



北京希望電腦公司技術叢書



高性能計算 機體系結構

海洋出版社

北京希望电脑公司计算机技术丛书

高性能计算机体系结构

高传善 编译
希 望 审校

海 洋 出 版 社

1992年·北京

内容提要

该书对高性能计算机体系结构进行全面论述,如存储器系统设计,流水线设计技术;数值应用的特性,向量计算机,多处理机及多处理机算法。全书共七章,并有思考习题。

本书适用于微机应用开发人员和科研教学人员,对于应用开发,培训和研究具有较高的参考价值。

需要本书的用户,请直接与北京 8721 信箱联系。邮政编码:100080,电话:2562329

(京)新登字 087 号

责任编辑 阎世尊

高性能计算机体系结构

高传善 编译

希 望 审校

* * * *

海洋出版社出版发行(北京市复兴门外大街1号)

常熟教育印刷二厂印刷

* * * *

开本:787×1092毫米 1/16 印张:18.4 字数:420千字

1992年12月第一版 1992年12月第一次印刷

印数:1—5000册 定价:15.00元

ISBN7-5027-2923-2/TP·143

前 言

1990年由美国IBM华生(Watson)研究中心H. S. 斯通(Harold S. Stone)编写, Addison — Wesley Publishing Company 出版发行的“高性能计算机体系结构(High — Performance Computer Architecture)”一书的第二版是在计算机体系结构方面叙说清晰、内容新的好教材。它具有如下的特点:

- 着重讲解基本的原理和方法,而不是描述一个流行的具体机器的体系结构。
- 反映了在计算机体系结构研究方面的新进展。
- 内容是模块化结构的,各章相对独立。
- 每章后都附有习题,全书后并附有详尽的参改文献。便于读者自我检查及进一步深入阅读。

为学校课堂教学和科学研究的需要,复旦大学高传善、黄德利、顾宝发和华东师范大学葛建峰同志共同翻译了本书。分工如下:高传善翻译序言、第六章和第七章;黄德利翻译第一章、第四章和第五章;顾宝发翻译第二章;葛建峰翻译第三章。现在,我们将其汇总印刷出来。供更多的教学或科研人员使用。相信对于大专院校有关专业的师生(包括本科生和研究生)以及从事有关工作的设计、研究和应用人员,不论作为教材、参改资料或自学课本都有很大的实用价值。限于水平,不当之处欢迎批评指正。

编译者

1992.

目 录

第一章 引言

- 1.1 技术与体系结构..... (1)
- 1.2 它是艺术吗?..... (2)
 - 1.2.1 价格因素..... (3)
 - 1.2.2 硬件考虑..... (5)
- 1.3 高性能技术..... (6)
 - 1.3.1 衡量价格..... (7)
 - 1.3.2 应用的角色..... (7)
 - 1.3.3 VLSI 的影响..... (8)
 - 1.3.4 技术变化对价格的影响..... (9)
 - 1.3.5 算法和体系结构..... (11)
- 1.4 历史的参考..... (12)

第二章 存贮器系统设计

- 2.1 发掘程序的特性..... (15)
- 2.2 高速缓冲存贮器..... (19)
 - 2.2.1 基本的高速缓存结构..... (19)
 - 2.2.2 高速缓存设计..... (20)
 - 2.2.3 高速缓存分析:迹生成和迹长度..... (25)
 - 2.2.4 高速缓存分析:迹捷联(Trace Stripping)..... (32)
 - 2.2.5 替换策略..... (37)
 - 2.2.6 高速缓存中的复盖区(Footprints)..... (41)
 - 2.2.7 写入高速缓存..... (47)
 - 2.2.8 其他性能尺度..... (49)
 - 2.2.9 高速缓存性能模型..... (50)
- 2.3 虚拟存贮器..... (56)
 - 2.3.1 虚存结构..... (57)
 - 2.3.2 虚存映象..... (59)
 - 2.3.3 改进程序的局部性..... (63)
 - 2.3.4 替换算法..... (65)
 - 2.3.5 虚存系统中的缓冲效应..... (67)
- 习题..... (70)

第三章 流水线设计技术

- 3.1 流水线设计原理..... (77)
- 3.2 流水线计算机中的存贮器结构..... (86)
- 3.3 流水线计算机的性能..... (88)
- 3.4 流水线级的控制..... (95)
 - 3.4.1 多功能流水线的设计..... (95)

3.4.2 碰撞向量和流水线控制	(99)
3.4.3 最佳性能流水线	(103)
3.4.4 利用延迟提高性能	(105)
3.4.5 消除互锁	(111)
3.5 开发流水线技术	(112)
3.5.1 条件转移	(112)
3.5.2 指令的内部超前和延迟	(116)
3.5.3 既有高速缓存又有虚存的机器	(123)
3.5.4 RISC 体系结构	(124)
3.6 历史的参考	(131)
习题	(136)

第四章 数值应用的特性

4.1 大型数值问题的分类	(137)
4.1.1 连续模型	(138)
4.1.2 粒子模型	(139)
4.2 高性能机器设计的限制	(140)
4.3 连续模型的体系结构	(145)
4.4 连续模型的算法	(145)
4.4.1 Cosmic Cube 与 ILLIAC IV	(146)
4.4.2 数据流的要求	(150)
4.4.3 并行解法	(153)
4.4.4 递归倍增和循环归约	(155)
4.5 完全混洗	(155)
4.5.1 完全混洗互连模式	(159)
4.5.2 完全混洗的应用	(165)
4.6 连续模型的体系结构—什么方向?	(167)
习题	(169)

第五章 向量计算机

5.1 普通的向量计算机	(169)
5.1.1 多存贮器模块	(171)
5.1.2 中间存贮器	(176)
5.2 数值算法的访问模式	(179)
5.2.1 高斯消去法	(180)
5.3 向量机的数据结构技术	(182)
5.4 附属向量处理机	(187)
5.5 稀疏矩阵技术	(191)
5.6 GF-11—一个超高速向量处理机	(192)
5.7 关于向量计算机的最后评注	(194)
习题	(196)

第六章 多处理机

6.1 背景	(199)
6.2 多处理机性能	(202)
6.2.1 基本模型——不重叠通信的双处理机	(203)
6.2.2 向 N 处理机扩展	(204)
6.2.3 统计模型	(207)
6.2.4 带有线性通信价格的模型	(207)
6.2.5 一个乐观的模型——完全重叠的通信	(208)
6.2.6 带有多通信链路的模型	(210)
6.2.7 多处理机模型	(211)
6.3 多处理机互连	(213)
6.3.1 总线互连	(213)
6.3.2 环互连	(216)
6.3.3 交叉开关互连	(218)
6.3.4 两维和三维网格	(221)
6.3.5 混洗交换互连和组合开关	(222)
6.3.6 蝶形运算和逆二进变换	(223)
6.3.7 组合网格与取并加(Fetch-and-Add)	(226)
6.3.8 超立方体互连	(231)
6.4 多处理机中的高速缓存相干	(231)
6.5 小结	(235)
习题	(235)

第七章 多处理机算法

7.1 简单的并行性	(237)
7.1.1 do par 和 do seq 结构	(238)
7.1.2 阻挡(Barrier)同步	(239)
7.1.3 性能考虑	(241)
7.1.4 增加颗粒度	(243)
7.1.5 启动任务	(245)
7.2 同步技术	(247)
7.2.1 用测试并设置(Test-and-Set)同步	(248)
7.2.2 用增量和减量(Increment and Decrement)同步	(250)
7.2.3 用比较并交换(Compare-and-Swap)同步	(252)
7.2.4 用取并加(Fetch-and-Add)同步	(258)
7.2.5 其它支持并行同步的体系结构	(260)
7.2.6 高速缓存相干与同步	(265)
7.3 并行搜索——如何采用和不用并行性	(267)
7.3.1 单峰函数最大值的搜索	(267)
7.3.2 并行分支与定界——售货员巡走问题	(270)

7.3.3 加速和并行的复杂性	(274)
7.4 将串行算法变换为并行算法	(275)
7.4.1 相关性分析	(276)
7.4.2 发掘迭代中的并行性	(277)
7.4.3 调度对并行性的影响	(280)
7.5 关于多处理机的最后评注	(281)
习题	(282)
参考文献	(287)

第一章 引言

本书研究的重点是高速计算机系统体系结构的设计和分析。我们把一个计算机系统看作是由各种功能模块,如处理机、存贮器、输入输出通道及开关网络等构成的系统。我们新讲的体系结构指的是由这些功能模块组织成一个计算机系统的构造。计算机系统的体系结构设计,包括选择处理机、存贮器等各种功能模块,以及通过设计互连网络把它们组织在一起构成一个系统。这同一幢大楼的建筑设计相似,那里涉及到选择各种材料、安装配备各种构件,并把它们组织在一起,构成一个可行的建筑结构。

1.1 技术和体系结构

计算机体系结构是受技术推动而发展的,每年都有许多新的器件,新的功能和新的可能性产生。对于今于还是富有想象力和有效的体系结构,到了明于亦许成为陈旧过时之物。相应地,今天看来还是稀奇古怪的想法也许明天一跃而成为新思想。没有一个绝对的准则说那一个体系结构一定优于另一个。

学习体系结构的关键是学会如何综合可用的技术来评价体系结构。如同了解计算机系统的最初始计算速度一样重要的是知道计算机系统是否有效地使用了处理机的周期、存贮空间和输入输出带宽。在评价体系结构时,注视的对象是价格和性能两方面,而不仅仅是性能。由于技术的改进,各种功能模块的相对价格和它的绝对价格每过几年都会有一个大的变化,使得不同类型模块的有效价格设计的比例也随着技术而改变。

这本书采用这样的方法,即需要讲的是方法学而不是结论。我们提出一份文件性的设计菜单,其中有些对今天是合理的,而另外一些则不一定适当。我们要说明如何按照设计菜单做出选择来构造高性能的系统,并且按照 90 年代现有的技术水平评价所产生的这些系统。通过这些评价得出的结论到 90 年代中期大概仍是合理的和适用的。但是决不能认为今天看来是最强的体系结构会永远都是最好的。

然而方法学是无时间限制的。有时计算机设计者需要构造一个新的设计选择菜单,以此菜单和这本书所描述的设计和评价技术,设计者在任何一个十年中,对于当时的技术,应该能够造出高质量的系统。

性能分析应以整个系统的体系结构为基础。设计和分析高性能系统是非常复杂的。然而,最好的方法是把一个大系统划分为许多功能模块,组成层次结构,每一个模块在体系结构上可以独立开来分析。如果单个功能模块仍然十分复杂,可把它再进一步细分为由许多原语功能组成的集合来分析。例如处理机体系结构包括了组合在一起的存贮器、算术运算单元以及生成运算处理的控制逻辑等。

处理机体系结构的一个重要方面是设计处理机的指令系统,在本书的阐述过程中我们将了解到当今关于指令系统是否应该简单或是复杂的狂热争论。这里我们并不想解决这个争论,答案不可能只是一个。但是我们要说明,确定这些答案的因素,不管用什么技术,在一个新

的设计过程中设计者是可以测量出来的。

计算机体系结构有时会同计算机硬件的设计混淆不清，因为计算机体系结构论述涉及的是功能级模块，与硬件级并不互斥，它比硬件包括的范围更广。例如，当我们说处理机执行算术与逻辑功能时，我们可以充分肯定这些功能是建立在硬件上的，并不需要附加程序设计。如果我们说的是处理机的存贮器管理功能，那末这些功能的实施则可能是某些软件和硬件的结合，这种软硬件的结合正确与否将依赖于系统的性能，现有软硬件成分的可用性以及价格。

当超大规模集成电路（VLSI）还处在初级阶段时，存贮管理功能是由软件实现的，处理机体系结构对这种软件的支持仅是提供寄存器组用于地址映照和保护。有了 VLSI 则可以嵌入许多硬件存贮管理功能。许多系统为了执行曾经单独由软件实现的存贮管理功能，采用了复杂的硬件算法。

软件和硬件之间的界限随着近年来软件直接嵌入到存贮管理芯片的只读存贮器中而变得有些模糊起来。这种芯片经封装仅是一个执行存贮管理的黑箱，它使问题的解决变成用硬件。设计者应用这些片子，对存贮管理不需要提供另外的软件。如果芯片执行了大多数存贮管理内部的但不是全部的功能，设计者则必须通过观察分析，以便把没有提供给芯片的那些特征合并到软件模块。

回顾一下，计算机体系结构是一个以多个成分做成的系统，这些成分可以是硬件、软件或软硬件的结合。包含在体系结构中的技巧就是选择一个好的成分集合，并把它们组合在一起，使他们成为同一个系统那样有效地工作。后面几章将给出体系结构的各种例子，其中有些表明是成功的，有些也许是一些可能成功的建议。

1.2 它是艺术吗？

在 1985 年 1 月的纽约时报上有一篇文章叙述了（Dekooning）一幅没有署名的画的发现，此事令一些艺术评论家瞠目结舌，尽管画上没有 Dekooning 的签名，但毫不怀疑此画乃是他的作品。而画是挂在公众参观的陈列室中，作品画的是夏季海滩别墅外面的一条长凳 Dekooning 用来抽象地给出大理石的外观。这幅画是知名大师艺术上的一件伟大作品呢？抑或仅是画一个私人的座位？艺术欣赏的重要一点是美学，对美学我们没有一个绝对的量度，也无绝对的测试方法来断定一幅画是名作抑或还是鹰品，如果艺术界赞同这是一件杰作，那末它就该是名作。

计算机体系结构也有美学的一面，但它和艺术的美学完全不同。我们评价体系结构的质量可以按照每个周期结果的最大数目、程序和数据的容量、价格以及从各方面看都是重要的其它度量来评定，我们并不需要争论像“但它是快的吗？”这样的问题。

当必须做出选择时，体系结构可以根据关键的度量值进行比较。挑战的到来，是因为技术每年都给予我们新的选择，去年的决策也许保持不到今年。设计者不仅必须理解对于今天是最好的决策，而且也要了解在整个设计生命期中影响技术所期望变化的因素。因此不仅评价技术在各个决策中起着关键的作用。而且通过数年时期使用这种技术，设计者获得了理解影响新体系结构技术发展的经验和能够判断出将来若干年的趋向。

以下是判断体系结构的主要标准

- 性能；
- 价格；

• 程序和数据的最大长度；

有成千和更多的其它判定标准，如重量、功耗、体积、编程的方便性等。在个别情形下，这些判定标准相对说也许有很重要的意义。但是，上面列出的 3 项标准在所有应用中是最重要的，而且也是他们大多数标准中最关键的标准。

1.2.1 价格因素

价格标准值得多做一点解释，这是因为有许多人对它是什么意义认识上还很混乱。对用户讲，计算机系统的价格是用户为系统所付出的钱，亦即价钱。对设计者，价格的定义并非如此简单清楚。在大多数情况下，价格是指制造的价格，包括了合理偿还为建造系统用去的开发和资本工具的价格。我们经常看到对体系结构所作的一些比较，即把系统 A 的部分价格同系统 B 的购买价格作比较，其中系统 A 是作为系统改进提出的一种新型体系结构，而系统 B 代表一种商业方面的模型。

另外还常常关于硬件和软件的虚妄比较。在计算机的早期年代，软件常常是同硬件结合在一起不收费的，但是随着计算机工业的逐渐成熟，软件本身已变成可以出售的有价值的商品。

如今我们发现曾经是免费的软件，现在控制着计算预算的重要部分，人们把其发展趋势

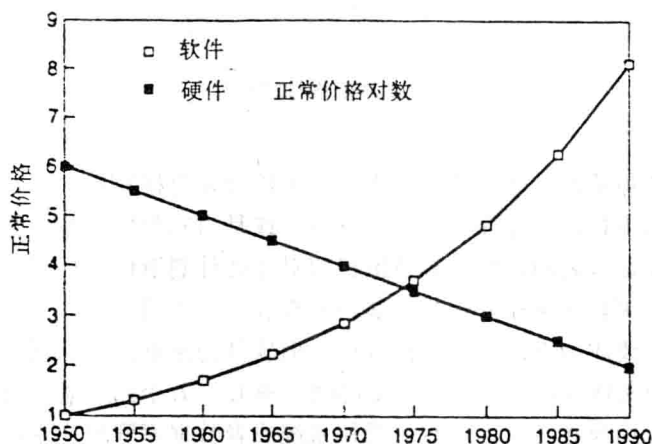


图 1.1 计算机价格趋势的直观图

描绘成图 1.1，其中我们看到软件的价格随其复杂性和软件的膨胀呈稳定的上升趋势，而软件工具的进步并没有明显地减缓这种趋势。图中绘出的另一根曲线是同一时间周期里硬件的总趋势，硬件成分以难以置信的速率减低其价格。如果我们设想这种势头一直持续二十年，也许我们会相信硬件大概将附在软件一起，购买软件时，硬件可免费提供。但这种观点看来未免是肤浅的。

软件和硬件价格的每一个都由两部分组成：

1. 一次开发价格；
2. 每个产品的制造价格。

软件或硬件的每个产品的实际价格由图 1.2 表示，它是该产品生产数量的函数。注意到第一个产品的价格等于开发价格。价格曲线随着产品数量的增加移动上升，但其斜率在产量

很大时趋向减缓。生产经验表明这是因为大批量生产时，每件产品的价格趋向减少。图 1.2 中的曲线表示一种产品产量总数的累积价格。产品的售价是曲线上所表示的价格除以产量，并加上毛利。于是，当开发价格高时，售价与产品数量的关系就显得很敏感。

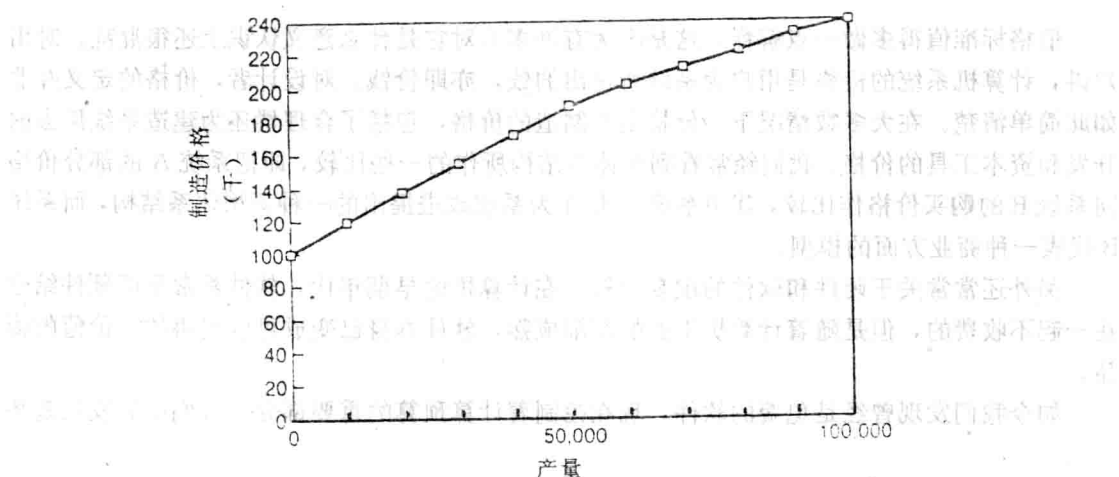


图 1.2 累积生产价格与产量

当软件基本上还是免费时，开发价格或者是同硬件开发价格算在一起，由开发他们自己的大多数软件的用户们承担；或是简单地不予计算；或是由软件生产者去弥补。因为硬件制造价格对比起来现今还是很高的，软件制造价格相对于硬件制造价格仍很小，只要开发价格没有被通过直接的软件销售所覆盖，还是有理由放弃软件的价格。

软件开发价格最终要成为有效和不再被忽视，但软件的复制基本上仍是免费的。我们可以容易地同大量生产的某些 VLSI 芯片作比较（例如，有几十万个晶体管的复杂的微处理机芯片）。芯片的开发价格大约与在芯片上运行的操作系统或者数据库管理应用软件包的开发价格是同一个数量级。

在数量很大时，每块芯片的制造价格与软件的制作价格是同样一个数量级。然而，我们看到出售芯片的售价可以小到软件售价的十分之一，而含有这些芯片的计算机系统大概可卖到 10 倍的软件价格。芯片、软件和计算机系统的售价似乎同制造价格无关，这部分的售价是由销售的数量决定的，因为出售价格必须弥补开发价格。复制出售的芯片可以是几百万，但是数据库管理软件的复制仅有几十万可以出售。仅此就说明了在售价上它们十分不同的原因。

我们的分析也说明了为什么若干年内，硬件价格相比于软件价格仍是占主要的。这里的问题是制造价格，今天软件的制作价格近乎等于零，不可能趋向更低，这使软件在市场竞争中的定价只能反映到开发价格的补偿中。

当芯片的基片很小时，硬件制造价格是软件制作价格的许多倍。今于进行准确的软件复制比之硬件复制远没有那末昂贵。硬件复制要求安装测试，以得证每个复制器忠实于原来的设计，它远比软件制作线上的质量保证复杂。而软件制作只要简单地比较软件的每个信息，看

它是否同原来的程序一致。

我们看到硬件的定价负担了每个产品同开发价格一起的制造价格，相反软件定价很大程度上反映开发价格。当计算机装在单个芯片上时，它们的售价同软件售价应有某些相似性。实际上我们看到手动计算器大致上以大多数流行软件工具的同样价格在出售。但是包含有成万成千个部件的计算机比任何软件包的复制更复杂，至少，硬件制造厂家必须测试芯片及系统，以便排除故障。而在软件制作中则省略了相应的过程。因为软件复制是低价格、可靠和验证廉价的，在市场竞争中，中等或高性能计算机给伴有软件的买主作为赠品是根本不可能的。

1.2.2 硬件考虑

关于将来新式设计的其它虚妄的争论涉及到在一个系统中大量硬件成分的使用问题。设计者们令人信服地申述，按照现在的发展趋势，硬件的价格将是无足轻重的，使得我们将来能建造出比之今天的硬件复杂性更大得多的系统。很清楚，在这一争论中有一点是真实的，即以现有系统相同的价格，将来的系统无疑其功能更强和更复杂。但是一定要慎重对待这一争论，因为十足的硬件浪费是谁也不想要的。

将来某系统 A，逻辑上是现行系统的 100 倍，而系统 B 的性能基本上同 A 的一致，但逻辑上仅有现行系统的 10 或 20 倍，那么系统 A 将处于十分不利的竞争地位。成百成千地复制系统 A 出售，系统 A 可以同系统 B 竞争定价。然而在很高的产量下，系统 A 体系结构的低效能迫使它的售价在相同系统价值下比系统 B 的高。当然这里假定系统 A 和系统 B 是由相同技术产生的那些部件建成的，如果 A 芯片的密度是 B 芯片的 10 倍，而每个器件却小于 10 倍的代价，那末情况就改变了，是器件技术而非体系结构成为系统价格的决定因素。

探讨将来通过依靠低价格的部件来改进体系结构贯穿着本书，但我们始终注意我们所肯定在很好使用着构造模块的哪些体系结构的效率。

例如，考虑没有共享存贮器的多处理机系统，并假定我们要运行一个其中每个处理机都同样执行的并程序。很显然，我们可以在每个处理机中装入同一个程序副本，当程序很小或者处理机的数目不过份多时，多副本的存贮器开销也许还是允许的。

但是，倘若程序是兆字节大小，并且如果我们计划在系统中采用 1000 个处理机，那又会怎样？那末多的程序副本占有几千兆字节的外存贮器，若有某种方式让所有的处理机共享一个代码副本，那末存贮器就不需要全部兑现。

如果系统 A 采用多个程序副本，而系统 B 通过巧妙的设计以单个副本获得近乎相同的性能，那末系统 A 所需的额外几千兆字节存贮器，可能使系统 A 整个地无法同系统 B 竞争，除非存贮器的价格低到使得系统几千兆字节的存贮器只占系统价格的微不足道部分。系统 A 的设计者希望存贮器每个位的价格今后将下降，但系统 A 需要 10^{10} 以上的位，而这是一个庞大的乘数，即使目前这种趋势继续下去，要补偿相抵这差额，每位价格的跌落要 20 到 30 年。

在刚才举的例子中，系统 A 的设计者必须意识到对于具体应用要寻找能够克服这一基本缺点的其它途径。在每个处理机需要不同程序的各种应用中，系统 A 也许是统统有效的。但在这当中，系统 B 有一个极大的难以制胜的优势，设计者应该通过刻划体系结构如何被应用的若干实例来衡量体系结构的质量。有效性随各种应用变化很大，这种衡量应该揭示出哪些方面体系结构真正对用户有益，而哪些方面采用其它途径更好。

一个计算机体系结构也许会有一些细小而代价高的固有缺陷逃过它的设计者的检查。能够构造出基本上一样且纠正了哪些缺陷的体系结构的不同设计者，他一定可以生产出更有效，

因此也更有竞争力的机器。设计者们不能用辩解硬件不花代价来隐瞒无效率。

作为这一准则的简单例子,考虑一个有 1600 个这样较大数量处理机的体系结构,并假定这些处理机的应用因 N 个处理机的加速与 $\log_2 N$ 成正比。(这听起来惊人,居然会作出这样的建议) 1600 个处理机对一些常数 X 仅产生 $14X$ 的加速。尽管设计者可以辩解 1600 个处理机是如此廉价以致于我们可以忽略它们的价格,但重要的事实是这种应用中系统运行比它在单个处理机上运行仅快 $14X$ 倍,加速增加的价值相对这些处理机的额外价格却是很小的。

在这个竞争的世界里,十足的无效率的体系结构不可能长久地逃避人们的注意。因此很快就出现系统 B 同系统 A 竞争。这里除了规模较小外,系统 B 的体系结构同系统 A 的一致,事实上如系统 B 仅 128 个处理机,而不是 1600 个,所以系统运行仅比单个处理机快 $7X$ 倍。

系统 A 的复杂性要超过系统 B 的 100 倍,然而系统 A 运行仅快 2 倍。硬件价格系统 B 几乎为 0,无法同系统 A 相比。至少后十年,关于这种重复 100 倍的硬件对 2 倍性能的价格基础不可能说是合理的。

这一节的争论告诉我们:

- 我们可以通过价格和性能评价体系结构;
- 体系结构的有效性必须根据体系结构所在应用的负荷来衡量;
- 由于浪费资源而低效的体系结构难于同比较简单但却更有效的体系结构竞争。

如果计算机体系结构纯粹是一门艺术和单是确定体系结构设计质量的美学,那我们就没有技术先进性的根据。计算机体系结构恰恰是把设计艺术同来自精心分析的远见卓识相结合,以制造出新型的计算机系统,为用户们提供更大和更好的服务。

1.3 高性能技术

上一节讨论了评价标准,这一节着重于高性能技术。我们的目标是描述多种不同的改进系统性能和为了评价哪些技术给出某些附加信息的方法。当今可用的评价技术还是比较广泛的。每产生一个新技术,也就涌现出新的思想。

这本书包括了现有各种设计选择菜单的主要部分,但这些菜单到出版时并不完整、因此我们要探讨设计方法学,即确定标准的设计问题,产生这些问题的解、评价和选择最佳或最合理的解。

虽然我们强调性能,但为了比较体系结构,一个全面的评价应该考虑所有的标准。只是在性能上我们赋予更大的份量。对大多数设计、空间、价格和性能是作为一个参数处理,即性能价格比。作这种比是合适的,因为当你由于相同因素增加性能和价格时,这种比保持不变。

我们相信多数用户是愿意为一台性能价格比同现有的机器一样,而速度要快 10% 的机器多付 10% 的费用。如果一台机器对提高 10% 的价格产生高于 20% 的性能,在更换为这种新机器时,用户们就会看到真正的利益。实际上,机器的性能价格比愈低反映出每次计算的价格也愈低。在多数情况中,用户对提高 10% 的价格仅产生高于 5% 的性能的机器将不会感兴趣,因为当他们更换到新机器上时,每次计算的价格是上升而不是下降。

例外的情形发生在现有设备饱和而用户一定要有更大的容量和能力的机器时,此时性能价格比不可能告知你真相,因为用户的更大能力的总利益也许比获得此能力的代价更大。用户为获得此能力,每次计算实际上付出更高价格的事实是同能够进行以前不能做的计算价值

一致的。然而，如果用户有一个如何获得其必要容量的选择，用户也许仍然会选取基于低价性能比的解决办法，虽然所有这些可能的解决办法要比用户现行系统的比率更高。

1.3.1 衡量价格

我们注意过给出基于价格和性能两方面变化的比较小的一些例子。对于小的变化性能价格比是相对质量的一个好的指示器，但是当价格和性能变化因素差别很大时，它的可用性就破坏了。

例如：衡量 8 位电视游戏系统这种小计算机的性能价格比，如把它同功能强得多的计算机辅助设计工作站系统比较就很靠不住。虽然两个系统都用作图像显示和实时交互操作，但假定我们可以找到某种衡量相对性能的方法，大概电视游戏机较之工作站有更好得多的性能价格比。问题是这里的系统相对价格是按 1000 比 1 的因素变化，类似地相对性能因素变化也很大，虽然大概没有相对价格变化那末大。

电视游戏机不可能做工作站那样的工作。而且即使你把足够多的电视游戏机集中放在一起，它们具有工作站相同的性能，价格比工作站的低，但这种电视游戏机的集合仍然不可能做工作站同样的工作。正是这样，要确定基于性能价格比的系统间的比较是否有效，一点应该注意的是做功能相似和性能相对接近的两个计算机之间的比较。

这个讨论指出了使体系结构保持先进的两种重要的方法：

1. 在价格和性能方面扰动要小，以产生低的价格性能比；
2. 提高绝对性能，使得设计在合理的价格上可行。

对“小”的变化，我们粗略地意味为因素 10 或更小。比这大的变化肯定是欢迎的，但性能价格比不可能像衡量评价变化那样可信。由上述的第 2 点，假如附加的价可以为用户接受，性能价格比实际上可以增加，因为更大能力的利益超出了取得此能力的价格。在整个这部分中，我们使用这两个标准作为评价思想的非正规方法。

因为在商品流通中绝对价格的衡量每年都在变化，就影响价格的其它参数讲，它们对确定价格更有用。这些参数包括物理参数，如芯片针数、芯片面积、数量、板面积和实现体系结构的电源功耗等。参数也包括了与开发有关的各种因素，如设计周期时间，编写有关软件的数量以及所需开发队伍的规模。

这一节不可能很容易地说出所有影响价格的因素，但可以分出它们中最重要的部分。特别是在比较两个密切相关的体系结构时，它们的差别仅限于少数几个关键的设计选择。我们的意图是集中在差别上讨论它们影响价格因素的那些方法。每一个不同的途径都有它自己的长处和短处，它们依次影响着该途径的价格。我们不可能给出一个绝对的价格，但我们可以说明在价格参数上那些是影响设计的决定因素，读者可以应用流行的价格功能完成评价。

1.3.2 应用的角色

随着技术进步的急剧的变化，我们如何看待高性能体系结构的设计问题呢？例如，新的技术使大规模的并行性成为可行。在我们通过重复处理机找到更高级的性能之前要投入多少的努力来增加单处理机的性能？对这此些问题设有简单的答案。我们需要结合各种解答，而且我们可以选择的几乎肯定将是依赖于应用。

应用的作用是高性能竞争中的关键，因为要向体系结构索取最大可能的吞吐量，其价格将趋向很高。在这当中无效率尤其代价高，因为当应用的作用比之应有的性能低时，无效率

更大地增加了已经高的价格。如果应用领域是着重于某些好确定的负荷，设计该负荷类型的体系结构则成为可能。结果是体系结构可去掉那些不相关的功能，而这一点在通用机中则是必要的。这样它可以主要由跟具体负荷有关的各种功能来构成。

那末，设计目标就是要通过确定体系结构的所有功能部件有效地获得高性能来减少无效率。假如能够建造一个通用机器，它对所有高性能应用都同样有效，那末工业界会接受推行。在未来的几年里，我们不能排除这种可能性。然而，在后几十年里，专门问题领域的计算要求设计出专用于这些问题的体系结构还是很多的。

在这些重要问题中，涉及到的领域有：

- 高结构数值计算—气象模型、流体流量，有限元素分析；
- 非结构性数值计算—蒙特卡洛模拟，稀疏矩阵问题；
- 实时多面向问题—话音识别、图像识别和计算机视觉；
- 大的存贮和输入输出密集问题—数据库系统，处理系统；
- 图型学与设计系统—计算机辅助设计；
- 人工智能—面向知识库的系统，推理系统。

显而易见，数值领域要求在体系结构中有完善的浮点处理机，而有多种要求的应用也许需要成百这样的处理机。图形系统可能更强调面向定点计算，为开辟窗口和透视观察提供数学支持。然而浮点处理在一些图形领域中则扮演着重要的角色，如需要执行曲线光滑，射线跟踪计算等。人工智能也许并不需要算术能力，但通常它们侧重于记忆存贮。

适合所有提出的领域的需要的高性能体系结构必定给每一个问题领域带来低效率的负担，因为它能力的替代部分对个别具体应用将不是很有用的。如果对一个应用领域的效率太低，那末一个对该领域有效的机器相比较于通用机器则更具有吸引力，因为专门化机器应该造价较低。

价格的优势依赖于专门化机器有足够大的市场，使得开发价格可以横向分布到许多生产出来的付本上。如果仅有少数付本被售出，那优势也就损失殆尽。因此，专门化高性能体系结构在问题领域内应解够通用，使得固定的价格分摊到尽可能大的基础上。

由于专用体系结构被推广到问题广泛的领域，它们的潜在市场增加，但同时它们的硬件使用效率趋向于变低。所以设计者面临着进行权衡，其思想是把专用体系结构同基于通用体系结构的广大市场相平衡。设计者必须在单用和多用体系结构之间找到一个合适的位置，使得新的设计产生在竞争价格上的高性能。因为设计依赖于开发价格和每个产品的生产价格两方面，设计策略正在及时地改变，由于技术的根本进步，上述两方面都在迅速地变化。

1.3.3 VLSI 的影响

由于 VLSI 的发展，高性能体系结构的价格结构有了极大的变化。在 50 年代，那时硬件是如此昂贵，以致一个用户不可能买得起 1M 字节的机器，许多用户共享大型计算机的价格，并发地在单机上运行它们的程序，从而减少了存贮器、处理机和外围设备空间的时间。扩大机器似乎是一种经济的办法，服务的用户数目以计算功能线性地增长，但机器的售价增长趋势却慢得多。

格劳希法则是一种流行可信的经验法则，此法表明计算功能的价格是以计算功能平方根的速率增长。虽然一直到 60 年代初，大量的证据支持格劳希法则，但尚不清楚的是该法则是否反映了性能价格比的基本概念，或者该法则仅仅是支付计算机的售价。

80 年代 VLSI 急剧地改变了计算机的经济学。我们不再追求 1M 字节机器的每个终止周期，而通常是找出一天 24 小时中多数处于搁置不用的那种机器。所以，对大量的小型计算，用户购买 1 台足够大的机器来完成其作业，也许比有某些储备能力的机器还要大一些。买一台过份大的机器，通过许多用户共享周期，获得以机器周期的存取并不是特别经济的。

严格地根据性能观点，我们并不注意朝着用更大的机器来驱动所有的用户而不管他们需要的方案的经济性，尽管以前曾经是这样做的。用比较简单的术语来描述这情形，即如我们所看到的小的作业在小机器上运行，而大的作业运行在大的机器上。今天的小机器有 25 年前大机器同样的计算能力，从前由 100 个用户共享的机器，现在全部由一个用户所拥有。

需要存取共享数据使这里的争论变得有些复杂，因为需要存取共享数据同需要共享机器周期相反正是推动发展的力量，如我们已看到的由大机器或小机器组成的网络支持着当今的许多并发用户。然而我们仍然也看到由许多用户共享的超级计算机，这是因为这些机器周期对任何单个用户实在是太昂贵了。

1.3.4 技术变化对价格的影响

如果我们注意某些时期的基础技术，我们看到类似于图 1.3 那样的一些曲线。这个图显示了每秒百万条指令的测量性能 (MIPS)。它显示出作为 MIPS 性能的每 MIPS 相对价格的粗略图像。图形故意不太明确，是因为它所根据的数据是高度易变的。其思想是曲线由若干台阶段组成，最低的台阶段代表了此段台阶段使用占支配地位技术的计算机每 MIPS 的价格。

在 80 年代，对低性能机器的支配技术是金属氧化物半导体 (MOS)，差不多 80 年初的 NMOS (N 沟道 MOS 器件) 同 CMOS (有 N 和 P 沟道互补的 MOS 器件)，到 80 年代后期则变得很流行。在某一段时间里，每 MIPS 的价格对所有用这些技术做成的机器是完全不变的。

后一个台阶段是后一级的技术，如射极耦合逻辑 (ECL) 那样的双极性技术，每 MIPS 的价格对所有用这些技术的机器几乎是不变的。

第三台阶段更是专门为尖端性能制造的特殊的器件技术。砷镓化合物出现在这段时间里。这一技术价格最高，因为有冷却的要求，制造困难，芯片密度低以及其它类似的因素。这个台阶段对所有由这种技术生产的机器每 MIPS 的价格几乎也是常数。

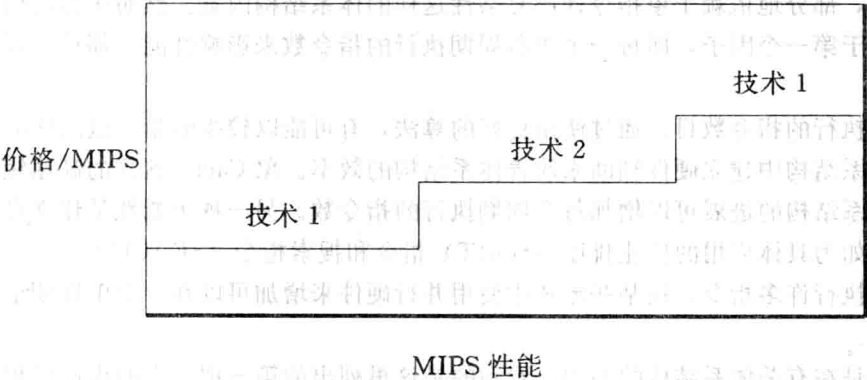


图 1.3 作为性能函数的价格/性能

虽然图 1.3 不太准确，但它主要是用来说明器件技术是如何影响性能价格关系。器件决定了计算机的基本周期时间，粗略地衡量处理能力可以用地址宽度和数据通路乘以时钟频率，这是一个计算机系统信息转换率的上限。