

经典解读

# 阿姆达尔定律原文“论以单处理器的方式实现大规模计算能力的有效性”<sup>1</sup>解读

作者: 吉恩·阿姆达尔 (Gene Amdahl)

编译者: 周嘉鹏 张映璇 刘宇航

编者按: 阿姆达尔定律 (Amdahl's Law) 是计算机体系结构的基本规律之一。阿姆达尔定律和摩尔定律虽然都是定律, 但二者性质不同。摩尔定律不仅反映了物理上的必然性, 还反映了市场竞争等经济学上的必然性。阿姆达尔定律反映了物理上的必然性, 是纯粹的定律。阿姆达尔定律讨论的主题包括“瓶颈、并行、串行、系统均衡”这些关键词, 毫无疑问, 这些关键词是设计任何一个计算机系统都必须始终优先考虑的, 因此 56 年前的这篇经典文献值得仔细回顾和深思。

## 引言<sup>2</sup>

十几年来, 预言家们一直声称单个计算机的结构(性能)已经达到极限, 计算机再想获得真正意义上的显著进步, 只能将很多计算机互连在一起合作求解问题。大家都指明了各自认为正确的研究方向, 有人觉得是具有通用化互连存储器的通用计算机, 有人觉得是拥有与几何学相关的存储器互连网络, 并由一个或多个指令流控制的专用计算机<sup>3</sup>。

本文展示了单处理器方法的持续有效性(continued validity), 以及多处理器方法用于解决实际问题的缺点和随之产生的不规则性(irregularities)<sup>4</sup>。

本文的论据是基于过去十年计算机计算的统计特性以及实际问题中的运算要求。同时我们还参考了斯坦福大学商学院教授肯尼斯·奈特(Ken-

neth F. Knight) 于 1966 年 9 月在《自动数据处理》(Datamation) 杂志上发表的一篇文章“计算机性能数据的变化”, 这篇文章可以说是截至目前对相对计算能力分析最透彻的文章之一。

## 串行负载的比例

我们关注的第一个特征是与数据管理总务(data management housekeeping)相关的计算负载的比例<sup>5</sup>。该比例近十年来几乎从未改变过, 并且占据了程序运行时被执行指令的 40%<sup>6</sup>。在完全专用的计算机中, 这个比例可以减少一半, 但不太可能减少三分之二<sup>7</sup>。这些开销的性质是串行的, 因此不太适合用并行处理技术进行处理<sup>8</sup>。仅开销一项就将吞吐量的上限定格在串行处理速度的 5~7 倍, 即使数据管理总务

在一个单独的处理器中完成。而问题的非数据管理总务部分对处理器性能的利用 (exploit) 可以达到数据管理总务处理器性能的 3~4 倍。在这一点上我们可以得出一个很明显的结论:在高并行处理速度上所做的努力是白费的,除非能同时以几乎相同的幅度改善串行处理速度<sup>9</sup>。

## 影响并行度的非规则性等因素

数据管理总务处理并不是困扰过度简化的高速计算的唯一问题。具有实际意义的物理问题往往具有很高的复杂性。造成这种复杂性的因素包括:边界可能是不规则的;内部可能是不均匀的;所需的计算可能取决于各点上变量的状态;不同物理效应的传播速度差异可能很大;收敛速率和收敛性可能在很大程度上取决于后续过程扫过网格时沿不同坐标轴的速度,等等。在并行处理系统中,这些因素中的每一个对基于几何相关处理器的计算机结构影响都非常大。即使对于有着规则矩形边界的问题,也存在一个很有意思的现象,即在  $N$  维空间中,最近邻计算要处理  $3^N$  个不同的几何点。如果还要考虑第二近邻,要处理的几何点会有  $5^N$  个。内部不均匀和边界不规则都会使这个问题复杂化。依赖于变量状态的计算要求处理每个点所消耗的计算时间与计算大区域内所有物理效应总和的时间大致相同。传播速度的差异或变化可能会影响网格点之间的关系。

理想情况下,在计算邻近点对所考虑的点作用时,应考虑它们之前的值,这些值与网格间距成正比,与传播速度成反比。由于时间步长通常保持不变,因此在某些效应中,更快的传播速度意味着与更远的点发生相互作用。在后续过程中沿不同坐标轴扫描网格的常规做法,会引起涉及全部处理器的数据管理问题;然而,且不提改进输入输出调度对几何相关处理器的影响,转置存储所有点造成的影响会更加严重。在考虑并行处理设备时,与对问题进行简化、正规化的抽象操作带来的性能上的影响相比,这些不规律性对实际性能的影响会低约二分

之一到一个数量级。

## 串行负载的比例和问题非规则性的影响的量化结果

为了总结数据管理内务处理和问题非规则性产生的影响,笔者比较了三种不同但硬件数量大致相同的机器结构。机器 A 有 32 个算术执行单元,这些单元由单个指令流控制。机器 B 有流水线算术执行单元,这些单元可在八元素的向量上进行最多三项在时间上重叠的运算。机器 C 同样具有流水线算术执行单元,并且其标量运算的速度与机器 B 容许的向量元素运算的速度相同。三台机器的性能以可并行指令数百分数的函数形式表示在图 1 中。运算的可能范围集中在一个点附近,与该点(可并行比例为 65%)对应的情形是,25% 是数据管理开销,10% 是问题求解过程中必须串行执行的运算。

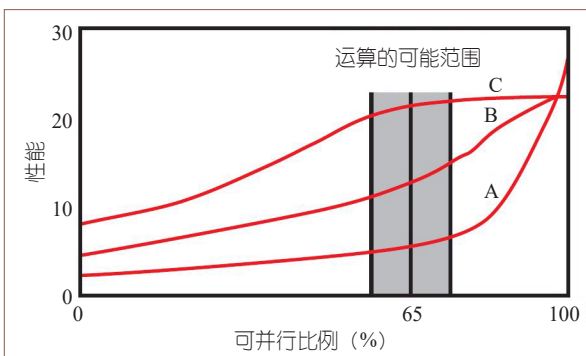


图 1 三种不同的计算机在不同可并行比例下的性能<sup>10</sup> (图题为编译器添加,论文原图无图题)

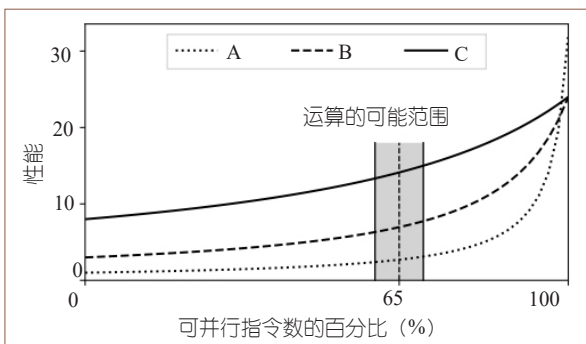


图 2 三种不同计算机在不同可并行比例下的性能 (编译器重新绘制的图)

## 多处理器的性价比较低

奈特 (Knight) 教授曾对历史上的计算机性能与其成本之间的关系做过很透彻的研究。他的数据经过细致的分析, 不仅反映了算术运算的执行时间, 还反映了推荐配置的最低成本。他将内存容量的影响、输入输出的重叠以及特殊功能都考虑进来。在任何工艺水平上, 获得的最佳统计拟合都对应性能与成本的平方之间的正比关系。这一结果非常有力地支持了常被引用的“格罗希定律”(Grosch's Law)<sup>11</sup>。通过这一分析, 我们可以得出, 如果单个系统中可被利用的硬件数量变为原来的两倍, 就有望将性能提升到原来的四倍。唯一的困难在于该如何利用这些额外的硬件。任何时候, 我们都很难预见能有效解决之前串行计算机中瓶颈的方式。因为如果这些瓶颈很容易解决, 就不会成为瓶颈。确实, 历史事例表明, 接二连三的障碍已经被跨越了, 所以我们可以援引亚当·克莱顿·鲍威尔 (Adam Clayton Powell) 牧师的话——“坚持你的信念, 亲爱的!” 或者, 如果我们决定通过将两个处理器与共享内存并排放置来提高性能, 那么硬件数量会变成原来的 2.2 倍左右。另外十分之二的硬件实现用于共享的交叉开关。最终获得的性能约为 1.8。这个数字是基于这个假设得出的: 每个处理器用大约一半的时间占用一半的内存空间。共享系统中产生的内存冲突将使两个操作之一的执行时间延长四分

编译者注:

<sup>1</sup> 这篇文章于 1967 年发表在美国信息处理协会联合会春季计算机会议上, 即 AFIPS (American Federation of Information Processing Societies) spring joint computer conference。注意本文的标题“Validity of the single processor approach to achieving large scale computing capabilities”中的第一个单词“validity”译为“有效性”或“正当性”。现在的主流处理器主要是多核处理器。无论是在处理器层面还是在超级计算机系统层面, 并行计算都成为了一种获得高性能的重要方式。在并行计算领域, 有三大定律,

之一<sup>12</sup>。最终结果就是单个较大处理器 (the single larger processor)<sup>13</sup> 的性价比 (price performance) 下降到了 0.8, 而非提升到 2.0<sup>14</sup>。

## 关联处理器与非关联处理器的比较分析

关联处理器 (associative processor) 与非关联处理器的比较分析远没有那么容易和明显。在一些常规条件下, 有一种相当直接的方法。考虑一个为模式识别设计的关联处理器, 在该处理器中, 一个单元内的决策被转发到其他一组单元中。在关联处理器设计中, 接收单元将有一组源地址, 这些源地址将通过关联技术识别是否要接收当前声明单元的决策。为了实现相应的专用的非关联处理器, 我们可以将接收单元及其源地址视为一条指令, 并将其二进制的决策保存在寄存器中。考虑到我们使用的是薄膜存储器, 一个关联周期会比一个非破坏性的读周期长。通过类比, 可以预计专用非关联处理器花费的存储周期数量大约是关联处理器的四分之一, 而使用时间只有辅助处理器的六分之一。这些数据完全是根据对识别任务进行计算得出的, 并且每个阶段的比例略有不同。我们不打算在这里提出一边倒的主张, 而是希望调查研究如何通过两种方法来实现每个需求。

本文提出的并行计算领域的基本定律“阿姆达尔定律”对应的是第一大定律。文章的标题是“论以单处理器的方式实现大规模计算能力的有效性”, 从标题看, 作者支持单处理器, 隐含地反对并行计算。这篇文章比较短, 写得有点随意, 没有一个公式。据考证, 其中的思想在 20 世纪 50 年代冯·诺伊曼出版的《计算机与人脑》中已有。从这个意义上说, “阿姆达尔定律”的源头至少可以从 1967 年向前推 14 年。

<sup>2</sup> 原文没有分节, 编译者对其进行了分节, 并拟了小标题。

<sup>3</sup> 对计算机体系结构来说，采用单处理器还是多处理器、串行还是并行、通用还是专用，是设计计算机时需要考虑的基本问题。

<sup>4</sup> 这一段给出了本文的核心观点。

<sup>5</sup> 这段原文的首句是“The first characteristic of interest is the fraction of the computational load which is associated with data management housekeeping.”其中的 data management housekeeping 翻译为“数据管理总务”，主要起到数据“划分、分发、协同、控制、同步”的作用，所以“housekeeping”的性质是串行的。

<sup>6</sup> 这段原文的第二句是“This fraction has been very nearly constant for about ten years, and accounts for 40% of the executed instructions in production runs.”这里的“production run”是指应用程序的运行。

<sup>7</sup> 这段原文的第三句是“In an entirely dedicated special purpose environment this might be reduced by a factor of two, but it is highly improbable that it could be reduced by a factor of three.”“reduced by a factor of three”表面意思是“减少三倍”，准确的理解是“减少为原来的三分之一”，而不是减少三分之一。“X is reduced by a factor of Y”等价于“X is divided by Y”。

<sup>8</sup> 这句话是这篇短文中最重要表述之一。

<sup>9</sup> 这段的最后一句可认为是“阿姆达尔定律”的定性表述。后来的研究者根据这个表述，给出了形式化描述：

$$Speedup = \frac{1}{f + \frac{1-f}{n}}$$

这里  $f$  为串行的比例， $n$  为处理器的数量， $Speedup$  为多处理器相对单处理器的加速比。当  $f$  为 20% 时，加速比的上限是 5；当  $f$  为 13.3% 时，加速比的上限约为 8。但后者很难达到，所以作者说“即使数据管理任务在单个处理器中完成，仅开销一项就将吞吐量的上限定格在串行处理速度的 5~7 倍。”

<sup>10</sup> 图 1 中纵轴的“性能”是指“加速比”，参照的基准对象 (baseline) 是没有单指令多数据流 (SIMD) 技术 (即只有一个算术执行单元)、没有流

水线的机器。

机器 A 没有流水线，但对可并行部分支持 SIMD 技术 (宽度为 32)，加速比曲线为

$$S_A = \frac{1}{\frac{1-x}{1} + \frac{x}{32}}$$

当  $x=0.65$  时，

$$S_A = \frac{1}{\frac{0.35}{1} + \frac{0.65}{32}} = 2.7$$

机器 B 和 C 是机器 A 的改进。

机器 B：对全部负载支持流水化 (3 级流水)，对可并行部分支持 SIMD 技术 (宽度为 8) 和流水化 (3 级流水)，加速比曲线为

$$S_B = \frac{1}{\frac{1-x}{3} + \frac{x}{8 \times 3}}$$

当  $x=0.65$  时，

$$S_B = \frac{1}{\frac{0.35}{3} + \frac{0.65}{8 \times 3}} = 6.9$$

机器 C：对全部负载支持流水化 (3 级流水)，启动标量运算的速度与机器 B 容许的向量元素运算的速度相同，对可并行部分支持 SIMD 技术 (宽度为 8) 和流水化 (3 级流水)，加速比曲线为

$$S_C = \frac{1}{\frac{1-x}{8} + \frac{x}{8 \times 3}}$$

当  $x=0.65$  时，

$$S_C = \frac{1}{\frac{0.35}{8} + \frac{0.65}{8 \times 3}} = 14.1$$

当可并行比例为 0% 时，程序中的指令只能串行执行。机器 A 的 SIMD 部件利用率仅为 1/32，性能与基准机器一样 (即加速比为 1)；机器 B 有 3 级流水，能够支持 3 条指令重叠执行，加速比为 3；机器 C 的标量运算速度为原来的 8 倍，加速比为 8。

当可并行比例为 100% 时，程序中的指令都可以并行执行。机器 A 支持的 SIMD 宽度为 32，加速比为 32；机器 B 支持的 SIMD 宽度为 8，且有 3 级

流水，加速比为24；机器C拥有与机器B相同的流水化向量执行单元，加速比也为24。

需要注意的是，原文中的图（即图1）存在一些不影响其结论的错误（图2是编译者对原图的修正）：虽然机器B和C在可并行比例为0%和100%时的性能加速比值是正确的，但中间的数值却不尽准确。机器B和C的加速比函数应是凹函数（即二阶导数大于0），加速比曲线的增长与机器A一样是逐渐加快，而不是先快后慢。虽然这导致机器B和C在并行比例为65%时的数值不准确，但并不影响它们的相对大小，即性能加速比仍然是 $C>B>A$ ，不影响基本结论。

<sup>11</sup> 这一段提到格罗希定律，这个定律比摩尔定律还要早。1953年，埃布·格罗希（Herb Grosch）提出一个观察：计算机性能随着成本的平方而增加，也就是说，如果计算机A的成本是计算机B的两倍，那么计算机A的速度应该是计算机B的四倍。

<sup>12</sup> 为什么“共享系统中产生的内存冲突将使两个操作之一的执行时间延长四分之一”？解答：在没有访存冲突时，假设每个操作的时间为 $T$ 。当两个操作发生访存冲突时（冲突发生的概率为 $1/2$ ），其中一个操作要被延后 $T/2$ ，所以共享系统中产生的内存冲突将使两个操作之一的执行时间延长四分之一（即 $1/2 \times T/2$ ）。被延后的那个操作的执行时间变为 $5T/4$ ，对应的处理器性能变为原来的 $4/5$ ；另一个处理器性能不变。所以整个系统“最终获得的性能约为1.8”。1.8与2.2之比约为0.8，所以性价比下降到了0.8。

<sup>13</sup> 这里单个较大处理器就是指共享内存的多“核”处理器（当然这里多核未必在同一芯片上）。

<sup>14</sup> 上面这段话的逻辑是什么？或者说，论点和论据是什么，论证是如何展开的？解答：作者说“很难预见能有效解决之前串行计算机中瓶颈的方式。因为如果这些瓶颈很容易解决，就不会成为瓶颈。”这里有个问题：瓶颈是否可以被逐渐解决呢？因此作者退一步说，“确实，历史事例表明，接二连三的障碍已经被跨越了”，但他很快就以共享存储系统的例子说明瓶颈不容易解决。要注意到，作者除了关

注性能，还关注性价比（在实际生活中，很多人往往只看加速比），共享内存的双核处理器的性能确实是增加了，变为单核时的1.8倍，但性价比变为单核时的0.8。

真理往往就隐藏在这些曲曲折折的论证、重要的概念思辨之中。在50多年后的今天来看，阿姆达尔定律是对还是错？答案是，它既对了，又错了；或者说，它既有对的一面，又有错的一面。对的一面是：作者看到了并行计算有很多不利的因素；错的一面是：作者对持续跨越障碍过于悲观，把不确定性当成了不可能性。但人类可以通过智慧和努力不断地改进算法，具体问题具体分析（通过定制），不断地创造奇迹（抓住了本来就客观存在但未必显而易见的优化机遇，做成了原先看起来很难做成的事情）。错误的一面后来被古斯塔夫森定律（Gustafson's Law）和孙-倪定律（Sun-Ni's Law）等理论修正。



周嘉鹏

中国科学院计算技术研究所研究生。主要研究方向为计算机体系结构、存储系统设计及性能优化。



张昶璇

曼彻斯特大学计算机系研究生。主要研究方向为信息系统、统计分布。



刘宇航

CCF高级会员，CCCF专栏编委，CCF职业伦理与学术道德委员会常务委员。中国科学院计算技术研究所副研究员。主要研究方向为计算机体系结构、高性能计算、存储系统、智能并发系统。  
liuyuhang@ict.ac.cn

（本文责任编辑：翟季冬）