

$$H_m = \frac{n}{\sum_{i=1}^n \frac{1}{R_i}} = \frac{n}{\sum_{i=1}^n T_i} \quad (\text{相当于速度：次/秒})$$

其中， R_i ：由 n 个程序组成的工作负荷中执行第 i 个程序的速度

$$R_i = 1/T_i$$

T_i ：第 i 个程序的执行时间

□ 加权调和平均值公式

$$H_m = \left(\sum_{i=1}^n \frac{W_i}{R_i} \right)^{-1} = A_m^{-1} = \left(\sum_{i=1}^n W_i T_i \right)^{-1}$$

1.3 计算机系统的性能评测

- **几何平均值法**：以某台计算机的性能作为参考标准，其他计算机性能则除以该参考标准而获得一个比值。

$$G_m = \sqrt[n]{\prod_{i=1}^n R_i} = \sqrt[n]{\prod_{i=1}^n \frac{1}{T_i}}$$

R_i ：由 n 个程序组成的工作负荷中执行第 i 个程序的速度

$$R_i = 1/T_i$$

\prod ：连乘

1.3 计算机系统的性能评测

- 加权几何平均值

$$G_m = \prod_{i=1}^n (R_i)^{W_i} = (R_1)^{W_1} \times (R_2)^{W_2} \times \cdots \times (R_n)^{W_n}$$

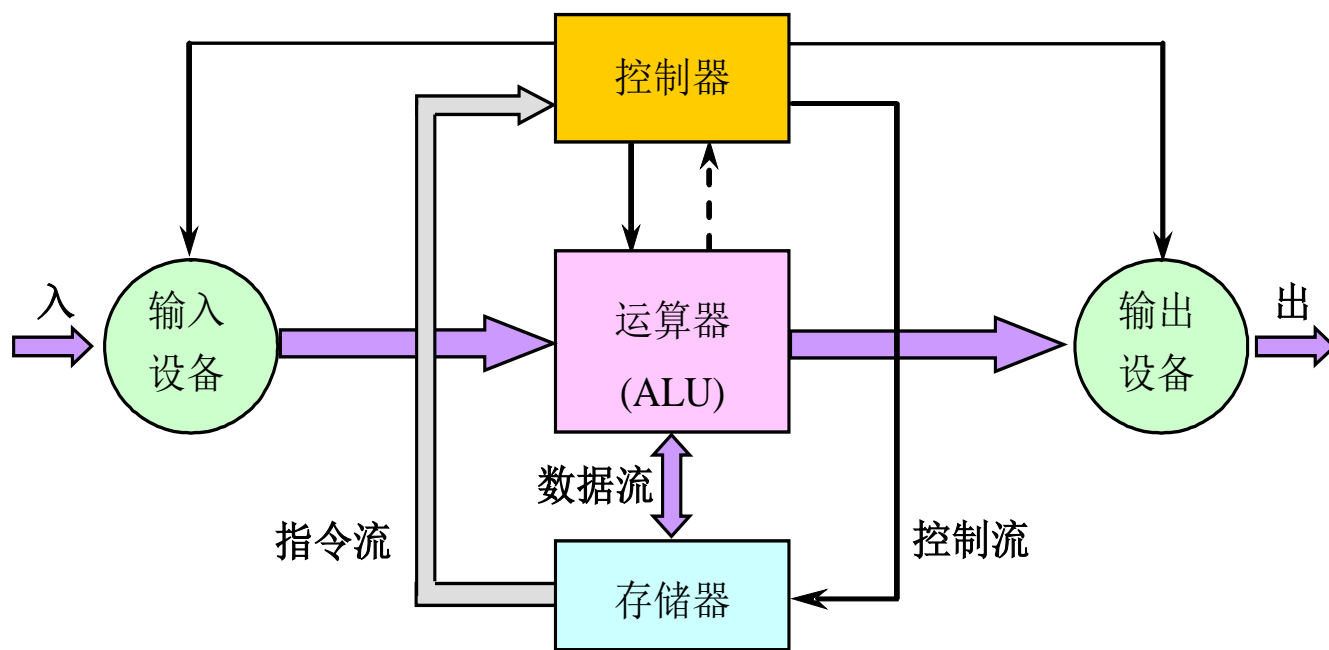
- G_m 表示法有一个很好的特性

几何平均值的比等于比的几何平均值

$$\frac{G_m(x_i)}{G_m(y_i)} = G_m\left(\frac{x_i}{y_i}\right)$$

1.4 计算机系统结构的发展

1.4.1 冯·诺依曼结构及其改进



存储程序计算机的结构

1.4 计算机系统结构的发展

1. 存储程序原理的基本点：指令驱动

程序预先存放在计算机存储器中，机器一旦启动，就能按照程序指定的逻辑顺序执行这些程序，自动完成由程序所描述的处理工作。

2. 冯·诺依曼结构的主要特点

- 计算机以运算器为中心。
- 在存储器中，指令和数据同等对待。

指令和数据一样可以进行运算，即由指令组成的程序是可以修改的。
- 存储器是按地址访问、按顺序线性编址的一维结构，每个单元的位数是固定的。

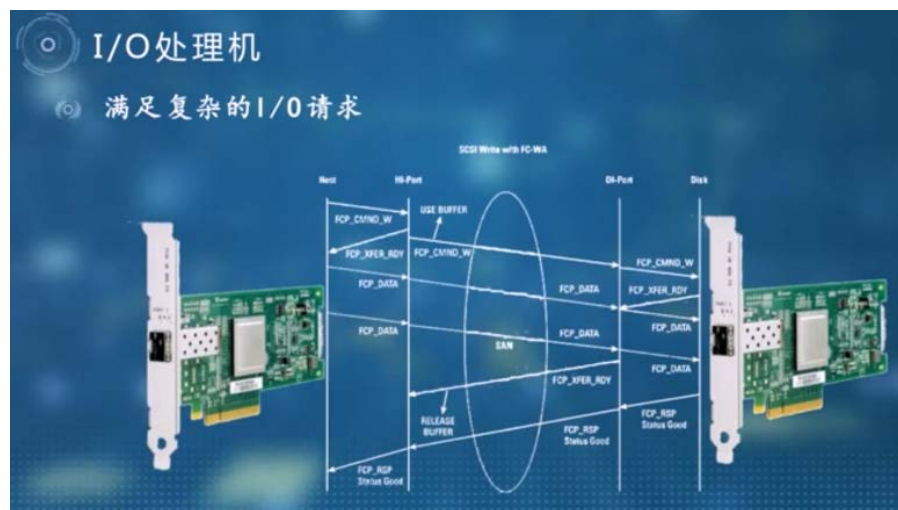
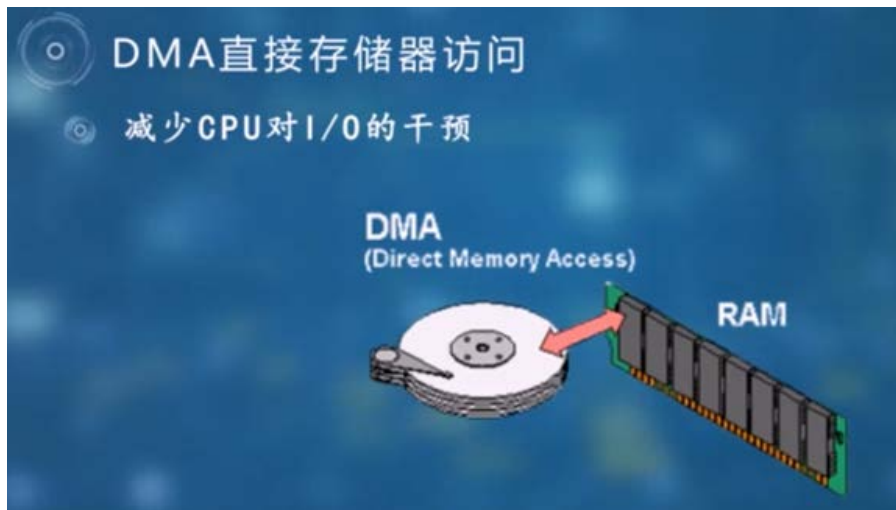
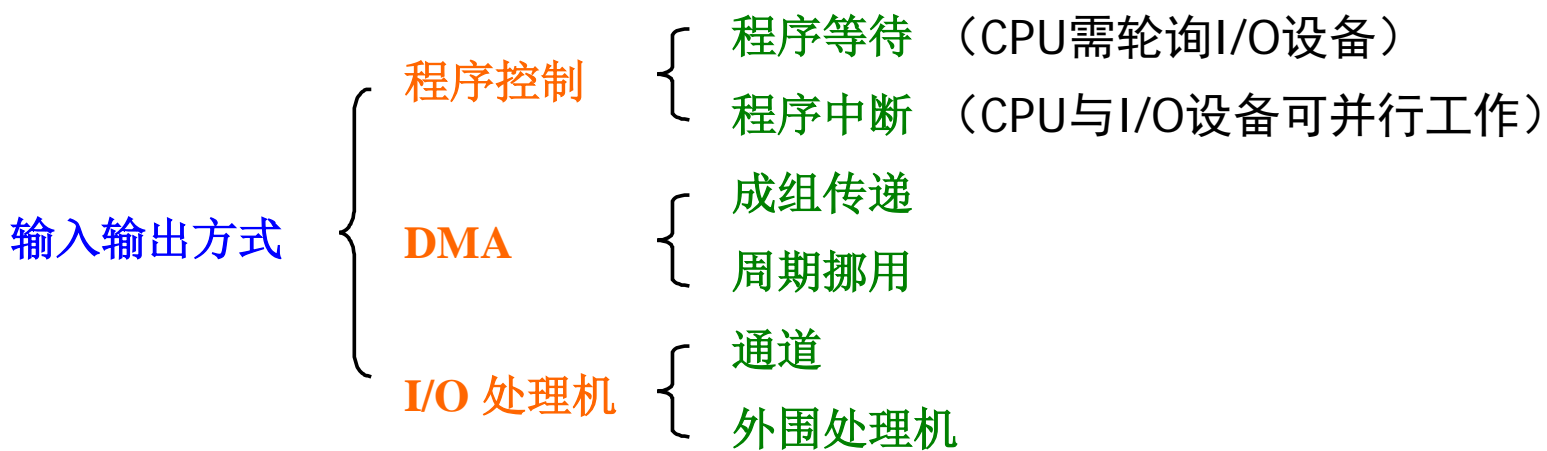
1.4 计算机系统结构的发展

- 指令的执行是顺序的。
 - ❑ 一般是按照指令在存储器中存放的顺序执行。
 - ❑ 程序的分支由转移指令实现。
 - ❑ 由指令计数器PC指明当前正在执行的指令在存储器中的地址。
- 指令由操作码和地址码组成。
- 指令和数据均以二进制编码表示，采用二进制运算。

1.4 计算机系统结构的发展

3. 对系统结构进行的改进

➤ 输入/输出方式的改进



1.4 计算机系统结构的发展

➤ 采用并行处理技术

- 如何挖掘传统机器中的并行性？
- 在不同的级别采用并行技术。

例如：微操作级、指令级、线程级、进程级、任务级等。



1.4 计算机系统结构的发展

- 存储器组织结构的发展
 - ❑ 相联存储器与相联处理机
 - ❑ 通用寄存器组
 - ❑ 高速缓冲存储器Cache

通用寄存器

用于传送和暂存数据，
也可参与算术逻辑运算，并
保存运算结果

高速缓冲存储器 Cache

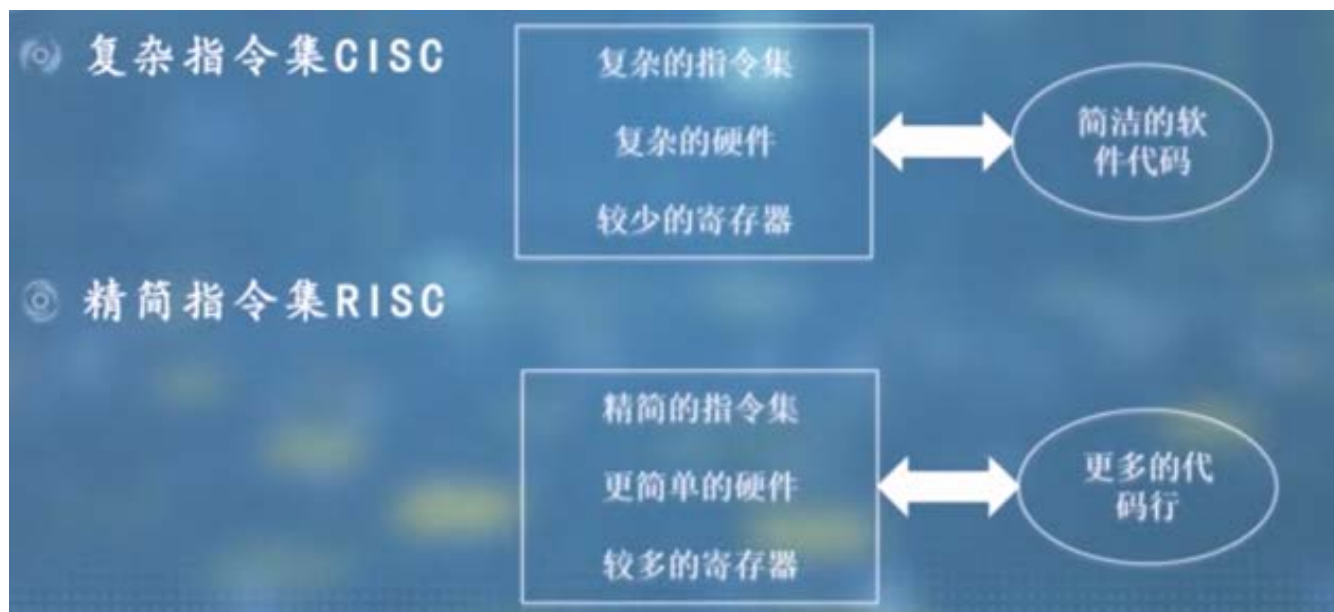
利用程序的局部性原理
将即将访问的数据存放在一个
小容量的高速缓冲存储器中

1.4 计算机系统结构的发展

➤ 指令系统的发展

两个发展方向：

- ❑ 复杂指令集计算机CISC
- ❑ 精简指令集计算机RISC



1.4 计算机系统结构的发展

1.4.2 软件对系统结构的影响

- **软件的可移植性**：一个软件可以不经修改或者只需少量修改就可以由一台机器移植到另一台机器上正确地运行。差别只是执行时间的不同。
我们称这两台机器是**软件兼容**（Software Compatible）的。
- **实现可移植性的常用方法**
采用统一高级语言、系列机、模拟与仿真来实现。

1.4 计算机系统结构的发展

1. 统一高级语言

- 实现软件移植的一种理想的方法，成功范例：Java
- 较难实现：不同计算机同一种语言也不完全相同；不能拿到源代码的问题。

2. 系列机

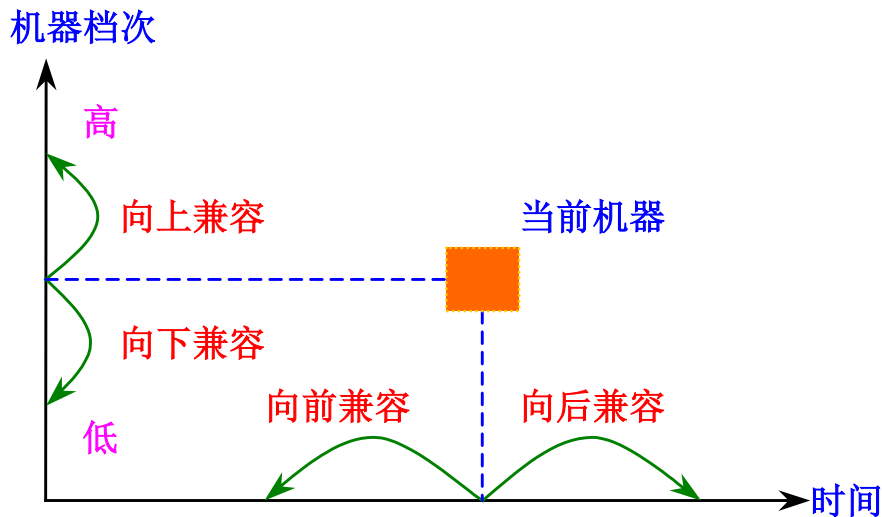
由同一厂家生产的具有相同的系统结构，但具有不同组成和实现的一系列不同型号的机器。

- 较好地解决软件开发要求系统结构相对稳定与器件、硬件技术迅速发展的矛盾。

1.4 计算机系统结构的发展

软件兼容

- **向上（下）兼容：**按某档机器编制的程序，不加修改就能运行于比它高（低）档的机器。
- **向前（后）兼容：**按某个时期投入市场的某种型号机器编制的程序，不加修改地就能运行于在它之前（后）投入市场的机器。
- 向后兼容是系列机的根本特征。

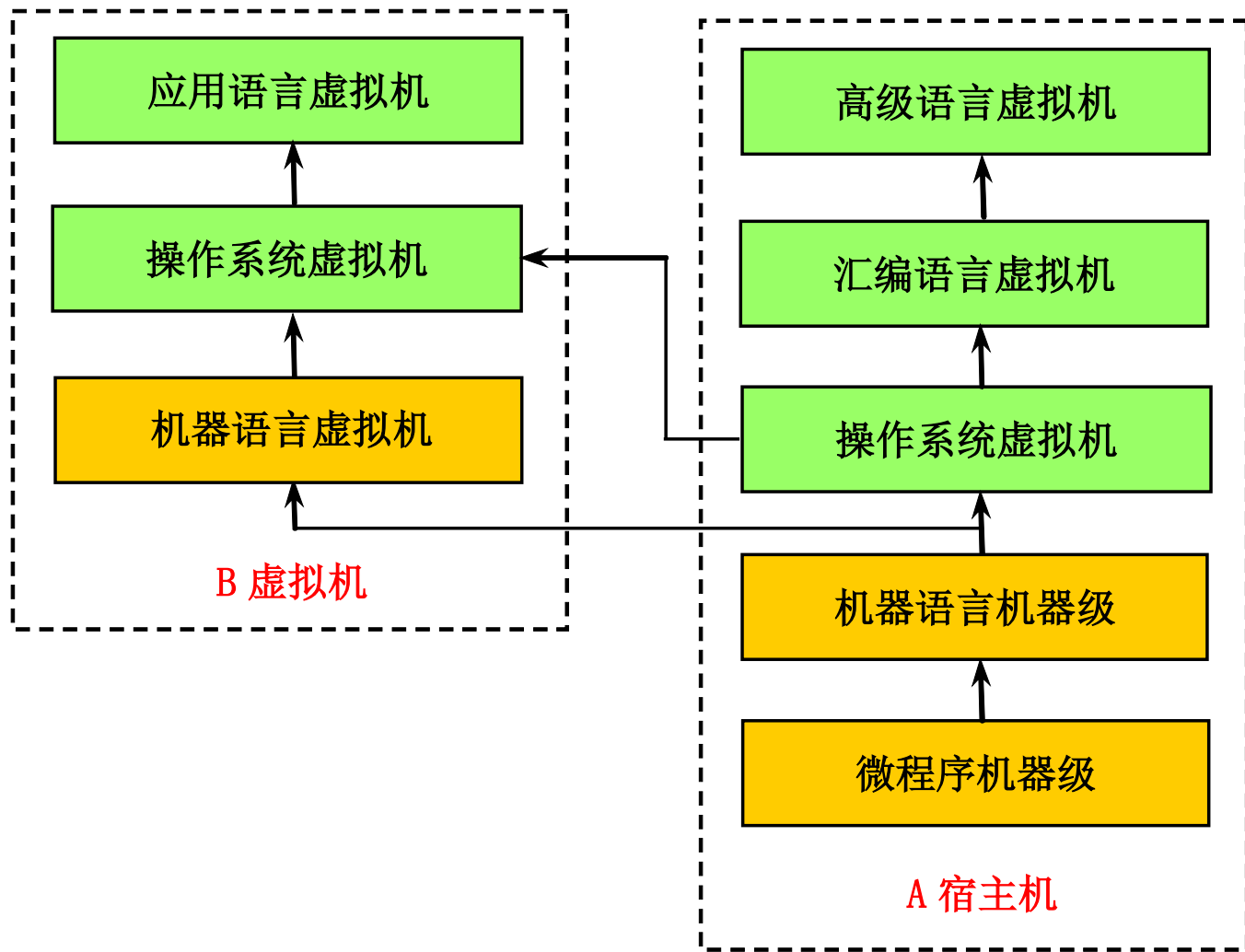


兼容机：由不同公司厂家生产的具有相同系统结构的计算机。

1.4 计算机系统结构的发展

3. 模拟和仿真

- 使软件能在具有不同系统结构的机器之间相互移植。
 - 在一种系统结构上实现另一种系统结构。
 - 从指令集的角度来看，就是要在一种机器上实现另一种机器的指令集。
- **模拟（Simulation）**：用软件的方法在一台现有的机器（称为**宿主机**）上实现另一台机器（称为**虚拟机**）的指令集。
 - 通常用解释的方法来实现。
 - 运行速度较慢，性能较差。

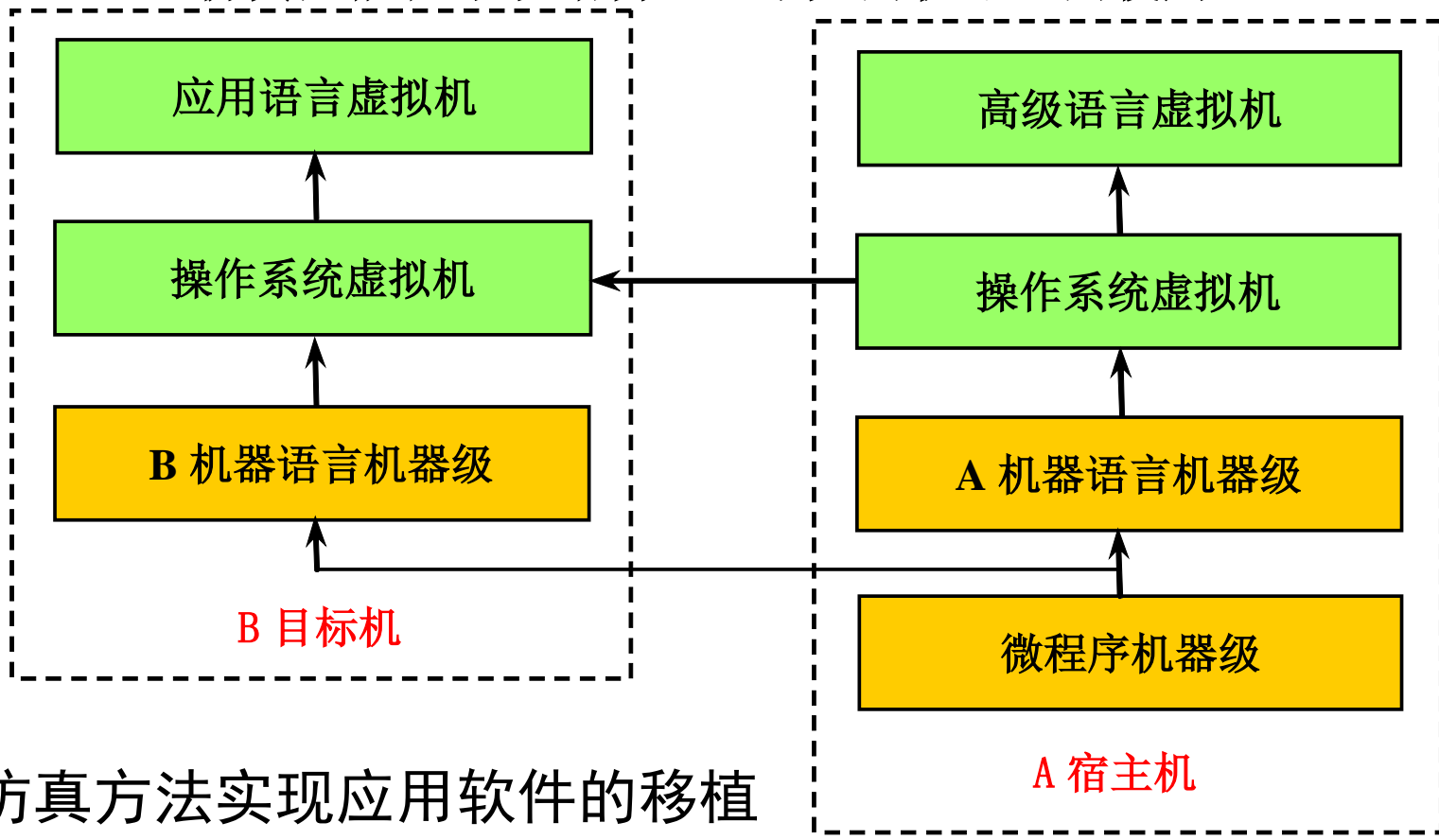


用模拟方法实现应用程序的移植

1.4 计算机系统结构的发展

➤ **仿真**：用一台现有机器（**宿主机**）上的微程序去解释实现另一台机器（**目标机**）的指令集。

- 运行速度比模拟方法的快
- 仿真只能在系统结构差距不大的机器之间使用



用仿真方法实现应用程序的移植

1.4 计算机系统结构的发展

1.4.3 器件发展对系统结构的影响

1. 推动计算机系统结构不断发展的最活跃的因素

2. 摩尔定律

集成电路芯片上所集成的晶体管数目每隔18个月就翻一番。

3. 计算机的分代主要以器件作为划分标准。

➤ 它们在器件、系统结构和软件技术等方面都有各自的特征。

□ **SMP:** 对称式共享存储器多处理机

MPP: 大规模并行处理机 **MP:** 多处理机

5代计算机的典型特征

分代	器件特征	结构特征	软件特征	典型实例
第一代 (1945—1954年)	电子管和继电器	存储程序计算机 程序控制I/O	机器语言 汇编语言	普林斯顿ISA, ENIAC, IBM 701
第二代 (1955—1964年)	晶体管、磁芯 印刷电路	浮点数据表示 寻址技术 中断、I/O处理机	高级语言和编译 批处理监控系统	Univac LAPC, CDC 1604, IBM 7030
第三代 (1965—1974年)	SSI和MSI 多层印刷电路 微程序	流水线、Cache 先行处理 系列机	多道程序 分时操作系统	IBM 360/370, CDC 6600/7600, DEC PDP-8
第四代 (1975—1990年)	LSI和VLSI 半导体存储器	向量处理 分布式存储器	并行与分布处理	Cray-1, IBM 3090, DEC VAX 9000, Convax-1
第五代 (1991年—)	高性能微处理器 高密度电路	超标量、超流水 SMP、MP、MPP 机群	大规模、可扩展 并行与分布处理	SGI Cray T3E, IBM SP2, DEC AlphaServer 8400

1.4 计算机系统结构的发展

1.4.4 应用对系统结构的影响

1. 不同的应用对计算机系统结构的设计提出了不同的要求。
2. 应用需求是促使计算机系统结构发展的最根本的动力。
3. 一些特殊领域：需要高性能的系统结构
 - 高结构化的数值计算
气象模型、流体动力学、有限元分析
 - 非结构化的数值计算
蒙特卡洛模拟、稀疏矩阵
 - 实时多因素问题
语音识别、图象处理、计算机视觉

1.4 计算机系统结构的发展

- 大存储容量和输入输出密集的问题
数据库系统、事务处理系统
- 图形学和设计问题
计算机辅助设计
- 人工智能
面向知识的系统、推理系统等

1.5 计算机系统结构中并行性的发展

1.5.1 并行性的概念

1. 并行性（Parallelism）：计算机系统在同一时刻或者同一时间间隔内进行多种运算或操作。

只要在时间上相互重叠，就存在并行性。

- 同时性（Simultaneity）：

两个或两个以上的事件在同一时刻发生。

- 并发性（Concurrency）：

两个或两个以上的事件在同一时间间隔内发生。

1.5 计算机系统结构中并行性的发展

2. 从处理数据的角度来看，并行性等级从低到高可分为：

- **字串位串**：每次只对一个字的一位进行处理。
最基本的串行处理方式，不存在并行性。
- **字串位并**：同时对一个字的全部位进行处理，不同字之间是串行的。
开始出现并行性。
- **字并位串**：同时对许多字的同一位（称为**位片**）进行处理。
具有较高的并行性。
- **全并行**：同时对许多字的全部位或部分位进行处理。
最高一级的并行。

1.5 计算机系统结构中并行性的发展

3. 从执行程序的角度来看，并行性等级从低到高可分为：

- **指令内部并行**：单条指令中各微操作之间的并行。
- **指令级并行（Instruction Level Parallelism, ILP）**：
并行执行两条或两条以上的指令。
- **线程级并行（Thread Level Parallelism, TLP）**：
并行执行两个或两个以上的线程。
通常是以一个进程内派生的多个线程为调度单位。
- **任务级或过程级并行**：并行执行两个或两个以上的过程或任务（程序段）
以子程序或进程为调度单元。
- **作业或程序级并行**：并行执行两个或两个以上的作业或程序。

1.5 计算机系统结构中并行性的发展

1.5.2 提高并行性的技术途径

三种途径：

1. 时间重叠（Time Interleaving）

引入时间因素，让多个处理过程在时间上相互错开，轮流重叠地使用同一套硬件设备的各个部分，以加快硬件周转而赢得速度。

2. 资源重复（Resource Replication）

引入空间因素，以数量取胜。通过重复设置硬件资源，大幅度地提高计算机系统的性能。

3. 资源共享（Resource Sharing）

这是一种软件方法，它使多个任务按一定时间顺序轮流使用同一套硬件设备。

1.5 计算机系统结构中并行性的发展

1.5.3 单机系统中并行性的发展

1. 在发展高性能单处理机过程中，起主导作用的是时间重叠原理。

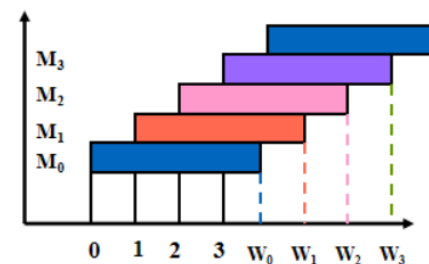
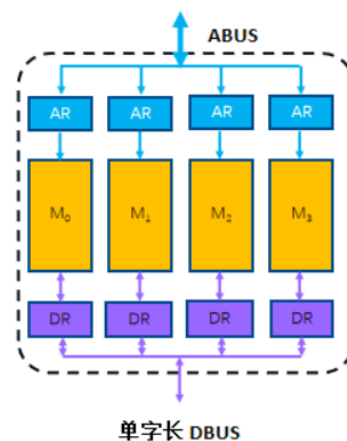
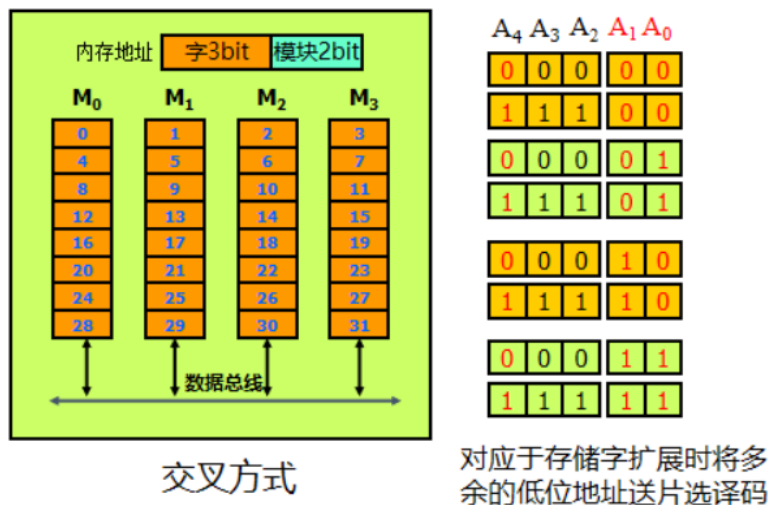
实现时间重叠的基础：部件功能专用化

- 把一件工作按功能分割为若干相互联系的部分；
- 把每一部分指定给专门的部件完成；
- 然后按时间重叠原理把各部分的执行过程在时间上重叠起来，使所有部件依次分工完成一组同样的工作。

2. 在单处理机中，资源重复原理的运用也已经十分普遍。

➤ 多体存储器

CPU以流水方式访问各存储模块的示意图：



连续并行读 m 个字的时间:

$$t_1 = T + (m - 1) \tau$$

顺序读 m 个字的时间:

$$t_1 = Tm$$

低位多体交叉存储器的组织方式

在不提高存储器速率、不扩展数据总线位数的前提下，通过存储芯片的交叉组织，提高CPU单位时间内访问的数据量。

1.5 计算机系统结构中并行性的发展

➤ 多操作部件

- 通用部件被分解成若干个专用部件，如加法部件、乘法部件、除法部件、逻辑运算部件等，而且同一种部件也可以重复设置多个。
- 只要指令所需的操作部件空闲，就可以开始执行这条指令（如果操作数已准备好的话）。
- 这实现了指令级并行。

➤ 阵列处理机（并行处理机）

更进一步，设置许多相同的处理单元，让它们在同一个控制器的指挥下，按照同一条指令的要求，对向量或数组的各元素同时进行同一操作，就形成了阵列处理机。

1.5 计算机系统结构中并行性的发展

3. 在单处理机中，资源共享的概念实质上是用单处理机模拟多处理机的功能，形成所谓虚拟机的概念。

- 分时系统

1.5 计算机系统结构中并行性的发展

1.5.4 多机系统中并行性的发展

1. 多机系统遵循时间重叠、资源重复、资源共享原理，发展为3种不同的多处理机：

同构型多处理机、异构型多处理机、分布式系统

2. 耦合度

反映多机系统中各机器之间物理连接的紧密程度和交互作用能力的强弱。

- **紧密耦合系统（直接耦合系统）**：在这种系统中，计算机之间的物理连接的频带较高，一般是通过总线或高速开关互连，可以共享主存。

1.5 计算机系统结构中并行性的发展

- **松散耦合系统（间接耦合系统）**：一般是通过通道或通信线路实现计算机之间的互连，可以共享外存设备（磁盘、磁带等）。机器之间的相互作用是在文件或数据集一级上进行。

表现为两种形式：

- ❑ 多台计算机和共享的外存设备连接，不同机器之间实现功能上的分工（功能专用化），机器处理的结果以文件或数据集的形式送到共享外存设备，供其它机器继续处理。
- ❑ 计算机网，通过通信线路连接，实现更大范围的资源共享。

1.5 计算机系统结构中并行性的发展

3. 功能专用化（实现时间重叠）

- 专用外围处理机

例如：输入/输出功能的分离

- 专用处理机

如数组运算、高级语言翻译、数据库管理等，分离出来。

- 异构型多处理机系统（Heterogeneous Multiprocessor System）

由多个不同类型、至少担负不同功能的处理机组成，它们按照作业要求的顺序，利用时间重叠原理，依次对它们的多个任务进行加工，各自完成规定的功能动作。

1.5 计算机系统结构中并行性的发展

4. 机间互连

➤ 容错系统

早期不是为了提高速度，而是为了保障系统的可靠性。

➤ 可重构系统（Reconfigurable System）

对计算机之间互连网络的性能提出了更高的要求。高带宽、低延迟、低开销的机间互连网络是高效实现程序或任务一级并行处理的前提条件。平时像普通多处理机一样工作，一旦某处理机发生故障，系统会重新组织，降低档次继续运行。

➤ 同构型多处理机系统（Homogeneous Multiprocessor System）

由多个同类型或至少担负同等功能的处理机组成，它们同时处理同一作业中能并行执行的多个任务。

1.5 计算机系统结构中并行性的发展

1.5.5 并行机的发展变化 4个阶段

1. 并行机的萌芽阶段（1964年～1975年）

- 20世纪60年代初期
 - **CDC6600**：非对称的共享存储结构，中央处理机采用了双**CPU**，并连接了多个外部处理器。
- 60年代后期，**一个重要的突破**
 - 在处理器中使用**流水线**和**重复设置功能单元**，所获得的性能提高是明显的，并比单纯地提高时钟频率更有效。
- 在**1972年**，**Illinois**大学和**Burroughs**公司联合研制**Illiac IV SIMD**计算机（**64**个处理单元构成的）
在**1975年** **Illiac IV**系统（**16**个处理单元构成）

1.5 计算机系统结构中并行性的发展

2. 向量机的发展和鼎盛阶段（1976年～1990年）

- 1976年，Cray公司推出了第一台向量计算机Cray-1
- 在随后的10年中，不断地推出新的向量计算机。
包括：CDC的Cyber205、Fujitsu的VP1000/VP2000、
NEC的SX1/SX2以、我国的YH-1等
- 向量计算机的发展呈两大趋势
 - 提高单处理器的速度
 - 研制多处理器系统

1.5 计算机系统结构中并行性的发展

3. 大规模并行处理机MPP出现和蓬勃发展阶段（1990年～1995年）

➤ 早期的MPP

- ❑ TC2000（1989年）、Touchstone Delta、Intel Paragon（1992年）、KSR1、Cray T3D（1993年）、IBM SP2（1994年）和我国的曙光-1000（1995年）等。（分布存储的MIMD计算机）

➤ MPP的高端机器

- ❑ 1996年，Intel公司的ASCI Red和1997年SGI Cray公司的T3E900（万亿次高性能并行计算机）
- 90年代的中期，在中、低档市场上，SMP以其更优的性能/价格比代替了MPP。

1.5 计算机系统结构中并行性的发展

4. 各种体系结构并存阶段（1995年～2000年）

- 从1995年以后，PVP（并行向量处理机）、MPP、SMP、DSM（分布式共享存储多处理机）、COW等各种体系结构进入并存发展的阶段。
- MPP系统在全世界前500强最快的计算机中的占有量继续稳固上升，其性能也得到了进一步的提高。

如：ASCI Red的理论峰值速度已达到了1Tflop/s

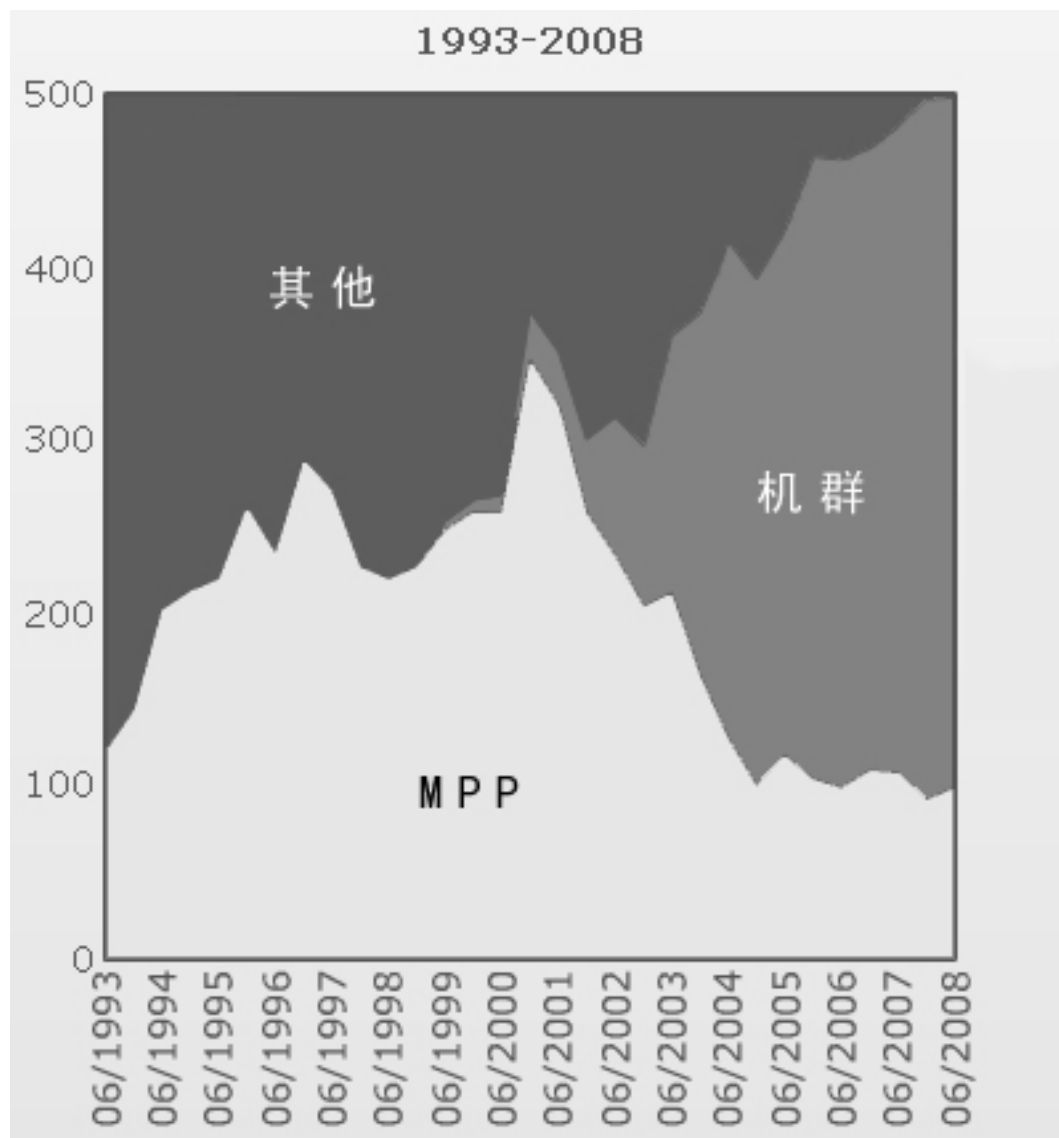
SX4和VPP700等的理论峰值速度也都达到了1Tflop/s

1.5 计算机系统结构中并行性的发展

5. 机群蓬勃发展阶段（2000年以后）

- **机群系统**：将一群工作站或高档微机用某种结构的互连网络互连起来，充分利用其中各计算机的资源，统一调度、协调处理，以达到很高的峰值性能，并实现高效的并行计算。
- 1997年6月才有第一台机群结构的计算机进入Top500排名
- 2003年11月，这一数字已达到208台，机群首次成为Top500排名中比例最高的结构。
- 截至2008年6月，机群已经连续10期位居榜首，其数量已经达到400，占80%。

- 机群已成为当今构建高性能并行计算机系统的最常用的结构。
- 1993年至2008年期间，Top500中机群和MPP的数量分布情况。





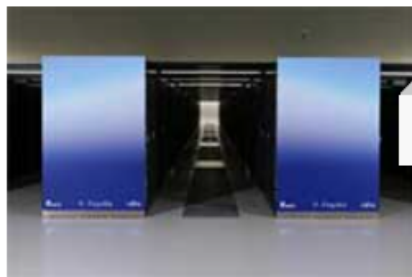
No.1 Frontier (美国)

美国 制造商: HPE 上届排名: 新机

处理器核芯: 1,110,144个; 峰值(Rmax): 1,102.00 PFlop/s

简介:

由AMD公司和超级计算机制造商HPE Cray为橡树岭国家实验室建造, 耗资6亿美元用于核和气候研究等领域进行高级计算。



No.2 Fugaku(富岳)

日本 制造商: 富士通 上届排名: No.1

处理器核芯: 7630848个; 峰值(Rmax): 442.01 PFlop/s

简介:

当前系统处理器与上届相同, 7630848个, Rmax 442010 TFlop/s也未变。Fugaku超算原来被称为“Post K”, 是曾经的世界第一K computer产品的第四代, 采用ARM架构的富士通A64FX处理器。



No.3 LUMI (芬兰)

芬兰 制造商: HPE 上届排名: 新机

处理器核芯: 1,110,144个; 峰值(Rmax): 151.90 PFlop/s

简介:

欧洲开放科学协调组织EOSC-Nordic 为由芬兰CSC托管的LUMI超级计算机, 主要用于气候研究。



No.4 Summit (美国)

美国 制造商: IBM 上届排名: No.2

处理器核芯: 2,414,592个; 峰值(Rmax): 148.60 PFlop/s

简介:

顶点Summit是IBM和美国能源部橡树岭国家实验室 (ORNL) 推出的新超级计算机, Summit 要比神威·太湖之光快 60%, 比同在橡树岭实验室的Titan——前美国超算记录保持者要快接近 8 倍。而在其之下, 近 28,000 块英伟达 Volta GPU 提供了 95% 的算力。



No.5 Sierra (美国)

美国 制造商: IBM 上届排名: No.3

处理器核芯: 1572480个; 峰值(Rmax): 94640 TFlop/s

简介:

Sierra超级计算机美国国家能源局橡树岭国家实验室已经给它定下来要做的事情, 助力科学家在高能物理、材料发现、医疗保健等领域的研究探索。其中在癌症研究方面将用于名为“CANcer分布式学习环境 (CANDLE)”的项目。



No.6 神威 太湖之光 (Sunway TaihuLight) 中国

中国 制造商: 国家并行计算机工程技术研究中心 上届排名: No.4

处理器核芯: 10649600个; 峰值(Rmax): 93015 TFlop/s

简介:

我国的神威“太湖之光”超级计算机曾连续获得top500四届冠军, 该系统全部使用中国自主知识产权的处理器芯片。



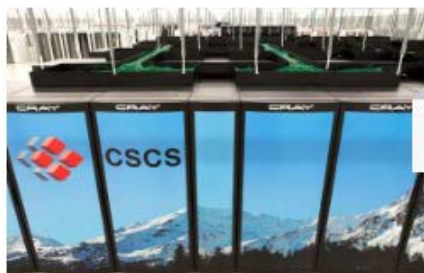
No.7 Perlmutter (美国)

美国 制造商: HPE Cray

处理器核芯: 761,856个; 峰值(Rmax): 70.87 PFlop/s

简介:

Perlmutter 超级计算机在美国国家能源研究科学计算中心 (NERSC) 服务于天体物理学、气候科学等方面, Perlmutter 中含有 6,159 个 NVIDIA A100 Tensor Core GPU, 是世界上最大的 A100 动力系统。



No.8 Selene (美国)

美国 制造商: Nvidia

处理器核芯: 555,520个; 峰值(Rmax): 63.46 PFlop/s

简介:

该系统处理器个数/Rmax 与上届形同。



No.9 TH—2 天河二号 (中国)

中国 制造商: 国防科大

处理器核芯: 4,981,760个; 峰值(Rmax): 61.44 PFlop/s

简介:

天河二号曾经6次蝉联冠军, 采用麒麟操作系统, 目前使用英特尔处理器, 将来计划用国产处理器替换, 不仅应用于助力探月工程、载人航天等政府科研项目, 还在石油勘探、汽车飞机的设计制造、基因测序等民用方面大展身手。

1.1 计算机系统结构的基本概念

系统结构、组成、实现

系统结构分类

1.2 计算机系统的设计

定量原理（经常性事件为重点、Amdahl定律、
CPU性能公式、程序的局部性原理）

1.3 计算机系统的性能评测

执行时间和吞吐率、Benchmark

性能比较：加权执行时间、调和平均值法、
几何平均值法

1.4 计算机系统结构的发展

系统结构的改进：输入输出、存储系统、指令系统
软件的影响（兼容性）、器件的影响、应用的影响

1.5 计算机系统结构中并行性的发展

并行性（从数据角度、从执行程序角度）

- 指令内部并行、指令级并行、线程级并行、任务级或过程级并行、作业或程序级并行
- 提升的三种途径：时间重叠+资源重复+资源共享