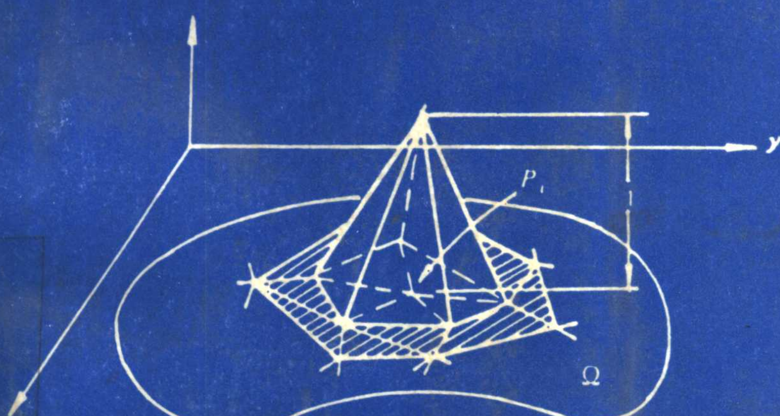
并行算法引论

陈景良 编著



石油工业出版社

并 行 算 法 引 论

陈景良　编著

石 油 工 业 出 版 社

（京）新登字082号

内 容 提 要

本书论述在科学计算中80年代以来新发展起来的常用的并行和向量算法。该书共分六章，第一章介绍向量和并行计算机的一些基本特征、适合这些机器处理的算法的框架以及若干基本概念。其余五章依次讨论线性方程组直接法和迭代法、非线性方程组解法、偏微分方程解法、FFT算法。在写作上注意深入浅出、系统化、揭示其思想方法。

本书可供从事并行计算的工作者和需要应用并行计算的工程技术人员使用，亦可作为计算数学、计算机硬件与软件等专业的工作者、研究生和大学生的参考书或选修教材。

并行算法引论

陈景良　编著

*

石油工业出版社出版

（北京安定门外安华里二区一号楼）

北京密云华都印刷厂排版印刷

新华书店北京发行所发行

850×1168毫米　32开本　印张12 [illegible]　印1—1,000

1992年8月北京第1版　1992年8月北京第1次印刷

ISBN 7-5021-0730-4/TE·690

定价：7.00元

序　言

70年代，最早的几种并行计算机和向量计算机提供给科学界时，由于数量少，且主要归少数部门使用，故产生的影响是比较有限的，人们对并行计算的意义尚处在一个实践与认识的过程中。然而，从那时到现在只经过二十来年，涓涓细水已汇成滔滔洪流，并行计算获得了举世瞩目的发展。计算机并行化日新月异，投入使用的机器数量大增，结构多样，更有数不清的包括具有数以百、千、万计的处理机系统正在研制或计划研制之中。并行算法也取得长足进步，研究领域和应用范围迅速扩大，文献资料不计其数。并行计算已受到各界学者越来越多的关注。

并行算法，这个把并行与向量计算机与计算数学紧密联系在一起的新兴学科，是研究对由多个可以并行执行的进程构成的给定计算问题在某一特定并行计算机上最佳实现的一门学问。它的研究内容包括：数值并行算法、非数值并行算法、并行算法设计和并行计算的复杂性分析与性能测试等。并行算法可以有不同的分类方法，大致可分为：SIMD算法、同步MIMD算法和异步算法。

本书命名为《并行算法引论》，是因其只讨论数值并行算法。由于任何一个可实现的并行算法都与计算机的体系结构密切相关，所以，本书的绪论中首先对并行计算机的体系结构及分类进行了介绍。

发展向量计算机和并行计算机最初的主要推动力多是科学计算。本书的宗旨是讨论与介绍科学计算中常用的并行和向量算法，共分六章。第一章讨论向量计算机和并行计算机的一些基本特征，适合这些机器处理的算法的框架以及若干基本概念。第二

章和第三章分别论述求解线性方程组的直接法与迭代法。在科学计算中最重要的问题之一是解线性方程组，即使对传统计算机来说，它仍然是一个活跃的研究领域，尤其是迭代方法，矩阵运算是线性方程组解法中的基本运算，我们专门集中地作了详述。对于直接方法主要突出它们在向量与并行计算机上的组织方案。对于迭代方法则大多注重它们与计算机系统无关的基本性质，原因是对它们的处理大多比较容易些。第四章是非线性方程组的数值解法，包括传统方法的并行化(传统方法通常不适合向量处理)以及新的并行方法。本章重点放在近些年新发展的专门适应并行系统的多分裂方法和异步迭代方法。传统方法仍然是建立新方法的基础。第五章介绍偏微分方程数值解法、差分方法与有限元方法是偏微分方程的两种基本解法。这两种方法本身具有良好的向量化度和并行化度，我们通过典型例子阐述它们的基本思想和线索，特别把近年发展起来的组显式差分格式放在显著的位置上，此类格式有可能兼备显式格式与隐式格式的优点，即计算简单、有高度并行性，且稳定性好。这一章还通过典型例子讲解了区域分裂方法的重迭型分裂和非重迭型分裂两个方向。由于并行计算的需要，近年来区域分裂方法受到人们广泛的重视。最后，第六章占较小的篇幅，讲述快速Fourier变换(FFT)的并行实现，以及应用FFT进行离散褶积（卷积）计算。

本书有以下几个特点：

第一，侧重于基本方法及80年代以来发展的新方法，且加以系统化。

第二，算法设计紧密结合机器，但也避免了仅仅针对某种特定的向量机或并行机而讨论算法。可以这样说，我们正处在计算机结构历史上变更最频繁的时期，目前还很难设想能通过精选使影响算法的结构单一化。这就使得即便是最基本的算法也因机器结构的不同而需不断作重新考虑，而且这种情况还会持续许多年。因此，我们在设计算法时，重要的是在面向机器的同时揭示

出哪些本质上与机器无关的共性途径。本书正是这样做的。

第三，表述算法没有采用某种向量语言或并行语言，也没有完备的规则，而主要是解说清楚其基本步骤。或者说，本书提供的是思想方法，而不是基本运算程序。

第四，该书假定读者学习过数值分析课程。因此，对传统数值方法不作详细描述和推导，虽然为了完整叙述了一些有关的结论，但不加证明。关于向量算法和并行算法的描述和推证，不少是作者加工或重新给出的，其中也包含作者的某些科研成果。这些在表达上都力求做到深入浅出。

在本书的编写过程中，梁国珍同志校阅和整理了全部书稿，汪肖蕙同志绘制了所有图稿。

悬请读者指正。

作　者

1991年8月于清华园

目　　录

第一章　绪论 ……………………………………………………… (1)
§1　向量计算机和并行计算机 ……………………………… (1)
1.1　超级计算和超级计算机 ……………………………… (1)
1.2　向量计算机 ………………………………………… (3)
1.3　并行计算机 ………………………………………… (9)
1.4　通信和同步 ………………………………………… (23)
§2　并行算法 …………………………………………… (26)
2.1　算法的概念 ………………………………………… (26)
2.2　并行算法的分类 …………………………………… (31)
2.3　并行算法发展概况 ………………………………… (37)
§3　并行化与向量化的基本概念 ……………………… (44)
3.1　并行度与向量化度 ………………………………… (44)
3.2　加速 ………………………………………………… (46)
3.3　相容性 ……………………………………………… (52)
第二章　线性方程组直接法 ……………………………………… (55)
§1　矩阵运算 …………………………………………… (55)
1.1　矩阵与向量乘法 …………………………………… (55)
1.2　矩阵与矩阵乘法 …………………………………… (61)
1.3　稀疏矩阵乘法 ……………………………………… (69)
§2　LU分解 ……………………………………………… (78)
2.1　Gauss消去法 ……………………………………… (78)
2.2　LU分解的向量算法 ……………………………… (80)
2.3　LU分解的并行算法 ……………………………… (90)
2.4　带状矩阵的LU分解 ……………………………… (100)
§3　Choleski分解 ……………………………………… (102)
3.1　*ijk*型向量算法 ……………………………………… (102)
3.2　分解的并行实现 …………………………………… (104)

§4 正交约化方法 …………………………………………………… (106)
4.1 QR分解…………………………………………………………… (106)
4.2 Householder约化法……………………………………………… (107)
4.3 Givens约化法……………………………………………………… (114)
§5 方程组的解 ……………………………………………………… (121)
5.1 三角形方程组 …………………………………………………… (121)
5.2 小带宽带状方程组 ……………………………………………… (127)
5.3 多右边的方程组 ………………………………………………… (136)
第三章 线性方程组迭代法……………………………………………… (140)
§1 Jacobi法 ………………………………………………………… (140)
1.1 迭代法及其收敛性 ……………………………………………… (140)
1.2 模型问题 ………………………………………………………… (143)
1.3 分块方法 ………………………………………………………… (153)
1.4 ADI方法 ………………………………………………………… (157)
§2 Gauss-Seidel法与SOR法 ……………………………………… (160)
2.1 基本描述 ………………………………………………………… (160)
2.2 红—黑排序 ……………………………………………………… (165)
2.3 多色排序 ………………………………………………………… (173)
2.4 多色LSOR法和SSOR法 ………………………………………… (178)
2.5 数据流SOR ……………………………………………………… (183)
2.6 半迭代方法 ……………………………………………………… (184)
§3 极小化方法 ……………………………………………………… (189)
3.1 线性方程组的解与二次函数的极值 …………………………… (189)
3.2 若干基本极小化方法 …………………………………………… (192)
3.3 预处理共轭梯度法 ……………………………………………… (199)
第四章 非线性方程组数值解法………………………………………… (216)
§1 预备知识和基本同步算法 ……………………………………… (216)
1.1 预备知识 ………………………………………………………… (216)
1.2 Newton迭代法 …………………………………………………… (222)
1.3 割线法 …………………………………………………………… (225)
1.4 线性和非线性复合方法 ………………………………………… (229)
§2 多分裂方法 ……………………………………………………… (234)

2.1 线性多分裂 …………………………………………… (234)
2.2 非线性多分裂 ………………………………………… (236)
§3 异步迭代方法一般论述 ………………………………… (244)
3.1 异步迭代基本描述及类型 ……………………………… (244)
3.2 收敛定理 ……………………………………………… (247)
3.3 带记忆的异步迭代方法 ………………………………… (252)
§4 几种异步迭代法 ………………………………………… (256)
4.1 Newton型异步迭代方法 ……………………………… (256)
4.2 异步非线性*SOR*型方法………………………………… (265)
第五章 偏微分方程数值解法……………………………… (271)
§1 差分方法 ……………………………………………… (271)
1.1 显式格式与隐式格式 ………………………………… (271)
1.2 组显式方法 …………………………………………… (286)
§2 有限元方法 …………………………………………… (293)
2.1 方法大意 ……………………………………………… (293)
2.2 三角形线性元 ………………………………………… (301)
2.3 其它有限元离散方法 ………………………………… (315)
2.4 并行实现基本问题 …………………………………… (324)
§3 区域分裂方法 ………………………………………… (325)
3.1 重迭型分裂 …………………………………………… (325)
3.2 非重迭型分裂 ………………………………………… (331)
第六章 快速Fourier变换 ………………………………… (340)
§1 Fourier变换简介 ……………………………………… (340)
1.1 Fourier积分与Fourier变换 ………………………… (340)
1.2 有限离散Fourier变换 ……………………………… (342)
§2 FFT及其并行计算………………………………………… (346)
2.1 Cooley—Tukey算法 ………………………………… (346)
2.2 并行FFT………………………………………………… (351)
§3 离散褶积计算 ………………………………………… (363)
3.1 褶积 …………………………………………………… (363)
3.2 循环褶积的并行计算 ………………………………… (366)
参考文献……………………………………………………… (369)

第一章 绪 论

§1 向量计算机和并行计算机

1.1 超级计算和超级计算机

70年代初，出现了包含多个算术逻辑部件在一个或多个控制部件控制下进行并行操作的并行计算机及备有向量硬件指令的向量计算机。1972年ILLIAC Ⅳ计算机开创了并行机的新纪元，在此前后TI-ASC，CDC STAR-100等等一代向量机问世。这些机器突破了传统的串行计算机只有一个算术逻辑部件或仅依赖软件方式处理向量的概念，使其最高速度达每秒几千万乃至亿次操作。于是，产生了“超级计算（Supercomputing）”一词，它作为能以非常高速度实现大规模计算的硬件（超级计算机）、软件、算法与语言的总称。有时也用它称谓在向量机和并行机上的计算。

自70年代中期开始，第二代向量处理机大放异彩，其向量处理速度为100～1000MOPS(Million Operations Per Second——每秒百万次操作)，并成为一种产业。第二代向量机又可分为：第一代超级机，以Cray-1（1976）为代表，速度为3～160MFLOPS(Million FLOating Point Operations Per Second——每秒百万次浮点运算)，平均20～80MFLOPS；第二代超级机，如Cyber205(1980)，向量速度为800MFLOPS，标量速度为50MIPS（Million Instructions Per Second——每秒百万条指令）。日本的富士通(Fujitsu)、日立(Hitachi)和电气公司(NEC)相继推出VP、S—810和SX系列；如

VP—100、VP—200(1982)、VP—400(1985)速度分别为267、533、1140MFLOPS。中国推出了YH—1(1983)。到80年代中期为止的十来年间是并行计算机获很大成功的时期。一方面，继ILLIAC Ⅳ之后，有若干单控制器多处理部件的计算机问世，如BSP(1979)含16个PE(Processing Element——处理部件)，其速度达50MFLOPS；MPP(1983)含128×128=16384个PE，进行八位整数运算时速度达1861～6553MOPS；还有英国的DAP(1978)等。另一方面，更多的是一些多控制器多处理部件的计算机的出现或研制，投入运行的如Cray X—MP(1983)含2台CPU(Central Processing Unit)——中央处理机，主要由PE与控制部件组成)，它由Cray—1发展而成，运算能力却为之5倍，峰值为400MFLOPS。此间，苏联研制的有EI′brus系列，M—10，MARS—M，ES—1766，PS—×000系列等，如PS—2000含8～64个PE，约1983年出成品，其运算速度达200MIPS。

自80年代中期以后进入了超级计算蓬勃发展时期，其主要标志为：(1) 超级计算已成为当今科学技术进步的主要推动力之一，它被视为一个国家科学技术发展和实力的显著象征。因此，尽管超级机的研制需花费巨额资金，而且推广应用也需用大量的钱，还是有越来越多国家的大学、研究所、实验室和公司从事超级机和并行处理的研究。(2) 超级机的系统结构百花争艳，千姿百态，由浅度并行向着深度和极度并行方向推进。到1989年为止各国所研制的各种超级机的系统结构形式差异很大；含处理机(指PE或CPU或微处理机或本身也是超级机)的台数2至2^{16}=65536不等，目前有人将处理机台数<100、100～10000、>10000的超级计算机分别称为是浅度、深度、极度并行的计算机；它们的浮点运算速度由每秒千万次至220亿次不等；应用目的从以科学技术与工程领域中的问题为主的数值计算发展到模拟人类智力活动的人工智能.(3) 并行算法与向量化并行化的研究

已形成一股热潮。并行算法的研究已有三十多年的历史，向量化并行化技术研制也有十余年历史。现在同步并行算法与向量化技术已基本成熟。目前所有向量机均备有向量化库程序和向量识别器。并行化理论研究也取得了较大的进展。一些多机系统已备有一定水平的并行化软件.（4）新的概念、词汇术语层出不穷，学术活动频繁。每年都有一个乃至几个以超级计算为主题的大型国际会议。各种专著及文献资料迅猛增加。

1.2　向量计算机

一般称采用流水线技术且备有向量硬件指令的计算机为**向量计算机**。

流水线的用途是处理能分成若干子任务的任务。这里“任务”可小至一条指令一个运算，大至一个问题。例如一条指令可分成读取、译码、取操作数、执行等子任务。又如浮点加法可分成对阶、尾数相加、规格化、处理上溢与下溢等子任务。**流水线**由若干处理段组成，每个处理段（小至如加法器大至如一台处理机）执行一个子任务，各段能同时执行，见图1.1。含n段的流水线加工m（$\gg n$）个任务，速度可近似地加快$n-1$倍。

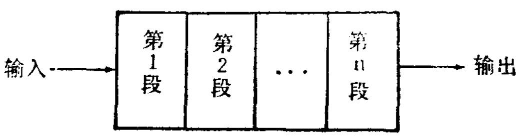
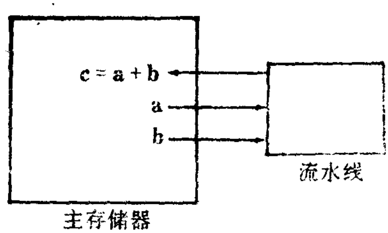


图　1.1

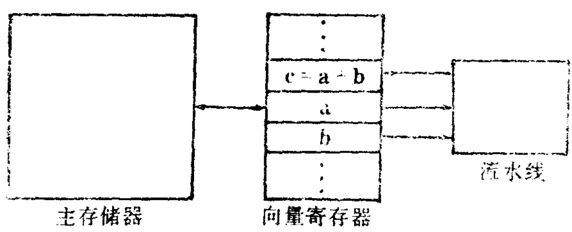
在向量机中使用流水线处理向量，这里的“向量”泛指一串有序的数。所谓**向量硬件指令**，其能处理一个或多个向量，执行时由向量长度寄存器并行控制，可省略标量处理方式对循环控制变量的修改和判别。

硬件为向量运算提供操作数有两种类型。一类是**存储器至存储器**型，即直接从主存储器取数，且将结果存回主存储器。这种

类型操作数的优点是对**向量长度**（即分量的个数）无须特殊关照，缺点是存取速度较慢。CDC所研制的向量机从STAR-100至Cyber系列均属此种类型，见图1.2(a)。另一类是**寄存器至寄存器**型，即进行向量运算时流水线运算部件不能直接对主存储器存取操作数，而只能从向量寄存器取数，并将结果存回向量寄存器。寄存器是存取速度比较快的存储单元，可直接与主存储器交换信息。这样由一定数量寄存器组成的向量寄存器充当主存储器与流水线运算部件之间的媒介，可以较大的提高向量处理的速度。然而由于寄存器成本较高，设置的数量受限，故对长向量必须分段处理。Cray研究公司70年代以来生产的向量机有八个向量寄存器，每个保存64个浮点数，见图1.2(b)。



(a)



(b)

图 1.2

向量机在进行向量运算时，对主存储器中如何存放的元素能构成一个向量是有限制的。例如，具有固定**步距**s（即地址为a，

$a+s$，$a+2s$，…）的元素可视为一向量；特别$s=1$，称为**顺序可寻址元素**构成的向量，有些机器（如CDC的向量机）只能有这种向量。因此，经常需要利用辅助硬件或软件指令，通过聚集（将地址表说明的元素映射成向量）、散布（聚集之逆——按地址表存放向量的元素）、压缩（将固定步距$s>1$的元素映射成顺序可寻址元素）等运算将数据调整为可接受的向量。在设计向量机的算法时首要问题之一是要考虑数据的布置，以使所需的数据管理的开销极小。

向量硬件运算通常有两个向量的加法、向量的数乘、两个向量对应元素的积或商、一个向量各元素求倒数或方根；更复杂些的运算有向量的内积、一个向量各元素之和、以及聚集、散布、合并、压缩、稀疏向量运算等等。标量处理执行一条指令得一个结果，通常用MIPS为单位来评价机器的速度。向量处理执行一条（向量硬件）指令可得多个结果，一般以MFLOPS为单位来度量机器的速度。

向量机采用的流水线运算部件有不同的方式。一种是**单功能流水线**，即一条流水线仅实现诸如加法或乘法等运算中的一种运算。象Cray机和日本NEC SX—2等有多条单功能流水线。另一种是**多功能**或称**可变换结构**流水线，即一条流水线可执行不同的算术运算，但在一种指定运算（比如加法）实现之前需对流水线作适当的结构配置。CDC的机器就采用此种方式，例如Cyber 205允许用1，2或4条多功能流水线。

使用处理向量的流水线时，**向量运算时间**T有如下近似公式：

$$T=S+KN \tag{1.1}$$

其中N是向量长度；K是结果离开流水线的时间间隔，称为**结果速度**；S是从取操作数至流水线装满所用的时间（对多功能流水线来说还包含流水线结构配置的时间），称其为**起动时间**。存储器至存储器型机器比寄存器至寄存器型机器一般要花费更多的运算时间。

如果仅用一条流水线处理向量，这时的结果速度K也称为该流水线的**时钟周期**或**循环时间**，记作τ。时钟周期是向量机的主要性能指标之一。几种机器的τ如下：

Cyber205	Cray—1	FACOM VP—200	NEC SX—2
20ns	12.5ns	7.5ns	6ns

当使用多条重复设置的流水线处理向量时，结果速度K=时钟周期τ/流水线条数。例如Cyber 205对加法与乘法用2条流水线时$K=10$ns，用4条流水线时$K=5$ns；NEC SX—2有4条流水线用于加法与乘法，$K=6\text{ns}/4=1.5\text{ns}$。在做平方根或内积等较复杂的运算时$K$的值自然要更大一些。

现有的向量机，K多为几个毫微秒的数量级；而对于起动时间S，寄存器至寄存器的机器为几十毫微秒，存储器至存储器的机器为几百毫微秒。由（1.1）式可知，每个结果的时间T_R是向量长度N的函数

$$T_R = T/N = K + S/N \tag{1.2}$$

图1.3是$N—T_R$关系曲线，它表明了充分长的向量才能抵消起动时间的影响。

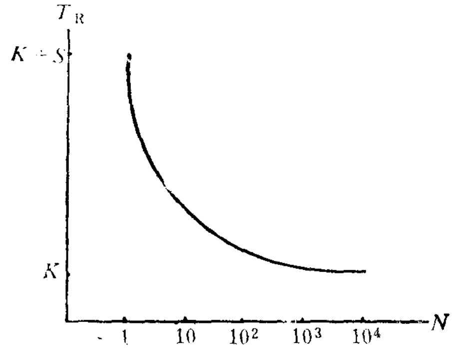


图 1.3

可以用另一种方式来定义结果速度。将结果速度定义为每单

位时间的结果数目并记作R，即

$$R=\frac{1}{T_R}=\frac{N}{S+KN} \tag{1.3}$$

由此，若$S=0$或$N\to\infty$，得

$$R_\infty=1/K \tag{1.4}$$

R_∞称为**渐近结果速度**，实际上它是一种只能趋近而不能达到的速度。例如，$K=10$ns时$R_\infty=10^8$结果/s（对浮点运算来说为100MFLOPS）.图1.4中给出$K=10$ns、$S=100$ns及$S=1000$ns时的$N-R$关系曲线，它表明随N的增大两曲线均趋于$R_\infty=$100MFLOPS，但较小S的曲线开始时趋近得更快。

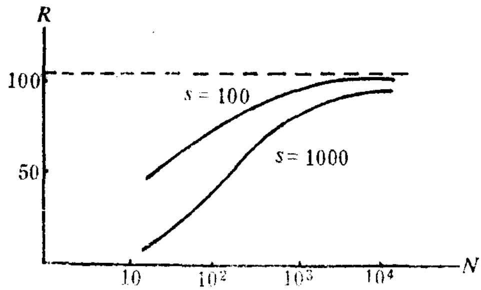


图 1.4

还有两个值得关注的数字。一个数字是达到渐近速度之半的向量长度，记作$N_{1/2}$，称为**半性能长度**。依（1.3）与（1.4）式，从

$$\frac{N}{S+KN}=\frac{R_\infty}{2}=\frac{1}{2K}$$

解出的N即为$N_{1/2}$，得

$$N_{1/2}=S/K \tag{1.5}$$

例如若$K=10$ns，则$S=100$ns时$N_{1/2}=10$，$S=1000$ns时$N_{1/2}=100$。另一个数字是使向量运算转变快于标量运算之向量长度，

记作N_C，称为**交叉点**。设标量运算平均速度为R_S，则N_C是使不等式

$$R = \frac{N}{S + KN} \geqslant R_S$$

成立的N的最小值。由此及R是N的递增函数，可得

$$N_C = \lceil SR_S/(1 - KR_S) \rceil \tag{1.6}$$

记号$\lceil x \rceil$表示不小于实数x的最小整数。例如$R_S = 10\text{MFLOPS}$，$K = 10\text{ns}$，当$S = 100\text{ns}$时

$$N_C = \lceil 100 \times 10^{-9} \times 10 \times 10^6/(1 - 10 \times 10^{-9} \times 10 \times 10^6) \rceil$$
$$= \lceil 1/0.9 \rceil = 2$$

这说明除向量长度为1即退化为标量的情形之外，向量运算更为有效，因此实际上对向量的长度并无限制；当$S = 1000\text{ns}$时

$$N_C = \lceil 1000 \times 10^{-9} \times 10 \times 10^6/(1 - 10 \times 10^{-9} \times 10 \times 10^6) \rceil$$
$$= \lceil 10/0.9 \rceil = 12$$

这时对于长度小于12的向量，使用向量运算将慢于标量运算。显然，交叉点高度依赖于标量运算速度，后者越慢（即R_S相对R越小）N_C越小。在前面的例子中，若R_S仅为5MFLOPS，则当$S = 100\text{ns}$时，$N_C = 1$，当$S = 1000\text{ns}$时，$N_C = 6$。

大多数向量机为标量运算提供有单独的部件，这些部件也常采用流水线结构。这种标量流水线按标量方式接受操作数，而不象向量流水线那样接受的是向量操作数，它们能与向量流水线同时运行，产生标量结果的速度比向量流水线的最高速度通常要慢5～10倍。

链接是一种十分吸引人的特性，它指的是具有多条运算流水线的机器常提供把若干流水线链接在一起的可能性，从而使得一条流水线的结果直接送至另一条流水线而无须先发送回寄存器或存储器。因此它可以非常有效地实现某些类型的向量运算，例如链三元组等。**链三元组**是形如$\boldsymbol{a} + \alpha\boldsymbol{b}$，$(\boldsymbol{a} + \alpha)\boldsymbol{b}$之类的运算，其中$\boldsymbol{a}$和$\boldsymbol{b}$是向量，$\alpha$是标量，$(\boldsymbol{a} + \alpha)\boldsymbol{b}$是$\alpha$加到$\boldsymbol{a}$的每个分量并