

计算机系统 性能评价

陈兴业 编著

华南工学院出版社

计算机系统性能评价

陈兴业 编著

华南工学院出版社

内 容 简 介

本书介绍计算机系统性能评价的对象及各种评价方法。全书分七章,内容包括性能评价综述;系统测量技术,性能分析技术,系统模拟技术,性能评价在计算机系统的选择问题、改善问题及设计问题上的应用等方面;书中还列举了不少有关计算机系统性能评价方面的应用事例。

本书可作为大专院校计算机科学与工程类专业本科生及研究生教材,也可供从事计算机设计应用方面的工程技术人员参考。

计算机系统性能评价

陈 兴 业 编 著

责任编辑 张巧巧

华南工学院出版社出版发行

(广州 五山)

广东省新华书店经销 广东番禺印刷厂印刷

开本850×1168 1/32 印张:10.75 字数270千

1987年11月第1版 1987年11月第1次印刷

印数:1—3 000 定价2.10元

ISBN 7-5623-0013-5/TP·2

统一书号 15410·039

前 言

计算机系统性能评价是随着计算机的发展而产生的一门重要学科。考虑到计算机是一种既昂贵又先进的计算、控制和信息处理工具，因此，它的设计调整人员，运行维护人员，系统分析人员，用户和管理及计划、采购人员都关心性能评价问题。如果一个系统的工作情况不能达到预期的结果，它的性能/价格比不能令人满意或运行效率不够理想而造成资源浪费，那就要运用性能评价技术来研究造成这种现象的原因，并找出解决问题的方法。迄今计算机系统工程仍不能说是很成熟的，对它进行研究也不象其它领域如力学，电磁学……那样有一套完整的定量的定律。从60年代以来，虽然不少学者及工程技术人员已在这一领域发表了大量的文章，形成了三种主要的性能研究技术。但由于计算机是一个复杂的非线性系统，它的性能与工作负载及应用环境有很密切的关系，因此仍有许多性能研究问题未得到解决，尚有待于广大计算机科学工作者今后努力。

本书共分七章。第一章是计算机系统性能评价概论，扼要介绍计算机系统的主要性能指标及各种评价性能的方法。第二章是测量技术，叙述硬件、软件及固件测量工具及其在性能评价上的应用。第三章是系统性能的分析方法，介绍用数学分析工具来研究系统的性能，使用确定性模型和概率模型来进行研究，在概率模型中使用的主要数学工具是排队论。第四章是系统性能的模拟方法，着重介绍跟踪驱动模拟和概率模拟。第五章是性能研究在选择问题上的应用，讨论在系统规划和采购时从性能评价的角度应考虑的一些基本原则。第六章是性能评价在系统改进中的应用，

研究在改善系统效能时使用的一些诊断和治疗方法。第七章是设计问题的性能评价,简要介绍性能评价在设计问题上的应用,主要介绍迭代式设计法。

本书可作为大专院校计算机科学及工程类专业本科生的选修课和研究生的必修课,教学时数为40到60学时。也可供从事计算机设计应用及管理方面的工程技术人员参考。

本书的内容曾作为研究生课程试用过三届,但由于性能评价技术发展很快,新理论新方法不断涌现,加上笔者本人水平及学识有限,经验不足,书中难免有错误之处,敬请读者批评指正。

编者 1986年12月

目 录

第一章 计算机系统性能评价概论	1
1.1 计算机发展简史	1
1.2 性能评价及性能预测的作用	5
1.3 性能的度量	7
1.4 工作负载特性及性能评价技术	13
1.5 计算机系统性能及模拟研究方法	23
1.6 系统可用性与可靠性问题	33
第二章 测量技术	39
2.1 测量概述	39
2.2 测量工具概述	40
2.3 硬件测量工具	42
2.4 软件及固件工具	56
2.5 测量实验设计	83
2.6 测量结果的表示	98
第三章 系统性能的分析方法	101
3.1 确定性模型	102
3.2 概率模型	108
第四章 系统性能的模拟方法	183
4.1 系统模型的模拟	183
4.2 计算机系统模拟的类型	184
4.3 模拟模型的构成	189
4.4 随机变量的产生方法	196
4.5 模拟器的实现	201
第五章 性能评价在选择问题上的应用	238
5.1 概述	238

5.2	选择方法	240
5.3	性能比较	242
5.4	性能评价在选择问题上应用的例	247
第六章	性能评价在系统改进中的应用	259
6.1	概述	259
6.2	改进方法	260
6.3	诊断	265
6.4	系统性能的改善	292
6.5	改进研究中的评价技术小结	311
第七章	设计问题的性能评价	313
7.1	设计问题概述	313
7.2	设计方法	315
7.3	迭代设计的性能考虑	317
参考文献		335

第一章 计算机系统性能评价概论

1.1 计算机发展简史

计算机是人类历史上发展最快的科学技术之一。自从1946年计算机问世以来，它已经走过了四个发展阶段，并开始踏入另一个新的阶段。人们习惯地把计算机发展的每个阶段称为“代”。

第一代计算机大约从1940~1950年，它的代表是ENIAC机，所用器件是电子管，速度为每个操作 $0.1\text{ms} \sim 1\text{ms}$ 。主存用静电管和延迟线，辅存采用纸带，穿孔卡和延迟线。

第二代计算机从1950~1964年，其代表是IBM7040和7094，采用晶体管器件，速度为每个操作 $1 \sim 10\mu\text{s}$ ，主存采用磁芯和磁鼓，其存取时间为 $1 \sim 10\mu\text{s}$ 。辅存采用磁带、磁盘和磁鼓。同时，开始出现第一批软件系统，例如汇编程序、装入程序、Fortran语言等。而且，浮点算法、中断功能、专用输入输出设备等也得到了发展。此外，软件服务程序如子程序库，批处理监控程序，输入输出控制程序等的使用也大大改善了计算机的使用效率。

第三代计算机是以IBM360系统及CDO6600的出现作为标志的。它采用IC器件，速度达到每个操作 $0.1 \sim 1\mu\text{s}$ 。主存及辅存的容量也大为扩展。除硬件得到大发展之外，软件技术也得到很大的进展，出现了操作系统。操作系统是计算机系统软件的集合，它负责硬件数据及程序资源的分配和控制，能使若干道程序或任务同时在系统上运行，从而把计算机的运行效率推向一个新的高度。此外，它还使用户或程序人员能从一些枯燥无味的工作

中解脱出来，例如，不必再为不同的硬件设备进行数据格式转换等等。操作系统OS/360能为各种型号的IBM360系统服务。

60年代末到70年代期间开发的计算机称为第四代机，其特点为：采用LSI及半导体存储器，存储器划分为不同层次。其典型代表是高速缓冲存储器Cache+主存+磁盘，磁鼓及磁带。程序及数据在不同层次上是自动实现传送的。除性能提高外，这些存储器的价格也下降了。此外，其它类型的新型存储器，例如CCD、磁泡存储器、光存储器及全息存储器等也得到发展。

基于段式或页式或两者综合的动态地址转换技术是60年代中期发展起来的一个重要技术，由这种存储管理策略所支持的存储器层次结构称为虚拟存储器。应用这种存储器技术的典型计算机代表是由MIT和GE联合研制的MULTICS(Multiplexed Information and Computing Service)系统及IBM370。虚存系统结构概念已经进一步扩展为虚拟机的概念，它能同时容纳数个操作系统，彼此动态地共享一个实际的计算机资源。CP—67(它是为IBM360/67而开发的)及后来的VM—370就是这种支持多个虚拟机的操作系统的例子。

60年代末及70年代初出现一些多处理机系统，通过增减处理机个数它就能使处理能力符合不同的需要。在这种系统中，当有某些处理单元失效时，通过性能降级就能维持系统的工作不致中断。一些并行处理机如IBM2938阵列机，ILLIAC IV，CDC STAR—100，TIASC(Advanced Scientific Computer)等就是这种类型机器的杰出代表，它们能在特定的环境下高速地进行运算。

目前，计算机系统结构及硬件/软件技术的进展已不仅仅是引入一些新的应用，它也大大影响了计算机处理信息的方式。现在，联机远程处理及具有大型数据库的系统的交互使用已经非常广泛。现行信息处理业的重点已从常规数学计算变为信息管理，

因此，数据库已日益成为整个系统设计的中心。

计算机应用的增长及改善人——机接口的努力已刺激了新型外围设备及数据输入技术的发展。信息处理及管理的进展已导致计算机通信网络的发展，这种网络的代表就是ARRANET(Advanced Research Project Agency Network)及其后的一些网络。我们可以把这些网络视为资源共享概念的延伸：位于远处之外的一些主计算机的资源（它们的计算能力，数据库及其它功能）能被用户通过陆地上的、无线电的或卫星的链路来使用。繁忙的信息传输要求采用新的共享或多路传输链路，分时多路与信息包切换技术的组合就是这种情况的例子。与大型通用机迅速发展的同时，小型机、微型机的发展也大大加快了。在短短的十多年内，微机的发展已经跨过了四个阶段，进入了第四代超级32位微型机的时代。

上面所述四代计算机都有一个共同的特点，即它们都是以冯·诺依曼结构为基础的，而它的换代则是以器件的革新为标志。第4代计算机虽然功能已经很强，其运算速度已达到每秒钟亿次操作以上，但它们有一定的局限性，归结起来就是：（1）它们的体系结构本身并不反映并行性，并行处理能力不强，主要处理“非知识性”信息，缺乏推理功能，难以直接处理“规则”。

（2）传统计算机只适宜于数值处理，不适宜于非数值处理。在处理文件，图象等非数值对象时功效不高。此外，由于它在本质上是串行的，适于处理顺序问题，如要处理并行问题，则要附加硬/软件措施，从而大大增加系统设计和使用的复杂性，（3）使用不便。由于没有自然语言接口作为人——机接口界面，故要掌握专门的程序设计语言才能用机。

（4）软件问题严重。由于要计算机解决的问题日趋庞大和复杂，软件研制周期长，价格高，可靠性欠佳，生产率低，而硬件功能日趋强大，价格下降，可靠性增加，软件与之不能匹配，

这就导致所谓软件危机。目前，软件价格已占整个计算机系统价格的80%以上，如不解决这个问题，就难以推广计算机的使用。

由于上述软硬件问题，进入80年代以来，国际上已提出开发第5代计算机的计划。前面四代计算机的换代都是以器件革新为主要标志，第5代机则不然，它与传统计算机的主要区别不是采用新一代器件，而是在于功能结构及软件上的突破。在功能上，它的基本特征不是数值处理而是知识处理；在结构上，不再沿用冯·诺依曼结构而采用本身就反映并行性的体系结构，例如数据流计算机来作知识信息项处理，采用分布式结构并与通讯技术密切结合，使第5代机具有智能接口，解题和推理，知识库管理三种基本功能。软件特征则是智能化、自然化、形式化，采用接近自然语言的程序设计语言作为人一机界面，以提高程序的正确性和改善软件生产率。据有关专家的看法，第5代机的主要研究课题有如下几方面：

(1) 核心语言

有人主张用PROLOG语言的某些扩展版本作为第5代机的核心语言。PROLOG是一种逻辑型程序设计语言，是根据与逻辑学中的演绎推理相类似的简单推理而设计的，人们期望它能解决目前软件面临的各种问题。

(2) 推理处理系统

这种系统根据原来所记忆的知识而自动地进行判断推理。为了使其有高性能，就要采用高度并行结构和VLSI技术。数据流型的结构是并行运算的一个理想方案，可以作为推理机的基础。据估计，推理机解题和推理的最大性能指标要能达到100M~1GLIPS（每秒完成的逻辑推理次数），而一个LIPS相当于传统计算机的每秒100~1000条指令，即是说，推理机的处理速度要能达到每秒处理100亿~10000亿条指令的水平，相当于CRAY—XMP（它的浮点处理速度为4.8亿次/秒）巨型机运算速度数十倍以上。

(3) 知识库处理系统

知识库不是指一些数据的集合,而是指象人的知识那样的有关联的情报检索系统,因此,它是管理大量知识信息的机构。知识可以看成是形式不定而结构复杂的通用数据,可以用知识表达语言来描述。知识包括单个事实和表示某些过程的数据等等,存贮知识数据要求存贮量非常大。由于关系模型能表示任何类型的数据,因此知识库可以用支持关系模型的数据库系统来作成。

(4) 智能接口机

它是第5代机人——机对话的工具,它本身就是一种知识信息处理系统。智能接口机的基本功能是通过自然语言、声音、图形来实现人——机通讯,因此,它包括三个方面的内容:①自然语言处理,②声音处理,③图形和图象处理。

第5代机研究的主要目标有2:①用自然语言进行人——机对话,②发展机器翻译系统和问题回答系统。

此外,还要发展VLSI技术,使单个芯片上能集成一百万个~一千万个晶体管元件,才能满足上述的需要。

以上就是自40年代以来计算机的发展概况。与计算机技术发展的同时,计算机系统性能评价技术也相应得到发展。虽然它的起步比计算机晚,但自50年代以来,许多学者这方面进行过大量工作,使性能评价成为促进计算机系统研究和发展的有力工具。

1.2 性能评价及性能预测的作用

性能评价及性能预测在计算机的发展史上与用户有密切关系。从计算机系统结构设计开始,到安装后的每日运行,都有必要对系统性能进行评价和预测。在计算机系统的早期设计阶段,厂家通常需要作两种类型的预测,第一种类型是预测应用的性质及这些应用的系统负载情况。这里,工作负载指的是对系统的服务要

求量。第二种类型的预测与系统结构设计的选择有关，它基于可用的软/硬件技术。这里，选择的准则是合适的性能/价格比。

一旦作出系统结构决定并开始设计及实现，性能预测及评价的规模就确定了。机器怎样去组织最好？采用什么样的操作系统？它应提供什么功能？操作系统各组成部分之间的相互作用关系，例如作业调度算法，处理机调度及存贮管理等都必须加以明确，它们对性能的影响也必须加以预测。在设计及实现阶段，用于性能评价及预测的技术包括从简单的手工计算到精心构成的模拟。通过把预测的性能与实际达到的性能相比较，常常可揭示在设计上或系统程序设计上出错的主要问题。现在，人们都接受这种观点，即：性能预测及评价过程应该是系统设计及实现阶段整个开发努力的一部分。

当一个新产品被开发出来后，计算机厂家必须作好准备为各种特定的应用场合及各种潜在的用户进行性能预测。厂家必须提出合适的软硬件组织及其最佳配置，以便为各种用户的要求提出一种最好的解决方案，这种工作称为系统配置过程。虽然在系统开发阶段所用到的性能预测，评价工具及评价方法也可用于系统配置的目的，但它也有一些附加的要求。计划中的用户环境必须转换为一组能用作性能预测模型输入的量化参数，这也是个工作负载特性问题。

为用户安装好计算机系统之后，厂家或服务机构必须检查一下该系统是否已实现了预定的性能要求，这就是所谓调整，这种调整活动传统上都是根据经验和直觉。但是，现代大型机是很复杂的，仅靠经验和直觉难以达到整体优化及找出稳定的工作点，系统的调整要求对各部分互相之间的作用情况有很清楚的了解。现在，执行这种任务的系统方法仍是不完善的，还有待于今后的研究和开发努力。

1.3 性能的度量

上面我们经常提到性能，但没有对它下过精确的定义。现在，让我们来研究这个问题。当我们说，“这个计算机的性能很好”的时候，可能指它的服务质量超过了我们的预定期望。但是，服务质量的度量及期望的程度是因人而异的，不管对系统设计人员，终端用户还是管理人员都是如此。

我们可以把性能度量划分为两类：面向用户的度量及面向系统的度量。前者包括在批处理系统环境下的周转时间（turnaround time）和在实时或交互式系统环境下的响应时间（response time）。作业的周转时间指的是自提交作业起，到处理结果可用为止所经历的时间。而响应时间指的是从服务要求抵达到服务完成之间的时间。人们对响应时间可以有不同的定义，其主要差别是对请求抵达及完成时刻的定义不同，例如，可以定义抵达时刻为用户按下回车键的时刻，而完成时刻则可定义为在终端上打出第一个字符输出的时刻。当要用响应时间来比较性能时，必须同时说明它是如何定义的。

在交互式系统中，有时使用系统反应时间（reaction time）这个词，它指的是从一个输入抵达系统到系统收到第一个服务时间片之间的时间间隔，它测量调度程序把服务分配给新抵达的输入请求的效率。周转时间、响应时间及反应时间全都是随机变量。

通常，我们把作业或请求划分为不同的优先类别，对每类的不同作业可分配给不同的优先数。确定及分配作业的优先度的因素是作业的紧急程度，它的重要性，资源要求特性等等。

面向系统的典型性能度量是吞吐率（throughput）及利用率（Utilization）。吞吐率定义为单位时间内处理作业的平均数，它表示该系统能提供的生产能力。如果作业按照某种独立于

系统状态的抵达机构到达系统内，则吞吐率就等价于平均抵达率。这时，假定系统能完成那些抵达的作业而不会产生作业堆积和无法及时完成的状况，否则，吞吐率就不足以代表系统的性能而仅仅是系统负载的度量。

当系统中经常有某个作业在等待系统服务，或作业抵达率与系统状态有关时吞吐率才有意义。在排队系统中，如果排队的顾客数是无限地增加的，就意味着系统的服务情况堆积，这时就难以表达系统的吞吐率。不过，在实际中我们可以在输入排队为非空的一段有限时间内定义并测量吞吐率，这样定义的吞吐率是系统能力的一个恰当的指示值。假定在一个交互式系统中，实际工作的终端用户数是个有限值 N ，我们进一步假定当某终端的请求已送入系统中，在等待服务或正在接受服务时它就不再送入新的输入请求。即是说，若已有 n 个作业在系统中，则只有 $N-n$ 个终端有资格发出服务请求，于是，有效的抵达率就是个线性递减的系统状态函数。在批处理系统中我们可以想像一种类似的情况，可能有足够多的用户使该系统始终保持忙碌，但实际上，随着系统繁忙程度的增加，用户可能会不乐意再提交新作业，所以，作业抵达率也将是未完成作业数目的递减函数。正是在这种作业产生机构中固有的负反馈环使系统能保持稳定。

某个资源的利用率就是它处于忙状态所占的时间百分比。例如CPU，当它处于非空闲状态时，它可能处于问题状态或管理状态。前者表示CPU花在执行用户程序的时间，后者表示花在执行操作系统某部分上的时间（例如调度程序，各种解析处理程序等）。前者又称为有用的工作，后者通常称为系统开销。但应注意，其实在管态下的操作也给用户程序提供很多有用的服务。因此，不要因为使用开销这个词面对它有所误解，以为它对用户程序是无用的。

若我们假设有一个单CPU系统，且CPU的利用率不包括管

理状态对用户程序有用的部分，则可发现在吞吐率 T （每秒作业数）和CPU利用率 ρ_{cpu} 之间有如下简单关系：

$$\rho_{cpu} = T \bar{S}_{cpu} \quad (1.1)$$

式中 $\bar{S}_{cpu}$ 是每个作业花费的时间（以秒计算），它代表处理一个作业所要花费的CPU平均时间。

平均响应时间 $\bar{t}_R$ 与吞吐率的关系如(1.2)式所示：

$$T \bar{t}_R = \bar{n} \quad (1.2)$$

式中， $\bar{n}$ 表示系统中的平均作业数（在等待或正在服务的作业数）。式(1.1)及(1.2)均是第三章中所述的Little公式的特殊情况。

上面我们提到性能指标是用一个数字或一些数字来表示的，例如周转时间、响应时间、反应时间等，它们都是随机数，除了可以用分布函数来表示外，也可用它们的数字特征如平均值（数学期望）和方差来表示。例如，设有一个交互式系统，有6个用户在工作，它们几乎在同一时刻输入6条命令，系统在多道程序环境下工作。假定当命令输入时，要执行的用户程序都已在内存中，且任何一个都不要求磁盘或磁鼓服务，在处理这些命令期间假定没有其它命令送入。这6条命令中有4条是编辑类型的，每条占用一个CPU时间单位，其余两条是编译命令，每条占用4个CPU时间单位。CPU执行命令的方式是FCFS（先来先服务）。设第一条是编辑命令 C_1 ，第二、三条是编译命令 C_2, C_3 ，其后依次是三条编辑命令 C_4, C_5 和 C_6 。此交互系统按分时方式工作，每个时间片为2个时间单位，当某进程超过时间片时就被挂起，加入到排队中去，排队规则是FIFO（先入先出）。于是，下个进程就从队中取出，投入运行。CPU由一个进程切换到下一个进程的时间设为0.2个时间单位，这等于开销时间。这6个命令

的处理情况如图1.1所示。

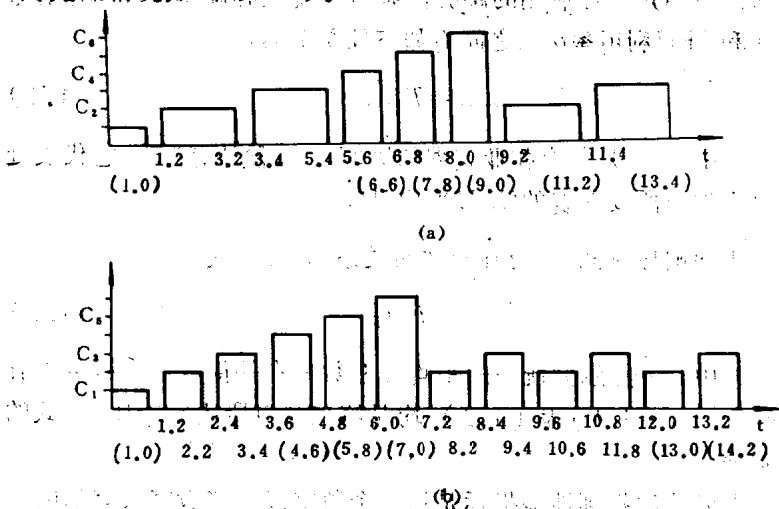


图1.1 命令 $C_1 \sim C_6$ 的执行时间图

(a) 时间片为2个单位；(b) 时间片为1个单位，

括号中的数字为命令处理完成时间。

因为这6条命令均是在 $t=0$ 时送入，因此，括号内的时间也就是响应时间。它的平均响应时间 $\bar{t}_R$ 是8.16个时间单位，其方差 σ 为9.55个时间单位。显然，编辑用户对系统的编辑命令的响应特性更感兴趣，而编译用户则对编译命令的响应特性有兴趣，故我们可以将这两种命令的性能指标分开。对图1.1(a)，编辑命令的平均响应时间是6.1个时间单位，而对编译命令则为12.3个时间单位。

对计算站的经理来说，他感兴趣的東西往往与用户不同，他关心的是系统的生产力。很明显，若每个命令一抵达就马上执行，则平均响应时间最小。当系统只给一个用户使用时就这种情况。而对多用户系统，则只有在系统的负载很轻时才能做到这点。但是，在“用户思考时间”的大部分时间内，CPU是处于