计算机体系结构：量化研究方法

◎ John L. Hennessy; David A. Patterson

附录翻译稿

目录

附录 M

历史视角和参考

如果……历史……教会了我们什么，那就是人类在追求知识与进步的道路上，意志坚定，不可阻挡。

John F. Kennedy

在莱斯大学的演讲（1962 年）

那些不能铭记过去的人，注定要重蹈覆辙。

George Santayana

《理性生活》（1905 年），第二卷，第三章

M.1 内容介绍

本附录介绍了各章节中一些关键思想的历史背景。我们将通过一系列计算机机器的发展或对重要项目的介绍来追溯某一思想的演变历程。如果你希望深入了解某一思想或机器的最初发展过程，或希望进行进一步阅读，每个部分末尾均提供了参考文献。

第 M.2 节从数字计算机的发明开始，对应第 1 章内容。第 M.3 节讲述存储器层次结构，对应第 2 章及附录 B。第 M.4 节讨论指令集架构，涵盖附录 A、J 和 K。第 M.5 节介绍流水线技术和指令级并行，对应第 3 章以及附录 C 和 H。第 M.6 节探讨向量架构、SIMD 和 GPU 架构中的数据级并行，对应第 4 章。第 M.7 节关于多处理器与并行编程，涵盖第 5 章以及附录 F、G 和 I。第 M.8 节讲述集群系统的发展，对应第 6 章。 最后，第 M.9 节关于输入/输出（I/O），对应附录 D。

M.2 计算机的早期发展

在本历史章节中，我们将探讨数字计算机的早期发展以及性能测量方法的演进。

第一台通用电子计算机

宾夕法尼亚大学摩尔工程学院的 J. Presper Eckert 和 John Mauchly 研制了世界上第一台完全投入运行的电子通用计算机。这台名为 ENIAC（电子数值积分计算器和计算机，Electronic Numerical Integrator and Calculator）的机器由美国陆军资助建造，于第二次世界大战期间建成并投入使用，但直到 1946 年才对外公开。ENIAC 最初被用于计算火炮射击表。这台机器体积庞大——长达 100 英尺，高 8.5 英尺，宽数英尺。其 20 个十位数寄存器中的每一个都有 2 英尺长，整台机器共使用了约 18,000 个真空管。

尽管 ENIAC 的体积比当今普通计算机大三个数量级，但其运行速度却慢了五个数量级以上，一次加法运算需要 200 微秒。ENIAC 具备条件跳转功能，并且可以编程，这使其明显区别于早期的计算器。编程需要通过手动插接电缆和设置开关来完成，耗时从半小时到一整天不等。数据通过打孔卡片输入。ENIAC 的主要局限在于存储容量小以及编程过程繁琐。

1944 年，von Neumann 开始关注 ENIAC 项目。该项目团队希望改进程序输入的方式，并讨论了将程序以数字形式存储的设想。其中 von Neumann 帮助系统化地整理了这些思想，并撰写了一份备忘录，提出了一种名为 EDVAC（电子离散变量自动计算机，Electronic Discrete Variable Automatic Computer）的存储程序计算机设计方案。Herman Goldstine 分发了这份备忘录，并将 von Neumann 的名字单独署在上面，这让 Eckert 和 Mauchly 感到十分不满，因为他们的名字被忽略了。这份备忘录后来成为“冯·诺依曼计算机”这一常用术语的来源。计算机领域的多位早期发明者认为，这一名称过分突出了 von Neumann 的贡献——他主要是对这些思想进行了概括和撰写，而对真正从事机器研制工作的工程师

Eckert 和 Mauchly 的贡献则强调不足。与大多数历史学家一样，本书作者（2000 年 IEEE 冯·诺依曼奖章获得者）认为，这三位人物在存储程序计算机的发展过程中都发挥了关键作用。而 von Neumann 在系统阐述这些思想、进行理论推广以及思考编程相关问题方面起到了至关重要的作用，使得这些理念得以传播到更广泛的受众中。

1946 年，剑桥大学的 Maurice Wilkes 访问了摩尔学院，参加了关于电子计算机发展的一系列讲座的后半部分。回到剑桥后，Wilkes 决定着手启动一个项目，建造一台名为 EDSAC（电子延迟存储自动计算机，Electronic Delay Storage Automatic Calculator）的存储程序计算机。（EDSAC 使用水银延迟线作为其存储器，因此其名称中带有“延迟存储”一词。）EDSAC 于 1949 年投入使用，成为世界上第一台全规模、可操作的存储程序计算机[Wilkes, Wheeler, and Gill 1951; Wilkes 1985, 1995]。（在此之前，曼彻斯特大学于 1948 年建造过一个名为 Mark I 的小型原型机，可能可以被称作第一台可操作的存储程序机器。）EDSAC 采用基于累加器的架构，这种指令集架构风格一直流行到 1970 年代初。（附录 A 以对 EDSAC 指令集的简要总结开始叙述）

1947 年，Mauchly 花时间帮助创立了计算机协会（Association for Computing Machinery，ACM）。他担任了 ACM 的第一任副主席和第二任主席。同年，Eckert 和 Mauchly 申请了电子计算机的专利。摩尔学院的院长要求将专利权交给大学，这可能促使 Eckert 和 Mauchly 决定离开。他们的离开严重影响了 EDVAC 项目，导致该项目制造的机器直到 1952 年才投入使用。

1946 年，Goldstine 离开摩尔学院，加入 von Neumann 在普林斯顿高等研究院的团队。他们与 Arthur Burks 共同发表了一份基于 1944 年备忘录的报告 [Burks, Goldstine, and von Neumann 1946]。这份报告促成了 Julian Bigelow 牵头在普林斯顿高等研究院建造 IAS 计算机。IAS 计算机共有 1024 个 40 位字的存储容量，运行速度约为 ENIAC 的 10 倍。该团队深入思考了这台计算机的应用前景，发表了一系列报告，并积极接待访问学者。这些报告和来访者极大地推动了多台新型计算机的研发，其中包括 IBM 的第一台计算机 701，其设计正是基于 IAS 计算机。Burks、Goldstine 和 von Neumann 的这篇论文在当时堪称非凡之作。即使今天读来，也很难相信这篇具有里程碑意义的论文竟写于 50 多年前，因为其中讨论的大多数计算机体系结构概念至今仍在现代计算机中广泛应用（参见第 2 章开头的引文）。

与 ENIAC 同一时期，Howard Aiken 在哈佛大学设计了一台机电式计算机，名为 Mark-I。Mark-I 由 IBM 的一支工程师团队建造。此后，Aiken 又相继推出了继电器计算机 Mark-II，以及两台真空管计算机 Mark-III 和 Mark-IV。Mark-III 和 Mark-IV 是在第一台存储程序计算机问世之后建造的。由于它们将指令和数据分别存储在不同的存储器中，因此被存储程序计算机的支持者视为倒退之举。这种结构后来被称为“哈佛架构”。尽管与原始含义已经不同，但如今该术语被用于描述那些主存统一，但在缓存层面将指令和数据分开的计算机。

1947 年，麻省理工学院（Massachusetts Institute of Technology）启动了“旋风”（Whirlwind）项目 [Redmond and Smith 1980]，旨在实现实时雷达信号处理

的应用。尽管该项目催生了多项发明，但其最重大的创新是发明了磁芯存储器——这是第一种可靠且成本低廉的存储技术。Whirlwind 计算机配备了 2048 个 16 位字的磁芯存储器。磁芯存储器在此后近 30 年中一直作为主流的内存技术。

重要的专用计算机

第二次世界大战期间，英国和美国的主要计算工作都集中在专用的密码破译计算机上。英国的工作目标是破解用德国恩尼格码（Enigma）编码机加密的消息。这项工作在一个名为 Bletchley Park 的地方进行，催生了两台重要的机器。第一台是由 Alan Turing 构思的机电机械装置，称为“BOMB”[see Good in Metropolis, Howlett, and Rota 1980]。第二台是由 Newman 和 Flowers 设计的体积更大、电子化的机器，称为“COLOSSUS”[see Randall in Metropolis, Howlett, and Rota 1980]。这些都是高度专业化的密码分析机器，尤其是能够读取发送给德国潜艇（U-boat）的加密信息，从而在战争中发挥了关键作用。Bletchley Park 的工作高度机密（事实上，有些至今仍属机密），因此它对 ENIAC、EDSAC 及其他计算机的发展所产生的直接影响难以追溯，但无疑在推动技术进步和理解其中的相关问题方面产生了间接影响。

美国也开展了类似的专用密码分析计算机工作。这一努力的最直接后继者是工程研究协会（Engineering Research Associates，简称 ERA）[see Thomash in Metropolis, Howlett, and Rota 1980]，该公司战后成立，试图将关键思想商业化。ERA 制造了几台机器，售予秘密政府机构，最终被 Sperry-Rand 公司收购，而 Sperry-Rand 公司此前已收购了 Eckert-Mauchly 计算机公司（Eckert Mauchly Computer Corporation）。

另一组值得称道的早期机器是 Konrad Zuse 在德国于 1930 年代末和 1940 年代初制造的一系列专用机器[see Bauer and Zuse in Metropolis, Howlett, and Rota 1980]。除了制造出可运行的机器外，Zuse 还是第一个实现浮点运算的人，尽管 von Neumann 声称这是不必要的！他的早期机器采用了机械存储，比当时其他机电解决方案更小巧。他的最后一台机器是机电式的，但因战争原因未能完成。

电子计算机发展的另一个重要早期贡献者是 John Atanasoff，他在 1940 年代初制造了一台小型电子计算机[Atanasoff 1940]。这台机器在爱荷华州立大学（Iowa State University）设计，是一台专用计算机（ABC，Atanasoff Berry Computer），但从未完全投入使用。Mauchly 在制造 ENIAC 之前曾短暂拜访过阿塔纳索夫，阿塔纳索夫的一些想法（例如使用二进制表示法）很可能影响了 Mauchly。ABC 机器的存在、ENIAC 专利申请的延迟（因工作机密，专利只能在战后申请）以及 von Neumann 关于 EDVAC 的论文的传播，成为推翻 Eckert-Mauchly 专利的依据[Larson 1973]。尽管关于 Atanasoff 的作用仍有争议，但 Eckert 和 Mauchly 通常被认为是第一台可工作的通用电子计算机的制造者[Stern 1980]。然而，Atanasoff 展示了后来计算机中包含的若干重要创新。他的工作值得高度评价，且他可能应被公认为世界上第一台专用电子计算机的发明者，并且可能影响了 Eckert 和 Mauchly。

商业研发

1947 年 12 月，Eckert 和 Mauchly 成立了 Eckert-Mauchly 计算机公司。他们的第一台机器 BINAC 是为 Northrop 制造的，并于 1949 年 8 月完成。经历了一些财务困难后，Eckert-Mauchly 计算机公司被 Remington-Rand 收购，后者改名为 Sperry-Rand。Sperry-Rand 将其收购的 Eckert-Mauchly、ERA 以及其自身的制表业务合并，成立了一个专门的计算机部门，称为 UNIVAC。UNIVAC 于 1951 年 6 月交付了其第一台计算机 UNIVAC I。UNIVAC I 售价为 25 万美元，是第一台成功的商业计算机——共制造了 48 台！如今，这台早期机器以及许多其他吸引人的计算机历史文物可以在加利福尼亚州山景城（Mountain View）的计算机历史博物馆看到。其他可以参观早期计算机系统的地方包括慕尼黑的德意志博物馆（Deutsches Museum）和华盛顿特区的史密森学会（Smithsonian Institution），以及众多在线虚拟博物馆。

IBM 早期从事穿孔卡和办公自动化业务，直到 1950 年才开始制造计算机。第一台 IBM 计算机 IBM 701 基于冯·诺依曼的 IAS 机器，于 1952 年出货，最终售出 19 台[see Hurd in Metropolis, Howlett, and Rota 1980]。20 世纪 50 年代初，许多人对计算机的未来持悲观态度，认为这些“高度专业化”机器的市场和机会非常有限。尽管如此，IBM 还是迅速成为了最成功的计算机公司。他们对可靠性的追求以及以客户和市场为导向的策略是成功的关键。虽然 701 和 702 只取得了一些小进展，IBM 随后推出的 650、704 和 705（分别于 1954 年和 1955 年交付）取得了重大成功，销量从 132 台到 1800 台不等。

多位先驱撰写了几本描述计算机早期发展的书籍[Goldstine 1972；Wilkes 1985, 1995]，以及 Metropolis、Howlett 和 Rota[1980]编辑的作品，该书汇集了早期先驱们的回忆录。此外，还有许多独立的历史著作，通常围绕相关人物展开[Slater 1987]，以及一本专门致力于计算机历史的期刊——《计算机历史年鉴》（Annals of the History of Computing）。

定量性能指标的发展：成功与失败

在计算机发展的最早期，设计者就开始设定性能目标——ENIAC 要比哈佛 Mark-I 快 1000 倍，IBM Stretch（7030）则要比当时最快的机器快 100 倍。然而，届时尚不清楚这些性能应如何衡量。回顾过去的岁月，一个贯穿始终的主题是，每一代计算机都会使上一代的性能评估方法变得过时。

最初的性能衡量标准是完成单个操作（如加法）所需的时间。由于大多数指令执行时间相同，测量其中一个操作的时间便能反映其他操作的时间。然而，随着机器中指令执行时间的多样化，单一操作的时间不再适用于比较。为了考虑这些差异，通过测量计算机在多个程序中指令的相对频率，计算出指令组合。Gibson 组合[Gibson 1970]是早期流行的指令组合。将每条指令的执行时间乘以其在组合中的权重，用户便能得到平均指令执行时间。（如果以时钟周期计，平均指令执行时间即为平均每条指令所需的时钟周期数。）由于早期计算机系统指令集相似，这种方法比单纯的加法时间更准确。从平均指令执行时间出发，计算 MIPS（Million Instructions Per Second，百万条指令每秒）就是顺理成章的了（正如我们所见，两者是互为倒数的关系）。MIPS 的优点在于普通人容易理解。

随着 CPU 变得更加复杂，依赖于存储层次结构和流水线技术，每条指令不再有唯一的执行时间；因此，不能再通过指令组合和手册计算 MIPS。下一步是使用内核程序和合成程序进行基准测试。Curnow 和 Wichmann [1976] 通过测量用 Algol 60 编写的科学程序，创建了 Whetstone 合成程序。该程序被转换为 FORTRAN，并被广泛用于表征科学程序的性能。McMahon [1986] 和劳伦斯利弗莫尔国家实验室（Lawrence Livermore Laboratory）的研究人员则开展了类似的工作，开发了 Livermore FORTRAN Kernels，旨在为超级计算机建立基准测试。当然，这些内核程序由真实程序中的循环组成。

随着人们逐渐意识到使用 MIPS 来比较具有不同指令集的架构是行不通的，便产生了相对 MIPS 的概念。1977 年，当 VAX-11/780 准备发布时，DEC 运行了一些在 IBM 370/158 上也运行过的小型基准测试。IBM 市场部门称 370/158 为 1 MIPS 计算机，由于这些程序在两台机器上的运行速度相同，DEC 市场部门也将 VAX-11/780 称为 1 MIPS 计算机。机器 M 的相对 MIPS 是基于某个参考机器定义的，公式为：

$$\text{MIPS}_{\text{M}} = \frac{\text{Performance}_{\text{M}}}{\text{Performance}_{\text{reference}}} \times \text{MIPS}_{\text{reference}}$$

VAX-11/780 的流行使其成为相对 MIPS 的一个经典参考机器，尤其是因为对于 VAX11/780 这样的一个 1 MIPS 的计算机来说，相对 MIPS 的计算非常简单：如果一台机器的速度是 VAX-11/780 的五倍，那么在该基准测试中，它的评分就是 5 相对 MIPS。1 MIPS 的评级在四年内无人质疑，直到 DEC 的 Joel Emer 在时间共享负载下测量了 VAX-11/780。他发现 VAX-11/780 的原生 MIPS 实际上是 0.5。因此，后续在某些基准测试中达到原生 3 MIPS 的 VAX 机型被称为 6 MIPS 机器，因为它们的速度是 VAX-11/780 的六倍。到了 1980 年代初，MIPS 一词几乎普遍被用来表示相对 MIPS。

1970 年代和 1980 年代是着超级计算机产业的增长时期，该产业以在浮点密集型程序上的高性能为特征。平均指令时间和 MIPS 显然不适合衡量这一行业的性能，因此人们发明了 MFLOPS（每秒百万次浮点运算），它有效地表示了基准测试执行时间的倒数。不幸的是，客户很快就忘记了用于评级的具体程序，市场部门于是开始在超级计算机性能竞赛中引用峰值 MFLOPS。

SPEC（System Performance and Evaluation Cooperative）成立于 1980 年代末，旨在改善基准测试的现状，并建立更有效的比较基础。该组织最初专注于 UNIX 市场中的工作站和服务器，这些领域至今仍是其基准测试的主要关注点。SPEC 基准测试的首次发布，现在被称为 SPEC89，是在采用更现实基准测试方面的重大进步。近二十年后，SPEC2006 仍然主导着处理器基准测试领域。

参考文献

Amdahl, G. M. [1967]. “Validity of the single processor approach to achieving large scale computing capabilities,” Proc. AFIPS Spring Joint Computer Conf., April 18–20, 1967, Atlantic City, N.J., 483–485.

Atanasoff, J. V. [1940]. “Computing machine for the solution of large systems of linear equations,” Internal Report, Iowa State University, Ames.

Azizi, O., Mahesri, A., Lee, B. C., Patel, S. J., & Horowitz, M. [2010]. Energy performance tradeoffs in processor architecture and circuit design: a marginal cost analysis. Proc. International Symposium on Computer Architecture, 26-36.

Bell, C. G. [1984]. “The mini and micro industries,” IEEE Computer 17:10 (October), 14–30.

Bell, C. G., J. C. Mudge, and J. E. McNamara [1978]. A DEC View of Computer Engineering, Digital Press, Bedford, Mass.

Burks, A. W., H. H. Goldstine, and J. von Neumann [1946]. “Preliminary discussion of the logical design of an electronic computing instrument,” Report to the U.S. Army Ordnance Department, p. 1; also appears in Papers of John von Neumann, W. Aspray and A. Burks, eds., MIT Press, Cambridge, Mass., and Tomash Publishers, Los Angeles, Calif., 1987, 97–146.

Curnow, H. J., and B. A. Wichmann [1976]. “A synthetic benchmark,” The Computer J. 19:1, 43–49. Dally, William J., “High Performance Hardware for Machine Learning,” Cadence Embedded Neural Network Summit, February 9, 2016. http://ip.cadence.com/uploads/presentations/1000AM_Dally_Cadence_ENN.pdf

Flemming, P. J., and J. J. Wallace [1986]. “How not to lie with statistics: The correct way to summarize benchmarks results,” Communications of the ACM 29:3 (March), 218–221.

Fuller, S. H., and W. E. Burr [1977]. “Measurement and evaluation of alternative computer architectures,” Computer 10:10 (October), 24–35.

Gibson, J. C. [1970]. “The Gibson mix,” Rep. TR. 00.2043, IBM Systems Development Division, Poughkeepsie, N.Y. (research done in 1959).

Goldstine, H. H. [1972]. The Computer: From Pascal to von Neumann, Princeton University Press, Princeton, N.J.

Gray, J., and C. van Ingen [2005]. Empirical Measurements of Disk Failure Rates and Error Rates, MSR-TR-2005-166, Microsoft Research, Redmond, Wash.

Jain, R. [1991]. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley, New York.

Kembel, R. [2000]. “Fibre Channel: A comprehensive introduction,” Internet Week (April).

Larson, E. R. [1973]. “Findings of fact, conclusions of law, and order for judgment,” File No. 4-67, Civ. 138, Honeywell v. Sperry-Rand and Illinois Scientific Development, U.S. District Court for the State of Minnesota, Fourth Division (October 19).

Lubeck, O., J. Moore, and R. Mendez [1985]. “A benchmark comparison of three supercomputers: Fujitsu VP-200, Hitachi S810/20, and Cray X-MP/2,” Computer 18:12 (December), 10–24.

Landstrom, B. [2014]. “The Cost Of Downtime,” http://www.interxion.com/blogs/2014/07/the-cost-of-downtime/

McMahon, F. M. [1986]. The Livermore FORTRAN Kernels: A Computer Test of Numerical Performance Range, Tech. Rep. UCRL-55745, Lawrence Livermore National Laboratory, University of California, Livermore.

Metropolis, N., J. Howlett, and G. C. Rota, eds. [1980]. A History of Computing in the Twentieth Century, Academic Press, New York.

Mukherjee S. S., C. Weaver, J. S. Emer, S. K. Reinhardt, and T. M. Austin [2003]. "Measuring architectural vulnerability factors," IEEE Micro 23:6, 70–75.

Oliker, L., A. Canning, J.Carter, J. Shalf, and S. Ethier [2004]. "Scientific computations on modern parallel vector systems," Proc. ACM/IEEE Conf. on Supercomputing, November 6–12, 2004, Pittsburgh, Penn., 10.

Patterson, D. [2004]. "Latency lags bandwidth," Communications of the ACM 47:10 (October), 71–75.

Redmond, K. C., and T. M. Smith [1980]. Project Whirlwind—The History of a Pioneer Computer, Digital Press, Boston.

Shurkin, J. [1984]. Engines of the Mind: A History of the Computer, W. W. Norton, New York.

Slater, R. [1987]. Portraits in Silicon, MIT Press, Cambridge, Mass.

Smith, J. E. [1988]. "Characterizing computer performance with a single number,"Communications of the ACM 31:10 (October), 1202–1206.

SPEC. [1989]. SPEC Benchmark Suite Release 1.0 (October 2).

SPEC. [1994]. SPEC Newsletter (June).

Stern, N. [1980]. "Who invented the first electronic digital computer?" Annals of the History of Computing 2:4 (October), 375–376.

Touma, W. R. [1993]. The Dynamics of the Computer Industry: Modeling the Supply of Workstations and Their Components, Kluwer Academic, Boston.

Weicker, R. P. [1984]. "Dhrystone: A synthetic systems programming benchmark," Communications of the ACM 27:10 (October), 1013–1030.

Wilkes, M. V. [1985]. Memoirs of a Computer Pioneer, MIT Press, Cambridge, Mass.

Wilkes, M. V. [1995]. Computing Perspectives, Morgan Kaufmann, San Francisco.

Wilkes, M. V., D. J. Wheeler, and S. Gill [1951]. The Preparation of Programs for an Electronic Digital Computer, Addison-Wesley, Cambridge, Mass.

M.3 存储层次结构与内存保护的发展

虽然计算机先驱们很早就认识到存储层次结构的必要性并创造了这一术语，但两级存储自动管理的概念最早却是由 Kilburn 等人在 1962 年才提出，并在曼彻斯特大学的 Atlas 计算机上得到了验证。这台计算机问世于 IBM 360 发布的前一年。尽管 IBM 计划在下一代 System/370 中引入该技术，但 1970 年的操作系统 TSS 未能应对这一挑战。虚拟内存于 1972 年在 370 系列中正式推出，正是在这款计算机上首次提出了"翻译后备缓冲器"（Translation Lookaside Buffer，TLB）这一术语[Case and Padegs, 1978]。现今，几乎所有计算机都配备了虚拟内存，唯有少数超级计算机、嵌入式处理器和较老的个人计算机例外。

Atlas 和 IBM 360 均在页（page）级别提供了存储保护，而 GE 645 是首个实现分页分段（paged segmentation）系统。早期的 Burroughs 计算机则采用分段（segmentation）实现虚拟内存，这与 Intel 8086 的分段地址方案类似。80286 作为首款具备附录 C 中描述的保护机制的 80x86 处理器，其设计灵感来源于运行于 GE 645 上的 Multics 保护软件。随着时间的推移，计算机发展出了更为复杂的机制。其中最复杂的机制是“capabilities”，这一机制在 20 世纪 70 年代末和 80 年代初引起了极大的关注 [Fabry 1974; Wulf、Levin and Harbison 1981]。Wilkes [1982] 是早期研究能力机制的人员之一，他这样说道：

任何曾经关注过上述（capabilities 系统）这种实现的人，或者尝试向他人解释过这种系统的人，都会觉得这种系统的复杂性将无法掌控。尤其令人失望的是，capabilities 作为可以自由传递的‘票据’这一吸引人的想法已经丧失了。

与传统计算机系统相比，提供一个保护域小且频繁变化的系统必然会付出代价。这种代价表现为额外的硬件需求、运行速度的降低以及内存占用的增加。目前尚不清楚通过采用能力机制，是否能够将这些代价降低到合理的范围内。[第 112 页]

今天，尽管对内存保护和安全的关注日益增长，操作系统和计算机体系结构领域对能力机制（capabilities）的关注却很少了。

Bell 和 Strecker [1976] 回顾了 PDP-11，认为其唯一且难以弥补的体系结构错误是地址空间过小。在 PDP-11 诞生之时，磁芯存储器的容量增长非常缓慢。此外，来自 100 多家迷你计算机公司的激烈竞争意味着，如果每个 32 位地址地址都必须经过 16 位数据通路两次，DEC 的产品可能无法在成本上具有竞争力，因此设计者决定只比 PDP-11 的前代机型增加 4 个位的地址位数。

IBM 360 的设计者们意识到了地址大小的重要性，并计划将体系结构扩展到 32 位地址。然而，在 1964 年，IBM 360 实际上只使用了 24 位地址，因为当时低端的 360 型号如果使用更大的地址位数，性能会更差。不幸的是，设计者没有将这一计划告知软件开发人员，程序员们将额外信息存储在上层 8 个位的“未使用”地址位中，这阻碍了地址扩展的努力。（20 年后，Apple 在 Motorola 68000 的 24 位地址设计上犯了类似的错误，当时为了支持后续使用完整 32 位虚拟地址的 68000 处理器，Macintosh 需要通过某种程序来判定哪些程序是“32 位兼容”的。）从那以后，几乎所有计算机都会检查这些未使用的位是否保持未使用状态，如果这些位的值不正确则会触发异常。

如前文所述，系统级别的虚拟机最早由 IBM 在其对虚拟内存的研究中开创。IBM 首台具有虚拟内存的计算机是 1967 年推出的 IBM 360/67。IBM 的研究人员编写了程序 CP-67，该程序创造了多个独立的 360 计算机的幻象。随后，他们又编写了一个名为 CMS 的交互式单用户操作系统，运行在这些虚拟机上。CP-67 促成了产品 VM/370 的诞生，如今 IBM 为其大型机销售 z/VM 操作系统 [Meyer and Seawright 1970; Van Vleck 2005]。

在 Atlas 机器的论文发表几年后，Wilkes 于 1965 年发表了第一篇描述高速缓存（cache）概念的论文：

- 译者注：这里的 capabilities 指的是一种访问控制机制，用来管理和限制程序对计算机内存的访问。Wilkes 在这里表示的是虽然 capabilities 机制的概念很吸引人（就像“票据”一样可以自由传递），但实际实现起来非常复杂，导致系统变得难以管理和理解。

- Wilks 的原话是：*Anyone who has been concerned with an implementation of the type just described [capability system], or has tried to explain one to others, is likely to feel that complexity has got out of hand. It is particularly disappointing that the attractive idea of capabilities being tickets that can be freely handed around has become lost…*

  Compared with a conventional computer system, there will inevitably be a cost to be met in providing a system in which the domains of protection are small and frequently changed. This cost will manifest itself in terms of additional hardware, decreased runtime speed, and increased memory occupancy. It is at present an open question whether, by adoption of the capability approach, the cost can be reduced to reasonable proportions. [p. 112]

讨论了使用一个容量约为32,000字的快速磁芯存储器作为容量约为一百万字的较慢磁芯存储器的辅助存储器,在实际情况下,使得有效访问时间更接近快速存储器而非慢速存储器。(第 270 页)

Wilks 的原话是：*The use is discussed of a fast core memory of, say, 32,000 words as slave to a slower core memory of, say, one million words in such a way that in practical cases the effective access time is nearer that of the fast memory than that of the slow memory. [p. 270]*

这篇两页的论文描述了直接映射缓存(direct-mapped cache)。虽然这是关于缓存的首次公开发表，但第一个实现缓存的系统很可能是剑桥大学建造的直接映射指令缓存，它基于当时最快的隧道二极管存储器。Wilkes 表示，G. Scarott 提出了缓存存储器的想法。

在该论文发表后，IBM 启动了一个项目，最终推出了首台带有缓存的商用计算机——IBM 360/85 [Liptay 1968]。Gibson [1967] 描述了如何通过内存访问量和未命中率来测量程序行为,并展示了不同程序之间未命中率的差异。Gibson 使用了 20 个程序样本(每个程序有 300 万个内存访问!),并依靠平均内存访问时间来比较有缓存和无缓存的系统。这一方法已有 40 多年历史,直到 1990 年代初，很多人仍然只使用未命中率作为性能指标。

Conti、Gibson 和 Pitkowsky [1968] 描述了 360/85 的性能表现。论文中提到，尽管 360/85 的时钟周期较慢(80 纳秒对 60 纳秒)、内存交错较少(4 路对 16 路)且主存速度较慢(1.04 微秒对 0.75 微秒)，但 360/91 仅在 11 个测试程序中的 3 个程序上性能优于 360/85。这篇论文也是首次使用“cache”一词的文献。

其他学者很快扩展了缓存相关的研究文献。Strecker [1976] 发表了首篇比较缓存设计的论文，研究了针对 PDP-11 的缓存设计。Smith [1982] 随后发表了一篇详尽的综述论文，首次使用了空间局部性(spatial locality)和时间局部性(temporal locality)这两个术语,该论文成为许多计算机架构师的重要参考文献。

虽然大多数研究依赖于模拟，Clark [1983] 则使用硬件监控器记录了 VAX-11/780 在数天内的缓存未命中情况。Clark 和 Emer [1985] 后来比较了模拟结果与硬件测量数据在地址转换方面的差异。

Hill [1987] 提出了附录 B 中用于解释缓存未命中的“三 C”模型。Jouppi [1998] 回顾时表示，正是 Hill 的三 C 模型直接促成了他发明牺牲缓存(victim cache)，以利用更快的直接映射缓存，同时避免大部分冲突未命中的开销。Sugumar 和 Abraham [1993] 认为，三 C 模型的基准缓存应采用最优替换策略;这将消除基于最近最少使用(LRU)算法的未命中分类中的异常情况,并允许将冲突未命中细分为由映射方式引起的和由非最优替换算法引起的两类。

关于非阻塞缓存(nonblocking caches)的早期论文之一是 Kroft [1981]。Kroft [1998] 后来解释说，他是第一位在 CDC 设计带缓存的计算机的人，当他将旧概念应用于新机制时,提出了让他的双端口缓存(two-ported cache)在发生未命中时继续为其他访问提供服务的想法。

Baer 和 Wang [1988] 进行了最早对多级缓存包含性质(multilevel inclusion property)的研究之一。随后，Wang、Baer 和 Levy [1989] 发表了一篇关于多级缓存性能评估的早期论文。之后，Jouppi 和 Wilton [1994] 提出了针对片上多级缓存的多级排除(multilevel exclusion)策略。

除了牺牲缓存,Jouppi [1990] 还研究了通过流缓冲区(streaming buffers)进行预取(prefetching)。Farkas、Jouppi 和 Chow [1995] 扩展了这一工作,提出了适

用于非阻塞加载和顺序处理器的推测执行的流缓冲区，后来 Farkas 等人 [1997] 证明，虽然乱序处理器（out-of-order processors）对不可预测的延迟容忍度更高，但仍然能从中受益，并且他们还细化了流缓冲区对内存带宽的需求。

1990 年代的编程语言和操作系统的体系结构支持国际研讨会（ASPLOS）和国际计算机体系结构研讨会（ISCA）上充斥着大量关于缓存的论文。（事实上，有人戏称 ISCA 实际上是“国际缓存体系结构研讨会”的缩写。）

第 2 章依赖于 Cantin 和 Hill [2001] 收集的 SPEC2000 基准测试的测量数据。第 2 章中还引用了多篇使用这些数据的论文，列于图注中，包括：Agarwal 和 Pudar [1993]；Barroso、Gharachorloo 和 Bugnion [1998]；Farkas 和 Jouppi [1994]；Jouppi [1990]；Lam、Rothberg 和 Wolf [1991]；Lebeck 和 Wood [1994]；McCalpin [2005]；Mowry、Lam 和 Gupta [1992]；以及 Torrellas、Gupta 和 Hennessy [1992]。

参考文献

Agarwal, A. [1987]. “Analysis of Cache Performance for Operating Systems and Multiprogramming,” Ph.D. thesis, Tech. Rep. No. CSL-TR-87-332, Stanford University, Palo Alto, Calif.

Agarwal, A., and S. D. Pudar [1993]. “Column-associative caches: A technique for reducing the miss rate of direct-mapped caches,” 20th Annual Int’l. Symposium on Computer Architecture (ISCA), May 16–19, 1993, San Diego, Calif. (Computer Architecture News 21:2 (May), 179–190).

Baer, J.-L., and W.-H. Wang [1988]. “On the inclusion property for multi-level cache hierarchies,” Proc. 15th Annual Int’l. Symposium on Computer Architecture (ISCA), May 30–June 2, 1988, Honolulu, Hawaii, 73–80.

Barham, P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, and R. Neugebauer [2003]. “Xen and the art of virtualization,” Proc. of the 19th ACM Symposium on Operating Systems Principles, October 19–22, 2003, Bolton Landing, N.Y.

Barroso, L. A., K. Gharachorloo, and E. Bugnion [1998]. “Memory system characterization of commercial workloads,” Proc. 25th Annual Int’l. Symposium on Computer Architecture (ISCA), July 3–14, 1998, Barcelona, Spain, 3–14.

Bell, C. G., and W. D. Strecker [1976]. “Computer structures: What have we learned from the PDP-11?” Proc. Third Annual Int’l. Symposium on Computer Architecture (ISCA), January 19–21, 1976, Tampa, Fla., 1–14.

Bhandarkar, D. P. [1995]. Alpha Architecture Implementations, Digital Press, Newton, Mass.

Borg, A., R. E. Kessler, and D. W. Wall [1990]. “Generation and analysis of very long address traces,” Proc. 17th Annual Int’l. Symposium on Computer Architecture (ISCA), May 28–31, 1990, Seattle, Wash., 270–279.

Cantin, J. F., and M. D. Hill [2001]. “Cache performance for selected SPEC CPU2000 benchmarks,” http://www.cs.wisc.edu/multifacet/misc/spec2000cache-data/.

Cantin, J., and M. Hill [2003]. “Cache performance for SPEC CPU2000 benchmarks, version 3.0,” http://www.cs.wisc.edu/multifacet/misc/spec2000cachedata/index.html.

Case, R. P., and A. Padegs [1978]. “The architecture of the IBM System/370,” Communications of the ACM 21:1, 73–96. Also appears in D. P. Siewiorek, C. G. Bell, and A. Newell, Computer Structures: Principles and Examples, McGraw-Hill, New York, 1982, 830–855.

Clark, B., T. Deshane, E. Dow, S. Evanchik, M. Finlayson, J. Herne, and J. Neefe Matthews [2004]. “Xen and the art of repeated research,” Proc. USENIX Annual Technical Conf., June 27–July 2, 2004, Boston, 1135–1144.

Clark, D. W. [1983]. “Cache performance of the VAX-11/780,” ACM Trans. on Computer Systems 1:1, 24–37.

Clark, D.W., and J. S. Emer [1985]. “Performance of the VAX-11/780 translation buffer: Simulation and measurement,” ACM Trans. on Computer Systems 3:1 (February), 31–62.

Compaq Computer Corporation. [1999]. Compiler Writer’s Guide for the Alpha 21264, Order Number EC-RJ66A-TE, June.

Conti, C., D. H. Gibson, and S. H. Pitkowsky [1968]. “Structural aspects of the System/360 Model 85. Part I. General organization,” IBM Systems J. 7:1, 2–14.

Crawford, J., and P. Gelsinger [1988]. Programming the 80386, Sybex, Alameda, Calif.

Cvetanovic, Z., and R. E. Kessler [2000]. “Performance analysis of the Alpha 21264-based Compaq ES40 system,” Proc. 27th Annual Int’l. Symposium on Computer Architecture (ISCA), June 10–14, 2000, Vancouver, Canada, 192–202.

Fabry, R. S. [1974]. “Capability based addressing,” Communications of the ACM 17:7 (July), 403–412.

Farkas, K. I., P. Chow, N. P. Jouppi, and Z. Vranesic [1997]. “Memory-system design considerations for dynamically-scheduled processors,” Proc. 24th Annual Int’l. Symposium on Computer Architecture (ISCA), June 2–4, 1997, Denver, Colo., 133–143.

Farkas, K. I., and N. P. Jouppi [1994]. “Complexity/performance trade-offs with non-blocking loads,” Proc. 21st Annual Int’l. Symposium on Computer Architecture (ISCA), April 18–21, 1994, Chicago.

Farkas, K. I., N. P. Jouppi, and P. Chow [1995]. “How useful are non-blocking loads, stream buffers and speculative execution in multiple issue processors?” Proc. First IEEE Symposium on High-Performance Computer Architecture, January 22–25, 1995, Raleigh, N.C., 78–89.

Gao, Q. S. [1993]. “The Chinese remainder theorem and the prime memory system,” 20th Annual Int’l. Symposium on Computer Architecture (ISCA), May 16–19, 1993, San Diego, Calif. (Computer Architecture News 21:2 (May), 337–340).

Gee, J. D., M. D. Hill, D. N. Pnevmatikatos, and A. J. Smith [1993]. “Cache performance of the SPEC92 benchmark suite,” IEEE Micro 13:4 (August), 17–27. Gibson, D. H. [1967]. “Considerations in block-oriented systems design,” AFIPS Conf. Proc. 30, 75–80.

Handy, J. [1993]. The Cache Memory Book, Academic Press, Boston.

Heald, R., K. Aingaran, C. Amir, M. Ang, M. Boland, A. Das, P. Dixit, G. Gouldsberry, J. Hart, T. Horel, W.-J. Hsu, J. Kaku, C. Kim, S. Kim, F. Klass, H. Kwan, R. Lo, H. McIntyre, A. Mehta, D. Murata, S. Nguyen, Y.-P. Pai, S. Patel, K. Shin, K. Tam, S. Vishwanthaiah, J. Wu, G. Yee, and H. You [2000]. "Implementation of third-generation SPARC V9 64-b microprocessor," ISSCC Digest of Technical Papers, 412–413 and slide supplement.

Hill, M. D. [1987]. "Aspects of Cache Memory and Instruction Buffer Performance," Ph.D. thesis, Tech. Rep. UCB/CSD 87/381, Computer Science Division, University of California, Berkeley.

Hill, M. D. [1988]. "A case for direct mapped caches," Computer 21:12 (December), 25–40.

Horel, T., and G. Lauterbach [1999]. "UltraSPARC-III: Designing thirdgeneration 64-bit performance," IEEE Micro 19:3 (May–June), 73–85.

Hughes, C. J., P. Kaul, S. V. Adve, R. Jain, C. Park, and J. Srinivasan [2001]. "Variability in the execution of multimedia applications and implications for architecture," Proc. 28th Annual Int'l. Symposium on Computer Architecture (ISCA), June 30–July 4, 2001, Goteborg, Sweden, 254–265.

IEEE. [2005]. "Intel virtualization technology, computer," IEEE Computer Society 38:5 (May), 48–56.

Jouppi, N. P. [1990]. "Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers," Proc. 17th Annual Int'l. Symposium on Computer Architecture (ISCA), May 28–31, 1990, Seattle, Wash., 364–373.

Jouppi, N. P. [1998]. "Retrospective: Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers," in G. S. Sohi, ed., 25 Years of the International Symposia on Computer Architecture (Selected Papers), ACM, New York, 71–73.

Jouppi, N. P., and S. J. E. Wilton [1994]. "Trade-offs in two-level on-chip caching," Proc. 21st Annual Int'l. Symposium on Computer Architecture (ISCA), April 18–21, 1994, Chicago, 34–45.

Kessler, R. E. [1999]. "The Alpha 21264 microprocessor," IEEE Micro 19:2 (March/April), 24–36.

Kilburn, T., D. B. G. Edwards, M. J. Lanigan, and F. H. Sumner [1962]. "One-level storage system," IRE Trans. on Electronic Computers EC-11 (April) 223–235. Also appears in D. P. Siewiorek, C. G. Bell, and A. Newell, Computer Structures: Principles and Examples, McGraw-Hill, New York, 1982, 135–148.

Kroft, D. [1981]. "Lockup-free instruction fetch/prefetch cache organization," Proc. Eighth Annual Int'l. Symposium on Computer Architecture (ISCA), May 12–14, 1981, Minneapolis, Minn., 81–87.

Kroft, D. [1998]. "Retrospective: Lockup-free instruction fetch/prefetch cache organization," in G. S. Sohi, ed., 25 Years of the International Symposia on Computer Architecture (Selected Papers), ACM, New York, 20–21.

Kunimatsu, A., N. Ide, T. Sato, Y. Endo, H. Murakami, T. Kamei, M. Hirano, F. Ishihara, H. Tago, M. Oka, A. Ohba, T. Yutaka, T. Okada, and M. Suzuoki

[2000]. “Vector unit architecture for emotion synthesis,” IEEE Micro 20:2 (March–April), 40–47.

Lam, M. S., E. E. Rothberg, and M. E. Wolf [1991]. “The cache performance and optimizations of blocked algorithms,” Proc. Fourth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 8–11, 1991, Santa Clara, Calif. (SIGPLAN Notices 26:4 (April), 63–74).

Lebeck, A. R., and D. A. Wood [1994]. “Cache profiling and the SPEC benchmarks: A case study,” Computer 27:10 (October), 15–26.

Liptay, J. S. [1968]. “Structural aspects of the System/360 Model 85. Part II. The cache,” IBM Systems J. 7:1, 15–21.

Luk, C.-K., and T. C Mowry [1999]. “Automatic compiler-inserted prefetching for pointer-based applications,” IEEE Trans. on Computers, 48:2 (February), 134–141.

McCalpin, J. D. [2005]. “STREAM: Sustainable Memory Bandwidth in High Performance Computers,” www.cs.virginia.edu/stream/.

McFarling, S. [1989]. “Program optimization for instruction caches,” Proc. Third Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 3–6, 1989, Boston, 183–191.

Menon, A., J. Renato Santos, Y. Turner, G. Janakiraman, and W. Zwaenepoel [2005]. “Diagnosing performance overheads in the xen virtual machine environment,” Proc. First ACM/USENIX Int’l. Conf. on Virtual Execution Environments, June 11–12, 2005, Chicago, 13–23.

Meyer, R. A., and L. H. Seawright [1970]. “A virtual machine time sharing system,” IBM Systems J. 9:3, 199–218.

Mowry, T. C., S. Lam, and A. Gupta [1992]. “Design and evaluation of a compiler algorithm for prefetching,” Proc. Fifth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 12–15, 1992, Boston (SIGPLAN Notices 27:9 (September), 62–73).

Oka, M., and M. Suzuoki [1999]. “Designing and programming the emotion engine,” IEEE Micro 19:6 (November–December), 20–28.

Pabst, T. [2000]. “Performance Showdown at 133 MHz FSB—The Best Platform for Coppermine,” www6.tomshardware.com/mainboard/00q1/000302/.

alacharla, S., and R. E. Kessler [1994]. “Evaluating stream buffers as a secondary cache replacement,” Proc. 21st Annual Int’l. Symposium on Computer Architecture (ISCA), April 18–21, 1994, Chicago, 24–33.

Przybylski, S. A. [1990]. Cache Design: A Performance-Directed Approach, Morgan Kaufmann, San Francisco.

Przybylski, S. A., M. Horowitz, and J. L. Hennessy [1988]. “Performance tradeoffs in cache design,” Proc. 15th Annual Int’l. Symposium on Computer Architecture (ISCA), May 30–June 2, 1988, Honolulu, Hawaii, 290–298.

Reinman, G., and N. P. Jouppi. [1999]. “Extensions to CACTI.”

Robin, J., and C. Irvine [2000]. “Analysis of the Intel Pentium’s ability to support a secure virtual machine monitor,” Proc. USENIX Security Symposium, August 14–17, 2000, Denver, Colo.

Saavedra-Barrera, R. H. [1992]. “CPU Performance Evaluation and Execution Time Prediction Using Narrow Spectrum Benchmarking,” Ph.D. dissertation, University of California, Berkeley.

Samples, A. D., and P. N. Hilfinger [1988]. Code Reorganization for Instruction Caches, Tech. Rep. UCB/CSD 88/447, University of California, Berkeley.

Sites, R. L. (ed.) [1992]. Alpha Architecture Reference Manual, Digital Press, Burlington, Mass.

Skadron, K., and D. W. Clark [1997]. “Design issues and tradeoffs for write buffers,” Proc. Third Int’l. Symposium on High-Performance Computer Architecture, February 1–5, 1997, San Antonio, Tex., 144–155.

Smith, A. J. [1982]. “Cache memories,” Computing Surveys 14:3 (September), 473–530.

Smith, J. E., and J. R. Goodman [1983]. “A study of instruction cache organizations and replacement policies,” Proc. 10th Annual Int’l. Symposium on Computer Architecture (ISCA), June 5–7, 1982, Stockholm, Sweden, 132–137.

Stokes, J. [2000]. “Sound and Vision: A Technical Overview of the Emotion Engine,” http://arstechnica.com/hardware/reviews/2000/02/ee.ars.

Strecker, W. D. [1976]. “Cache memories for the PDP-11?” Proc. Third Annual Int’l. Symposium on Computer Architecture (ISCA), January 19–21, 1976, Tampa, Fla., 155–158.

Sugumar, R. A., and S. G. Abraham [1993]. “Efficient simulation of caches under optimal replacement with applications to miss characterization,” Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, May 17–21, 1993, Santa Clara, Calif., 24–35.

Tarjan, D., S. Thoziyoor, and N. Jouppi [2006]. CACTI 4.0. Technical Report HPL-2006-86, HP Laboratories.

Torrellas, J., A. Gupta, and J. Hennessy [1992]. “Characterizing the caching and synchronization performance of a multiprocessor operating system,” Proc. Fifth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 12–15, 1992, Boston (SIGPLAN Notices 27:9 (September), 162–174).

Van Vleck, T. [2005]. “The IBM 360/67 and CP/CMS,” http://www.multicians.org/thvv/360-67.html.

Wang, W.-H., J.-L. Baer, and H. M. Levy [1989]. “Organization and performance of a two-level virtual-real cache hierarchy,” Proc. 16th Annual Int’l. Symposium on Computer Architecture (ISCA), May 28–June 1, 1989, Jerusalem, 140–148.

Wilkes, M. [1965]. “Slave memories and dynamic storage allocation,” IEEE Trans. Electronic Computers EC-14:2 (April), 270–271.

Wilkes, M. V. [1982]. “Hardware support for memory protection: Capability implementations,” Proc. Symposium on Architectural Support for Programming Languages and Operating Systems (ASPLOS), March 1–3, 1982, Palo Alto, Calif., 107–116.

Wulf, W. A., R. Levin, and S. P. Harbison [1981]. Hydra/C.mmp: An Experimental Computer System, McGraw-Hill, New York.

M.4 指令集的发展

每当设计出采用堆栈式或寄存器式指令集的未来架构时，人们的眉头都应当微微扬起。

Meyers 原文：*One's eyebrows should rise whenever a future architecture is developed with a stack- or register-oriented instruction set.*

—— Meyers [1978，第 20 页]

最早的计算机，包括 UNIVAC I、EDSAC 和 IAS 计算机，都是基于累加器的计算机。这种类型计算机的简洁性使其在硬件资源非常有限的情况下成为自然的选择。第一台通用寄存器计算机是由 Ferranti 有限公司于 1956 年制造的 Pegasus。Pegasus 拥有八个通用寄存器，其中 R0 始终为零。块传输操作将八个寄存器的数据从鼓式存储器中加载。

堆栈体系结构

1963 年，Burroughs 公司推出了 B5000。B5000 或许是第一台认真考虑软件与硬件-软件权衡的计算机。Barton 和 Burroughs 的设计师们将 B5000 设计成堆栈体系结构（如 Barton [1961]所述）。该堆栈体系结构旨在支持诸如 ALGOL 等高级语言，操作系统 MCP 也是用高级语言编写的。B5000 还是美国制造商生产的第一台支持虚拟内存的计算机。1968 年推出的 B6500（在 Hauck 和 Dent [1968]中有讨论）增加了硬件管理的激活记录。在 B5000 和 B6500 中，堆栈的顶部两个元素保存在处理器中，其余部分保存在内存中。堆栈体系结构带来了良好的代码密度，但只提供了两个高速存储位置。IBM 360 原始论文的作者[Amdahl, Blaauw, and Brooks 1964]以及 PDP-11 原始论文的作者[Bell et al. 1970]都反对堆栈组织。他们反对堆栈的主要理由有三点：

- 性能来自于高速寄存器，而不是寄存器的使用方式。
- 堆栈组织过于受限，且需要大量的交换和复制操作。
- 堆栈有底部，当堆栈放置在较慢的存储器中时，会导致性能损失。

译者注：这句话的含义是堆栈的空间是有限的，并且堆栈的增长方向是有固定限制的。当堆栈放置在较慢的存储器中时，访问堆栈底部附近的数据会比较慢，导致整体性能下降。

基于堆栈的硬件在 1970 年代末逐渐失去关注，除了 Intel 80x86 的浮点架构外，基本上消失了；实际上，除了 80x86 之外，SPEC 报告中列出的计算机几乎没有使用堆栈结构。

然而，在 1990 年代，随着 Java 虚拟机（JVM）的成功，堆栈架构重新获得了关注。JVM 是一个软件解释器，用于执行由 Java 编译器生成的中间语言，称为 Java 字节码[Lindholm and Yellin 1999]。该解释器的目的是在多个平台之间提供软件兼容性，希望实现“编写一次，到处运行”的目标。尽管由于解释执行，性能会降低大约 10 倍，但在某些情况下，兼容性比性能更重要，比如将 Java“小程序”下载到互联网浏览器时。

虽然有人提出过直接硬件执行 JVM 指令的方案（参见 McGhan 和 O'Connor [1998]），但到目前为止，这些方案在商业上尚未取得显著成功。人们更希望通过即时编译（JIT）Java 编译器——在运行时将 Java 程序编译成本地指令集——来克服解释执行带来的性能损失。Java 的流行也促使出现了直接编译成本地硬件指令集的编译器，从而绕过了 Java 字节码的“中间层”假象。

计算机体系结构的定义

IBM 在 1960 年代初创造了“计算机体系结构”（computer architecture）这一术语。Amdahl、Blaauw 和 Brooks 在 1964 年的论文中使用该术语，指代 IBM 360 指令集中的程序员可见部分。他们认为，同一体系结构系列的计算机应该能够运行相同的软件。虽然这一观点在今天看来很显而易见，但在当时却相当新颖。尽管 IBM 是业界的领导者，但在 360 之前，它拥有五种不同的体系结构；因此，一家公司统一采用单一体系结构的理念是非常激进的。

360 的设计者希望通过定义一个通用体系结构，将 IBM 内部六个不同的部门团结起来。他们对计算机体系结构的定义是：

体系结构是计算机的结构，机器语言程序员必须理解该结构，才能为该机器编写正确的（与硬件时序无关的）程序。◆

◆论文的原文是：… the structure of a computer that a machine language programmer mustunderstand to write a correct (timing independent) program for that machine.

这里的“机器语言程序员”意味着即使在机器语言层面也能保持兼容性，而“与时序无关”则允许不同的硬件实现方式。这个架构为二进制兼容性开辟了道路，后来被其他厂商所效仿。

IBM 360 是第一台大规模销售的计算机，它采用了 8 位字节的字节寻址方式和通用寄存器。360 还支持寄存器-内存和有限的内存-内存指令。附录 K 对该指令集进行了总结。

1964 年，CDC 交付了第一台超级计算机 CDC 6600。正如 Thornton [1964] 所述，他、Cray 和其他 6600 设计者是最早深入探索流水线技术的人之一。6600 是第一台通用的加载-存储（load-store）计算机。在 1960 年代，6600 的设计者们意识到，为了实现高效的流水线处理，需要简化架构。微处理器和小型计算机设计者在 1970 年代大多忽视了架构简化与实现效率之间的这种相互关系，但这一问题在 1980 年代重新受到关注。

高级语言计算机体系结构

在 1960 年代末和 1970 年代初，人们意识到软件成本增长速度超过了硬件成本。McKeeman [1967] 指出，编译器和操作系统变得过于庞大和复杂，开发周期过长。由于当时编译器性能较差且计算机内存有限，大多数系统程序仍然使用汇编语言编写。许多研究人员提出通过创建更强大、面向软件的架构来缓解软件危机。Tanenbaum [1978] 研究了高级语言的特性。和其他研究者一样，他发现大多数程序相对简单。他主张架构设计应考虑这一点，优化程序大小和编译的简易性。Tanenbaum 提出了一种采用频率编码指令格式的堆栈计算机以实现这些目标；然而，正如我们所观察到的，程序大小并不直接等同于成本性能比，堆栈计算机在此后不久便逐渐消失。

Strecker 的文章 [1978] 讨论了他和 DEC 的其他架构师如何通过设计 VAX 架构来应对这一问题。VAX 旨在简化高级语言的编译。编译器开发者曾抱怨 PDP-11 缺乏完全的正交性。VAX 架构设计高度正交，允许将高级语言语句映射为单条 VAX 指令。此外，VAX 设计者还试图优化代码大小，因为编译后的程序往往太大，难以适应当时有限的内存。附录 K 对该指令集进行了总结。

◆译者注：这里的“正交性”主要说的是同一指令集系统设计中指令、寻址方式和操作数类型之间可以自由组合的特性，这些要素彼此之间没有不兼容或限制。

VAX-11/780 是 VAX 系列中首款发布的计算机，也是历史上最成功且研究最为深入的计算机之一。DEC 的战略基石是统一的架构 VAX，运行统一的

操作系统 VMS。这一战略在十多年内取得了良好效果。大量关于 VAX 指令混合、实现测量及分析的论文使其成为理想的案例研究对象 [Clark and Levy 1982; Wiecek 1982]。Bhandarkar 和 Clark [1991]对 VAX 与 RISC 计算机的劣势进行了定量分析，基本上为 VAX 的衰落提供了技术解释。

在设计 VAX 的同时，研究界还倡导了一种更为激进的方法，称为高级语言计算机体系结构（High-Level Language Computer Architecture，简称 HLLCA）。这一运动旨在消除高级语言与计算机硬件之间的差距——Gagliardi [1973]称之为“语义鸿沟”，即通过将硬件“提升”到编程语言的层次来实现这一目标。Meyers [1982]对这一观点及高级语言计算机架构项目的历史做了很好的总结。然而，HLLCA 并未在商业上产生显著影响。计算机内存容量的增加解决了由高级语言导致的代码体积问题，并使得操作系统能够用高级语言编写。更简单的架构与软件的结合，提供了更高的性能、更大的灵活性，同时降低了成本和复杂度。

精简指令集计算机

20 世纪 80 年代初，计算机体系结构的发展方向开始从为高级语言提供硬件支持转向更简化的设计。Ditzel 和 Patterson [1980] 分析了高级语言体系结构遇到的困难，认为解决方案在于更简单的架构。在另一篇论文中 [Patterson 和 Ditzel 1980]，他们首次讨论了精简指令集计算机（Reduced Instruction Set Computer, RISC）的概念，并提出了简化架构的论点。作为 VAX 体系结构设计者，Clark 和 Strecker [1980] 对他们的提议提出了反驳。

像 MIPS 这样的简单 load-store 计算机通常被称为 RISC 架构。RISC 架构的起源可以追溯到类似 6600 的计算机，当时 Thornton、Cray 等人认识到指令集简单性在构建高速计算机中的重要性。Cray 在 CRAY-1 中延续了保持计算机简洁的传统。商业 RISC 主要基于三个研究项目的成果：伯克利 RISC 处理器、IBM 801 和斯坦福 MIPS 处理器。这些架构因声称在相同技术条件下性能比其他计算机高出 2 到 5 倍而引起了工业界的极大兴趣。

始于 1975 年的 IBM 项目是最早启动但最晚公开的项目。IBM 计算机被设计为 24 位 ECL 小型机，而大学项目则是基于 MOS 的 32 位微处理器。John Cocke 被认为是 801 设计之父，因其贡献获得了 Eckert–Mauchly 奖和图灵奖。Radin [1982] 描述了 801 架构的亮点。801 是一个实验性项目，从未打算成为商业产品。实际上，为了降低成本和复杂性，该计算机仅配备了 24 位寄存器。

1980 年，Patterson 和他在伯克利的同事们启动了一个项目，赋予了这种架构方法“RISC”这一名称（see Patterson and Ditzel [1980]）。他们构建了两台计算机，分别称为 RISC-I 和 RISC-II。由于 IBM 项目并不为大众所熟知，伯克利团队在推广 RISC 方法中的作用对该技术的接受至关重要。他们还设计了最早支持混合格式 RISC 的指令缓存之一（see Patterson et al. [1983]），该缓存支持内存中的 16 位和 32 位指令，但缓存中均为 32 位。伯克利团队随后还开发了面向 Smalltalk 的 RISC 计算机（由 Ungar 等人 [1984] 介绍）和面向 LISP 的 RISC 计算机（由 Taylor 等人 [1986] 介绍）。

1981 年，Hennessy 和他在斯坦福的同事们发表了对斯坦福 MIPS 计算机的介绍。高效的流水线设计和编译器辅助的流水线调度是 MIPS 原始设计中的两

个重要方面。MIPS代表“无互锁流水线阶段的微处理器”(Microprocessor without Interlocked Pipeline Stages), 反映了该设计缺少硬件用于暂停流水线的机制, 因为编译器负责处理指令间的依赖关系。

这些早期的RISC计算机——801、RISC-II和MIPS——有许多共同点。两个大学项目都致力于设计一种能够在大学环境中用超大规模集成电路 (VLSI) 制造的简单计算机。所有三台计算机都采用了简单的load-store架构和固定格式的32位指令, 并强调高效的流水线设计。Patterson [1985] 描述了这三台计算机及其基本设计原则, 这些原则后来成为RISC计算机的典型特征; Hennessy [1984] 则从另一个角度阐述了相同的理念, 以及VLSI处理器设计中的其他问题。

1985年, Hennessy发表了一篇关于RISC性能优势的解释, 指出其根源在于显著更低的每条指令周期数 (CPI): RISC处理器的CPI低于2, 而VAX-11/780的CPI超过10 (尽管两者的工作负载不完全相同)。Emer和Clark [1984] 的一篇论文对VAX-11/780性能的分析, 帮助RISC研究人员理解了其计算机性能优势的来源。

自1983-1984年大学项目完成后, 这项技术被工业界广泛接受。许多早期计算机制造商 (1986年前生产的) 声称他们的产品是RISC计算机, 但这些声明往往更多是出于市场营销的目的, 而非工程实际。

1986年, 计算机行业开始推出基于这三个RISC研究项目技术的处理器。Moussouris等人 [1986] 描述了MIPS R2000整数处理器, Kane的书籍 [1986] 则提供了该架构的完整描述。惠普将其现有的小型机系列转换为RISC架构; Lee [1989] 描述了HP Precision Architecture。IBM没有直接将801转化为产品, 而是将其理念应用于一种新的低端架构, 该架构被整合进IBM RT-PC, 并在一系列论文中有所介绍 [Waters 1986]。1990年, IBM发布了新的RISC架构RS 6000, 这是首款超标量RISC处理器。1987年, Sun Microsystems开始推出基于SPARC架构的计算机, SPARC是伯克利RISC-II处理器的衍生架构; 相关内容见Garner等人 [1988]。PowerPC架构则是苹果、IBM和摩托罗拉三方的合作成果。附录K总结了几种RISC架构。

为了帮助解决RISC与传统设计的争论, VAX处理器设计者后来对VAX和RISC处理器进行了定量比较, 选取了结构相近的VAX 8700和MIPS M2000。VAX和MIPS的设计目标不同, 导致架构差异显著。VAX注重简单的编译器和代码密度, 采用了强大的寻址模式、复杂指令、高效的指令编码和少量寄存器; 而MIPS追求通过流水线实现高性能、硬件实现简易性以及与高度优化编译器的兼容性, 因此采用了简单指令、简单寻址模式、固定长度指令格式和大量寄存器。

图M.1显示了执行指令数比率、CPI比率和以时钟周期计的性能比率。由于结构相似, 假设时钟周期时间相同。MIPS执行的指令数量约是VAX的两倍, 而VAX的CPI约是MIPS的六倍。因此, MIPS M2000的性能几乎是VAX 8700的三倍。此外, 构建MIPS处理器所需的硬件远少于VAX处理器。这种性能与成本的差距促使曾经制造VAX的公司推出基于MIPS的产品, 最终完全放弃了VAX, 转而采用与MIPS非常相似的Alpha架构。Bell和Strecker [1998] 总结了

该公司内部的争论。如今，曾经是第二大计算机公司、迷你计算机行业的主要成功者 DEC，只作为 HP 和 Intel 的残余存在。

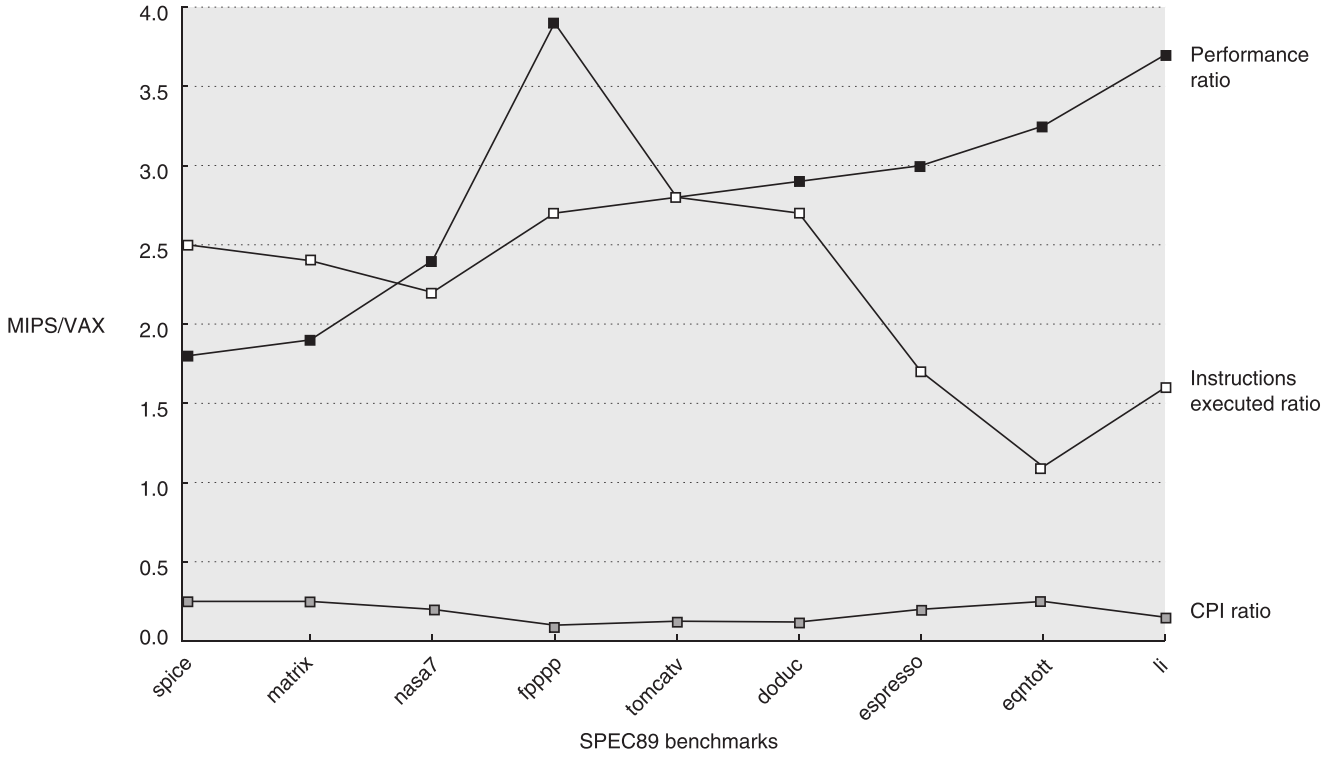

图 M.1 使用 SPEC89 程序测得的 MIPS M2000 与 VAX 8700 在执行指令数和时钟周期性能方面的比率。平均来看，MIPS 执行的指令数量略多于 VAX 的两倍，但 VAX 的每条指令周期数（CPI）几乎是 MIPS 的六倍，因此 MIPS 在性能上具有近三倍的优势。（数据基于 Bhandarkar 和 Clark [1991]。）

回顾过去，在 RISC 与 CISC 的争论中，只有一种复杂指令集计算机指令集得以存活下来，而且该指令集与 PC 软件保持二进制兼容。由于 PC 行业芯片的出货量极大，产生了足够的收入流来支付额外的设计成本，同时得益于摩尔定律带来的资源充裕，使得制造能够在内部将 CISC 指令转换为 RISC 指令的微处理器成为可能。尽管这种转换导致流水线阶段变长、芯片面积增大，从而降低了效率，但这种损失被庞大的市场规模和专门为该产品配备的集成电路生产线所弥补。

有趣的是，Intel 也曾认为 80x86 系列的未来充满不确定性。于是他们设计了 IA-64 架构，以支持 64 位寻址并向 RISC 风格指令集转变。IA-64 架构的实现体现在 Itanium-1 和 Itanium-2 处理器上（see Huck et al. [2000]），其成效喜忧参半。虽然在浮点应用方面取得了高性能，但整数性能一直不尽如人意。此外，Itanium 处理器的晶体管数量和芯片面积都很大，功耗也较高。IA-64 指令集的复杂性，与 RISC 理念至少部分相悖，无疑是导致其功耗和面积效率低下的原因之一。

相比之下，AMD 选择了直接将 32 位地址扩展到 64 位地址，类似于 Intel 当年将 80386 从 16 位地址扩展到 32 位地址的做法。随后 Intel 也效仿了 AMD。最终，80x86 架构在市场上的巨大优势，甚至让这项遗产的拥有者 Intel 也难以撼动。

参考文献

Alexander, W. G., and D. B. Wortman [1975]. "Static and dynamic characteristics of XPL programs," IEEE Computer 8:11 (November), 41–46.

Amdahl, G. M., G. A. Blaauw, and F. P. Brooks, Jr. [1964]. "Architecture of the IBM System 360," IBM J. Research and Development 8:2 (April), 87–101.

Barton, R. S. [1961]. "A new approach to the functional design of a computer," Proc. Western Joint Computer Conf., May 9–11, 1961, Los Angeles, Calif., 393–396.

Bell, G., R. Cady, H. McFarland, B. DeLagi, J. O'Laughlin, R. Noonan, and W. Wulf [1970]. "A new architecture for mini-computers: The DEC PDP-11," Proc. AFIPS SJCC, May 5–7, 1970, Atlantic City, N.J., 657–675.

Bell, G., and W. D. Strecker [1998]. "Computer structures: What have we learned from the PDP-11?" in G. S. Sohi, ed., 25 Years of the International Symposia on Computer Architecture (Selected Papers), ACM, New York, 138–151.

Bhandarkar, D. P. [1995]. Alpha Architecture and Implementations, Digital Press, Newton, Mass.

Bhandarkar, D., and D. W. Clark [1991]. "Performance from architecture: Comparing a RISC and a CISC with similar hardware organizations," Proc. Fourth nt'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 8–11, 1991, Palo Alto, Calif., 310–319.

Bier, J. [1997]. "The evolution of DSP processors," paper presented at University of California, Berkeley, November 14.

Boddie, J. R. [2000]. "History of DSPs," www.lucent.com/micro/dsp/dsphist.html.

Case, R. P., and A. Padegs [1978]. "The architecture of the IBM System/370," Communications of the ACM 21:1, 73–96.

Chow, F. C. [1983]. "A Portable Machine-Independent Global Optimizer— Design and Measurements," Ph.D. thesis, Stanford University, Palo Alto, Calif.

Clark, D., and H. Levy [1982]. "Measurement and analysis of instruction set use in the VAX-11/780," Proc. Ninth Annual Int'l. Symposium on Computer Architecture (ISCA), April 26–29, 1982, Austin, Tex., 9–17.

Clark, D., and W. D. Strecker [1980]. "Comments on 'the case for the reduced instruction set computer,'" Computer Architecture News 8:6 (October), 34–38.

Crawford, J., and P. Gelsinger [1988]. Programming the 80386, Sybex Books, Alameda, Calif.

Darcy, J. D., and D. Gay [1996]. "FLECKmarks: Measuring floating point performance using a full IEEE compliant arithmetic benchmark," CS 252 class project, University of California, Berkeley (see http://www.sonic.net/#jddarcy/Research/fleckmrk.pdf).

Digital Semiconductor. [1996]. Alpha Architecture Handbook, Version 3, Digital Press, Maynard, Mass.

Ditzel, D. R., and D. A. Patterson [1980]. "Retrospective on high-level language computer architecture," Proc. Seventh Annual Int'l. Symposium on Computer Architecture (ISCA), May 6–8, 1980, La Baule, France, 97–104.

Emer, J. S., and D. W. Clark [1984]. “A characterization of processor performance in the VAX-11/780,” Proc. 11th Annual Int’l. Symposium on Computer Architecture (ISCA), June 5–7, 1984, Ann Arbor, Mich., 301–310.

Furber, S. B. [2000]. ARM system-on-chip architecture. Addison-Wesley, Boston, Mass.

Gagliardi, U. O. [1973]. “Report of workshop 4—software-related advances in computer hardware,” Proc. Symposium on the High Cost of Software, September 17–19, 1973, Monterey, Calif., 99–120.

Game, M., and A. Booker [1999]. “CodePack code compression for PowerPC processors,” MicroNews, 5:1.

Garner, R., A. Agarwal, F. Briggs, E. Brown, D. Hough, B. Joy, S. Kleiman, S. Muchnick, M. Namjoo, D. Patterson, J. Pendleton, and R. Tuck [1988]. “Scalable processor architecture (SPARC),” Proc. IEEE COMPCON, February 29– March 4, 1988, San Francisco, 278–283.

Hauck, E. A., and B. A. Dent [1968]. “Burroughs’ B6500/B7500 stack mechanism,” Proc. AFIPS SJCC, April 30–May 2, 1968, Atlantic City, N.J., 245–251.

Hennessy, J. [1984]. “VLSI processor architecture,” IEEE Trans. on Computers C-33:11 (December), 1221–1246.

Hennessy, J. [1985]. “VLSI RISC processors,” VLSI Systems Design 6:10 (October), 22–32.

Hennessy, J., N. Jouppi, F. Baskett, and J. Gill [1981]. “MIPS: A VLSI processor architecture,” in CMU Conference on VLSI Systems and Computations, Computer Science Press, Rockville, Md.

Hewlett-Packard. [1994]. PA-RISC 2.0 Architecture Reference Manual, 3rd ed., Hewlett-Packard, Palo Alto, Calif.

Hitachi. [1997]. SuperH RISC Engine SH7700 Series Programming Manual, Hitachi, Santa Clara, Calif.

Huck, J. et al. [2000]. “Introducing the IA-64 Architecture” IEEE Micro, 20:5, (September–October), 12–23.

IBM. [1994]. The PowerPC Architecture, Morgan Kaufmann, San Francisco.

Intel. [2001]. “Using MMX instructions to convert RGB to YUV color conversion,” cedar.intel.com/cgi-bin/ids.dll/content/content.jsp?cntKey=Legacy:: irtm_AP548_9996&cntType=IDS_EDITORIAL.

Kahan, J. [1990]. “On the advantage of the 8087’s stack,” unpublished course notes, Computer Science Division, University of California, Berkeley.

Kane, G. [1986]. MIPS R2000 RISC Architecture, Prentice Hall, Englewood Cliffs, N.J.

Kane, G. [1996]. PA-RISC 2.0 Architecture, Prentice Hall, Upper Saddle River, N.J.

Kane, G., and J. Heinrich [1992]. MIPS RISC Architecture, Prentice Hall, Englewood Cliffs, N.J.

Kissell, K. D. [1997]. “MIPS16: High-density for the embedded market,” Proc. Real Time Systems ’97, June 15, 1997, Las Vegas, Nev.

Kozyrakis, C. [2000]. "Vector IRAM: A media-oriented vector processor with embedded DRAM," paper presented at Hot Chips 12, August 13–15, 2000, Palo Alto, Calif, 13–15.

Lee, R. [1989]. "Precision architecture," Computer 22:1 (January), 78–91.

Levy, H., and R. Eckhouse [1989]. Computer Programming and Architecture: The VAX, Digital Press, Boston.

Lindholm, T., and F. Yellin [1999]. The Java Virtual Machine Specification, 2nd ed., Addison-Wesley, Reading, Mass.

Lunde, A. [1977]. "Empirical evaluation of some features of instruction set processor architecture," Communications of the ACM 20:3 (March), 143–152.

Magenheimer, D. J., L. Peters, K. W. Pettis, and D. Zuras [1988]. "Integer multiplication and division on the HP precision architecture," IEEE Trans. on Computers 37:8, 980–990.

McGhan, H., and M. O'Connor [1998]. "PicoJava: A direct execution engine for Java bytecode," Computer 31:10 (October), 22–30.

McKeeman, W. M. [1967]. "Language directed computer design," Proc. AFIPS Fall Joint Computer Conf., November 14–16, 1967, Washington, D.C., 413–417.

Meyers, G. J. [1978]. "The evaluation of expressions in a storage-to-storage architecture," Computer Architecture News 7:3 (October), 20–23.

Meyers, G. J. [1982]. Advances in Computer Architecture, 2nd ed., Wiley, New York.

MIPS. [1997]. MIPS16 Application Specific Extension Product Description.

Mitsubishi. [1996]. Mitsubishi 32-Bit Single Chip Microcomputer M32R Family Software Manual, Mitsubishi, Cypress, Calif.

Morse, S., B. Ravenal, S. Mazor, and W. Pohlman [1980]. "Intel microprocessors — 8080 to 8086," Computer 13:10 (October).

Moussouris, J., L. Crudele, D. Freitas, C. Hansen, E. Hudson, S. Przybylski, T. Riordan, and C. Rowen [1986]. "A CMOS RISC processor with integrated system functions," Proc. IEEE COMPCON, March 3–6, 1986, San Francisco, 191.

Muchnick, S. S. [1988]. "Optimizing compilers for SPARC," Sun Technology 1:3 (Summer), 64–77.

Palmer, J., and S. Morse [1984]. The 8087 Primer, John Wiley & Sons, New York, 93.

Patterson, D. [1985]. "Reduced instruction set computers," Communications of the ACM 28:1 (January), 8–21.

Patterson, D. A., and D. R. Ditzel [1980]. "The case for the reduced instruction set computer," Computer Architecture News 8:6 (October), 25–33.

Patterson, D. A., P. Garrison, M. Hill, D. Lioupis, C. Nyberg, T. Sippel, and K. Van Dyke [1983]. "Architecture of a VLSI instruction cache for a RISC," 10th Annual Int'l. Conf. on Computer Architecture Conf. Proc., June 13–16, 1983, Stockholm, Sweden, 108–116.

Radin, G. [1982]. “The 801 minicomputer,” Proc. Symposium Architectural Support for Programming Languages and Operating Systems (ASPLOS), March 1– 3, 1982, Palo Alto, Calif., 39–47.

Riemens, A., K. A. Vissers, R. J. Schutten, F. W. Sijstermans, G. J. Hekstra, and G. D. La Hei [1999]. “Trimedia CPU64 application domain and benchmark suite,” Proc. IEEE Int’l. Conf. on Computer Design: VLSI in Computers and Processors (ICCD’99), October 10–13, 1999, Austin, Tex., 580–585.

Ropers, A., H. W. Lollman, and J. Wellhausen [1999]. DSPstone: Texas Instruments TMS320C54x, Tech. Rep. Nr. IB 315 1999/9-ISS-Version 0.9, Aachen University of Technology, Aaachen, Germany (www.ert.rwth-aachen.de/Projekte/Tools/coal/dspstone_c54x/index.html).

Shustek, L. J. [1978]. “Analysis and Performance of Computer Instruction Sets,” Ph.D. dissertation, Stanford University, Palo Alto, Calif.

Silicon Graphics. [1996]. MIPS V Instruction Set (see http://www.sgi.com/MIPS/arch/ISA5/~MIPSV_indx).

Sites, R. L., and R. Witek, eds. [1995]. Alpha Architecture Reference Manual, 2nd ed., Digital Press, Newton, Mass.

Strauss, W. [1998]. “DSP Strategies 2002,” www.usadata.com/market_research/spr_05/spr_r127-005.htm.

Strecker, W. D. [1978]. “VAX-11/780: A virtual address extension of the PDP-11 family,” Proc. AFIPS National Computer Conf., June 5–8, 1978, Anaheim, Calif., 47, 967–980.

Sun Microsystems. [1989]. The SPARC Architectural Manual, Version 8, Part No. 800-1399-09, Sun Microsystems, Santa Clara, Calif.

Tanenbaum, A. S. [1978]. “Implications of structured programming for machine architecture,” Communications of the ACM 21:3 (March), 237–246.

Taylor, G., P. Hilfinger, J. Larus, D. Patterson, and B. Zorn [1986]. “Evaluation of the SPUR LISP architecture,” Proc. 13th Annual Int’l. Symposium on Computer Architecture (ISCA), June 2–5, 1986, Tokyo.

Texas Instruments [2000]. “History of innovation: 1980s,” www.ti.com/corp/docs/company/history/1980s.shtml.

Thornton, J. E. [1964]. “Parallel operation in Control Data 6600,” Proc. AFIPS Fall Joint Computer Conf., Part II, October 27–29, 1964, San Francisco, 26, 33–40.

Ungar, D., R. Blau, P. Foley, D. Samples, and D. Patterson [1984]. “Architecture of SOAR: Smalltalk on a RISC,” Proc. 11th Annual Int’l. Symposium on Computer Architecture (ISCA), June 5–7, 1984, Ann Arbor, Mich., 188–197.

van Eijndhoven, J. T. J., F. W. Sijstermans, K. A. Vissers, E. J. D. Pol, M. I. A. Tromp, P. Struik, R. H. J. Bloks, P. van der Wolf, A. D. Pimentel, and H. P. E. Vranken [1999]. “Trimedia CPU64 architecture,” Proc. IEEE Int’l. Conf. on Computer Design: VLSI in Computers and Processors (ICCD’99), October 10–13, 1999, Austin, Tex., 586–592.

Wakerly, J. [1989]. Microcomputer Architecture and Programming, Wiley, New York.

Waters, F. (ed.) [1986]. IBM RT Personal Computer Technology, SA 23-1057, IBM, Austin, Tex.

Weaver, D. L., and T. Germond [1994]. The SPARC Architectural Manual, Version 9, Prentice Hall, Englewood Cliffs, N.J.

Weiss, S., and J. E. Smith [1994]. Power and PowerPC, Morgan Kaufmann, San Francisco.

Wiecek, C. [1982]. "A case study of the VAX 11 instruction set usage for compiler execution," Proc. Symposium on Architectural Support for Programming Languages and Operating Systems (ASPLOS), March 1–3, 1982, Palo Alto, Calif., 177–184.

Wulf, W. [1981]. "Compilers and computer architecture," Computer 14:7 (July), 41–47.

M.5 流水线和指令级并行的发展

早期的流水线 CPU

第一个通用流水线处理器被认为是 Stretch，即 IBM 7030。Stretch 继承了 IBM 704，目标是比 704 快 100 倍。这个目标在当时的技术水平上是一个很大的挑战，因此得到了"Stretch"（伸展）的昵称。计划是通过采用四级流水线，实现取指、译码和执行的重叠，从而获得 1.6 倍的性能提升。Bloch [1959] 和 Bucholtz [1962] 描述了该设计及工程上的权衡，包括算术逻辑单元（ALU）旁路的使用。

20 世纪 70 年代末和 80 年代初出现的一系列通用流水线描述，提供了大部分术语并介绍了简单流水线中使用的大部分基本技术。这些综述包括 Keller [1975]、Ramamoorthy 和 Li [1977]、Chen [1980]，以及专门论述流水线的 Kogge [1981]。Davidson 及其同事 [1971, 1975] 提出了流水线预留表（pipeline reservation tables）的概念，作为带反馈的多周期流水线设计方法论（Kogge [1981] 中也有描述）。许多设计者在设计流水线或编写调度软件时，都采用了这些概念的变体。

RISC 处理器最初设计时就考虑了易于实现和流水线化。20 世纪 80 年代初发表的几篇早期 RISC 论文，试图量化指令集简化带来的性能优势。然而，最有说服力的分析是 Bhandarkar 和 Clark 于 1991 年发表的一篇对 VAX 和 MIPS 实现的比较（见图 M.1），这距第一批 RISC 论文发表已有 10 年。经过 10 年的关于 RISC 实现优势的争论，这篇论文使得即使是最怀疑的设计者也信服了 RISC 指令集架构的优越性。

J. E. Smith 及其同事发表了多篇论文，研究高速标量 CPU 的指令发射、异常处理和流水线深度。Kunkel 和 Smith [1986] 评估了流水线开销和依赖关系对最优流水线深度选择的影响，并对锁存器设计及其对流水线的影响进行了出色的讨论。Smith 和 Pleszkun [1988] 评估了多种保持精确异常的技术。Weiss 和 Smith [1984] 评估了多种硬件流水线调度和指令发射技术。

MIPS R4000 是最早的深度流水线微处理器之一，Killian [1991] 和 Heinrich [1993] 对其进行了描述。最初的 Alpha 实现（Alpha 21064）具有类似的指令集和整数流水线结构，浮点单元则采用了更多的流水线级数。

动态指令调度的引入

1964 年，CDC 交付了首台 CDC 6600。CDC 6600 在许多方面都非常独特。除了引入记分牌技术（scoreboarding）外，CDC 6600 还是第一台广泛使用多个功能单元的处理器。它还配备了使用多线程的外围处理器。设计者理解了流水线与指令集设计之间的相互作用，并采用了简单的 load-store 指令集以促进流水线的实现。CDC 6600 还采用了先进的封装技术。Thornton [1964] 描述了流水线和 I/O 处理器架构，包括乱序指令执行的概念。Thornton 的著作 [1970] 对整个处理器进行了详尽描述，从技术到架构，并由 Cray 撰写了前言。（遗憾的是，这本书目前已绝版。）CDC 6600 还配备了用于 FORTRAN 编译器的指令调度器，相关内容由 Thorlin [1967] 进行了描述。

IBM 360 Model 91：具有里程碑意义的计算机

IBM 360/91 引入了许多新概念，包括数据标记（tagging）、寄存器重命名、动态检测内存冲突以及广义的转发技术。Tomasulo 算法在其 1967 年的论文中进行了描述。Anderson、Sparacio 和 Tomasulo [1967] 介绍了该处理器的其他方面，包括分支预测的使用。360/91 中的许多思想在后 25 年内逐渐淡出应用，直到 1990 年代才被广泛重新采用。不幸的是，360/91 并未取得成功，销量非常有限。设计的复杂性导致其推向市场较晚，使得配备缓存的首款 IBM 处理器 Model 85 表现优于 360/91。

分支预测方案

J. E. Smith [1981] 描述了 2 位动态硬件分支预测方案。Ditzel 和 McLellan [1987] 介绍了用于 CRISP 的创新设计的分支目标缓冲器（branch-target buffer），该缓冲器实现了分支折叠（branch folding）。本书研究的相关预测器由 Pan、So 和 Rameh [1992] 描述。Yeh 和 Patt [1992, 1993] 推广了相关性预测的思想，描述了使用每个分支历史的多级预测器，这与 21264 中使用的局部历史预测器类似。McFarling 在其 1993 年的技术报告中描述了他称之为组合预测器（combined predictor）的锦标赛预测方案。近年来还有许多基于多级和相关预测器思想变体的分支预测论文。Kaeli 和 Emma [1991] 描述了返回地址预测，Evers 等人 [1998] 对多级预测器进行了深入分析。第 3 章中展示的数据来自 Skadron 等人 [1999]。除了锦标赛预测器之外，还有几种预测方案可能带来额外的性能提升，Eden 和 Mudge [1998] 以及 Jimenez 和 Lin [2002] 对此类方法进行了描述。

多发射处理器的发展

IBM 在多发射（multiple issue）技术方面做出了开创性工作。1960 年代，加利福尼亚州曾进行一个名为 ACS 的项目。该项目包含了多发射的概念，提出了动态调度方案（虽然比 Tomasulo 方案更简单，使用了备份寄存器），并实现了对分支路径的双向预取。ACS 项目最初作为一个新架构的涉及项目来进行，继 Stretch 之后，旨在超越 CDC 6600/6800。它最初在纽约进行，后来迁至加利福尼亚，修改为兼容 S/360 架构，最终被取消。John Cocke 是该团队的核心成员之一，团队成员包括许多 IBM 资深专家和年轻贡献者，如 Jack Bertram、Ed Sussenguth、

Gene Amdahl、Herb Schorr、Fran Allen、Lynn Conway 和 Phil Dauber 等。尽管编译器团队发表了许多洞见并在 IBM 外部产生了深远影响，但当时该架构思想未被广泛传播。关于 ACS 项目最完整的公开资料可见于 www.cs.clemson.edu/~mark/acs.html，其中包括对 ACS 最初参与者的访谈及相关资料，Sussenguth [1999]提供了 ACS 的良好综述。

实际上早期真正推向市场的多发射处理器大多采用了 LIW 或 VLIW 设计方法。Charlesworth [1981] 报道了 Floating Point Systems 的 AP-120B，这是最早的宽指令处理器之一，每条指令包含多个操作。该公司在编译器和手写汇编库中都应用了软件流水线技术以提升处理器效率。由于 AP-120B 是附加处理器，因此可以忽略通用处理器中实现多发射时遇到的许多难题（如虚拟内存和异常处理）。

早期 VLIW 处理器（如 AP-120B 和 i860）采用了一种有趣的流水线组织方式：操作需要被“推送”通过功能单元，结果则在流水线末端被“接收”。在此类处理器中，操作只有在后续操作推送时才前进；一条指令会指定一个之前发出的指令的结果目标寄存器，当新操作被推入时，之前的结果被推出流水线。这种方法的优点是结果目标寄存器只有在实际写入时才指定，从而避免了硬件中写后写（WAW）和写后读（WAR）冒险的检测。缺点是代码体积增大，因为在存在对仍在流水线中的操作的依赖且没有其他相同类型操作时，可能需要插入空操作（no-ops）来推动结果推出流水线。与此“推送-接收”方案不同，几乎所有设计者都选择了自排空流水线（self-draining pipelines），即在发射指令时指定目标寄存器，指令发射后即可完成，无需额外动作。这种设计在代码密度和代码生成简化方面的优势超过了“推送-接收”结构的好处。

1980 年代中期，多个研究项目引入了某种形式的多发射。例如，斯坦福的 MIPS 处理器曾支持在单条指令中放置两个操作，但出于性能考虑，这一功能在商业版本中被取消。Fisher [1983] 与耶鲁大学同事提出了超长指令字处理器概念，设计了 512 位宽的指令字。该处理器的代码生成采用了 Fisher [1981] 最初为生成水平微码开发的追踪调度（trace scheduling）技术。Fisher 等人 [1984] 和 Ellis [1986] 描述了该处理器追踪调度的实现细节。

虽然 IBM 取消了 ACS 项目，但在 1980 年代该领域的研究仍在继续。ACS 被取消十多年后，John Cocke 提出了一个新的超标量处理器的方案，该处理器能够动态地进行发射决策；他和 Tilak Agerwala 在 1980 年代中期的几次演讲中描述了这些关键理念，并创造了“超标量（superscalar）”这个术语。他称这个设计为 America；这一设计由 Agerwala 和 Cocke [1987] 描述。IBM 的 Power1 架构（RS/6000 系列）基于这些理念（see Bakoglu et al. [1989]）。

J. E. Smith [1984] 和他在威斯康星大学的同事们提出了一种解耦的方法，该方法包括有限动态流水线调度的多发射。该处理器的一个关键特征是使用队列来维护指令类别（如内存引用）之间的顺序，同时允许其在其他指令类别之前或之后滑动。Smith 等人 [1987] 描述的 Astronautics ZS-1 采用了这种方法，使用队列连接 load-store 单元和操作单元。Power2 设计以类似的方式使用队列。J. E. Smith [1989] 还描述了动态调度的优势，并将其与静态调度进行了比较。

推测执行的概念源于最初的 360/91，该处理器执行了一种非常有限的推测形式。最近处理器中使用的方法结合了 360/91 的动态调度技术和一个缓冲区以允许按序提交。Smith 和 Pleszkun [1988] 探讨了使用缓冲来维护精确中断，并描述了重排序缓冲区（re-order buffer）的概念。Sohi [1990] 描述了添加重命名和动态调度，使得推测机制的使用成为可能。Patt 和他的同事是积极重新排序和推测的早期支持者。他们专注于检查点和重启机制，并开创了一种称为 HPSm 的方法，这也是 Tomasulo 算法的扩展 [Hwu and Patt 1986]。

Smith、Johnson 和 Horowitz [1989] 使用重排序缓冲技术评估了在多发射处理器中使用推测的技术；他们的目标是研究在非科学代码中利用推测和多发射的可用 ILP。在随后的书中，Johnson [1990] 描述了一个推测执行超标量处理器的设计。Johnson 后来领导了 AMD K-5 设计，这是最早的推测执行超标量之一。

与超标量发展的同时，对 VLIW 方法的商业兴趣也在增加。Multiflowsee Colwell et al. [1987]）基于耶鲁大学开发的概念，为了提高该方法的实用性进行了许多重要的改进。其中之一是一个可控的存储缓冲区，提供了一种推测形式的支持。虽然售出了超过 100 台 Multiflow 处理器，但由于引入一个小公司新指令集的困难以及商业 RISC 微处理器的竞争改变了小型计算机市场的经济性，导致了 Multiflow 公司的失败。

与 Multiflow 同期，Cydrome 成立，其构建了一种 VLIW 风格的处理器（see Rau et al. [1989]），但在商业上也未能成功。Dehnert、Hsu 和 Bratt [1989] 解释了 Cydrome Cydra 5 的架构和性能，这是一种具有动态寄存器重命名和软件流水线支持的超长指令字处理器。Cydra 5 是硬件和软件的独特结合，包括条件指令和寄存器旋转，旨在发掘 ILP。Cydrome 比 Multiflow 处理器依赖更多的硬件，并主要在向量风格代码上实现了有竞争力的性能。最终，Cydrome 遭遇了与 Multiflow 类似的问题，未能商业成功。尽管 Multiflow 和 Cydrome 作为商业实体未能成功，但培养了一批在利用 ILP 以及先进编译器技术方面经验丰富的人士；其中许多人将他们的经验和技术手段融入了新一代处理器。Fisher 和 Rau [1993] 编辑了一本涵盖这两个重要处理器的硬件和软件的综合论文集。

Rau 还开发了一种称为多循环调度的调度技术，这是大多数软件流水线方案的基础（see Rau, Glaeser and Picard [1982]）。Rau 的工作建立在 Davidson 和他的同事们关于流水线处理器的最佳硬件调度器设计的早期工作基础上。其他历史上的 LIW 处理器包括 Apollo DN 10000 和 Intel i860，这两者都可以双发射浮点和整数操作。

编译技术和硬件对指令调度的支持

循环级并行性和依赖性分析主要由 D. Kuck 和他在 1970 年代伊利诺伊大学的同事们开发。他们还创造了常用的术语“反依赖（anti-dependency）”和“输出依赖（output-dependency）”，并开发了几种标准依赖性测试，包括 GCD 和 Banerjee 测试。后者测试以 Uptal Banerjee 命名，并有多种变体。最近关于依赖性分析的工作集中在使用一系列精确测试，最终以一种称为 Fourier–Motzkin 的线性规划算法结束。D. Maydan 和 W. Pugh 都表明这些精确测试序列是一个实用的解决方案。

在揭示和调度 ILP 的领域，早期的许多工作与 VLIW 处理器的开发有关，如前所述。Lam [1988] 开发了软件流水线算法，并评估了其在 Warp（一种为特殊应用设计的超长指令字处理器）上的使用。Weiss 和 Smith [1987] 比较了软件流水线与循环展开作为在流水线处理器上进行代码调度的技术。Rau [1994] 开发了模调度，以处理软件流水线循环的问题，同时处理寄存器分配。

在各种情况下探索了对推测性代码调度的支持，包括一些处理器提供了一种忽略异常的模式，从而允许更积极地调度加载（例如，MIPS TFP 处理器 [Hsu 1994]）。几个小组探讨了对推测性代码调度进行更积极硬件支持的想法。例如，Smith、Horowitz 和 Lam [1992] 创造了一个称为 boosting 的概念，其中包含一个支持推测的硬件设施，但提供了一个检查和恢复机制，类似于 IA-64 和 Crusoe。哨兵调度（sentinel scheduling）的想法，也类似于 Crusoe 和 IA-64 架构中使用的推测和检查方法，是由伊利诺伊大学和 HP 实验室的研究人员共同开发的（see Mahlke et al. [1992]）。

在 1990 年代初，Wen-Mei Hwu 和他在伊利诺伊大学的同事们开发了一个编译器框架，称为 IMPACT（see Chang et al. [1991]），用于探索多发射架构与编译器技术之间的互动。该项目产生了几个重要的想法，包括超块调度（superblock scheduling）（see Hwu et al. [1993]）、广泛使用分析指导各种优化（例如，过程内联）以及使用一个特殊缓冲区（类似于 ALAT 或程序控制存储缓冲区）进行编译辅助的内存冲突检测（see Gallagher et al. [1994]）。他们还探讨了在 Mahlke 等人 [1995] 中部分和完全支持谓词化之间的性能权衡。

早期的 RISC 处理器都有延迟分支，这一方案受微程序设计启发，而几个关于编译时分支预测的研究受延迟分支机制启发。McFarling 和 Hennessy [1986] 对各种编译时和运行时分支预测方案进行了定量比较。Fisher 和 Freudenberger [1992] 使用错误预测之间的距离这一指标评估了一系列编译时分支预测方案。Ball 和 Larus [1993] 以及 Calder 等人 [1997] 描述了使用收集的程序行为的静态预测方案。

EPIC 架构与 IA-64 处理器的发展

EPIC（显式并行指令计算）方法可以追溯到早期构建 LIW 和 VLIW 机器的尝试，特别是 Cydrome 和 Multiflow 的项目，以及这些公司失败后在 HP、伊利诺伊大学及其他地方持续进行的编译器工作的悠久历史。从对这些工作的洞察中，HP 的设计师提出了一种 VLIW 风格的 64 位架构，以继承 HP PA RISC 架构。英特尔正在寻找一种新的架构来替代 x86（现称为 IA-32）架构并提供 64 位能力。1995 年，他们形成了一个合作伙伴关系，以设计一种新的架构，即 IA-64（see Huck et al. [2000]），并基于此构建处理器。Itanium（参见 Sharangpani 和 Arora [2000]）是第一个这样的处理器。2002 年，英特尔推出了第二代 IA-64 设计，即 Itanium 2（see McNairy and Soltis [2003] and McCormick and Knies [2002]）。

对 ILP 和提高 ILP 的研究

一系列早期的论文，包括 Tjaden 和 Flynn [1970] 以及 Riseman 和 Foster [1972]，得出结论认为，在不投入大量硬件的情况下，指令级别的并行性只能获

得很小的提升。这些论文在超过十年的时间里削弱了多指令发射的吸引力。Nicolau 和 Fisher [1984] 基于他们的轨迹调度工作发表了一篇论文，声称科学程序中存在大量潜在的指令级并行性（ILP）。

自那时起，关于可用 ILP 的研究有很多。这类研究受到批评，因为它们假设了一定程度的硬件支持和编译器技术。尽管如此，这些研究对于设定期望以及理解 ILP 限制的来源仍然有用。Wall 参与了几项此类研究，包括 Jouppi 和 Wall [1989] 以及 Wall [1991, 1993]。虽然早期的研究被批评为保守（例如，它们没有包括推测），但最后一项研究是迄今为止最雄心勃勃的 ILP 研究，并且是第 3.10 节数据的基础。Sohi 和 Vajapeyam [1989] 提供了对超长指令字处理器可用并行性的测量。Smith、Johnson 和 Horowitz [1989] 也使用推测性超标量处理器来研究 ILP 的限制。在他们研究时，他们预计他们所指定的处理器是合理设计的上限。然而，近期和即将推出的处理器很可能至少与他们的处理器一样具有雄心。

Skadron 等人 [1999] 研究了与 2005 年最先进的处理器相当的处理器中的性能权衡和限制，得出结论认为，如果没有在整数程序的分支预测上取得显著改进，较大的窗口尺寸将没有意义。

Lam 和 Wilson [1992] 研究了推测执行带来的限制，并表明通过允许处理器在多个方向上进行推测可以获得额外的收益，这需要多个程序计数器（PC）。这种方案无法超越完美推测的成就，但它们有助于缩小现实预测方案与完美预测之间的差距。Wall 1993 年的研究包括对此方法的有限评估（最多探索八个分支）。

突破数据流的局限

在文献中探索的另一种方法是利用值预测（value prediction）。值预测可以基于数据值进行推测。关于值预测的使用已经有许多研究。Lipasti 和 Shen 在 1996 年发表了两篇论文，评估了值预测的概念及其对 ILP 利用的潜在影响。Calder、Reinman 和 Tullsen [1999] 探索了选择性值预测的想法。Sodani 和 Sohi [1997] 从重用指令生成的值的角度解决了同样的问题。Moshovos 等人 [1997] 表明，决定何时对值进行推测，通过跟踪过去这种推测是否准确，对于通过值推测实现性能提升是重要的。Moshovos 和 Sohi [1997] 以及 Chrysos 和 Emer [1998] 关注预测内存依赖性并利用这些信息通过内存消除依赖性。González 和 González [1998]、Babbay 和 Mendelson [1998] 以及 Calder、Reinman 和 Tullsen [1999] 是关于值预测使用的较新研究。该领域目前非常活跃，每次会议都有新的研究结果发表。

近期的先进处理器

1994 年和 1995 年，各大处理器厂商纷纷宣布推出宽超标量处理器（每个时钟周期发出三条或更多指令）：Intel 的 Pentium Pro 和 Pentium II（这两款处理器共享相同的核心流水线架构，详见 Colwell 和 Steck [1995]）；AMD 的 K-5、K-6 和 Athlon；Sun 的 UltraSPARC（参见 Lauterbach 和 Horel [1999]）；Alpha 21164（参见 Edmondson 等人 [1995]）和 21264（参见 Kessler [1999]）；MIPS R10000 和 R12000（参见 Yeager [1996]）；PowerPC 603、604 和 620（参见 Diep、Nelson 和 Shen [1995]）；以及 HP 8000（Kumar [1997]）。在十年的后期（1996–2000 年），

许多这些处理器推出了第二代产品（如 Pentium III、AMD Athlon 和 Alpha 21264 等）。虽然第二代产品的发出率相似，但它们能够维持更低的 CPI，并提供更高的时钟频率。所有这些处理器都包含动态调度，并几乎普遍支持推测执行。在实际应用中，许多因素，包括实现技术、内存层次结构、设计者的技能以及应用程序的基准测试类型，都会影响哪种方法是最佳的。

2000 年至 2005 年期间，超标量处理器的三个趋势占据主导地位：通过更深的流水线实现更高时钟频率的引入（例如在 Pentium 4 中，see Hintonet et al. [2001]），IBM 在 Power 4 和 Intel 在 Pentium 4 Extreme 中引入多线程技术，以及 IBM 在 Power 4、AMD 在 Opteron（see Keltcher et al. [2003]），以及最近 Intel（see Douglas [2005]）开始的多核化趋势。

多线程和同时多线程

多线程的概念可以追溯到最早的晶体管计算机之一——TX-2。TX-2 还因 Ivan Sutherland 在其上创建了 Sketchpad（第一个计算机图形系统）而闻名。TX-2 是在麻省理工学院的林肯实验室建造的，并于 1959 年投入使用。它使用多线程来支持快速的上下文切换以处理 I/O 功能。Clark [1957]描述了基本架构，Forgie [1957]描述了 I/O 架构。多线程还被用于 CDC 6600 中，其中一种细粒度多线程（fine-grained multithreading）方案通过线程间的交错调度用于 I/O 处理器的架构。HEP 处理器是由 Denelcor 设计并于 1982 年推出的一种流水线多处理器，使用细粒度多线程来隐藏流水线延迟以及隐藏在所有处理器之间共享的大内存的延迟。由于 HEP 没有缓存，隐藏内存延迟是至关重要的。主要架构师之一 Burton Smith 在 1978 年的一篇论文中描述了 HEP 架构，Jordan [1983]发布了性能评估。TERA 处理器扩展了多线程的思想，Alverson 等人在 1992 年的一篇论文中进行了描述。Niagara 多线程方法类似于 HEP 和 TERA 系统，尽管 Niagara 采用了缓存，减少了基于线程的延迟隐藏的需求。

在 20 世纪 80 年代末和 90 年代初，研究人员探索了粗粒度多线程（coarse-grained multithreading）（也称为块多线程）的概念，作为一种掩盖延迟的方法，特别是在多处理器环境中。Alewife 系统中的 SPARCLE 处理器使用了这种方案，每当发生高延迟的异常事件（如长时间的缓存未命中）时就切换线程。Agarwal 等人在 1993 年的一篇论文中描述了 SPARCLE。IBM 的 Pulsar 处理器采用了类似的思想。

到 20 世纪 90 年代初，几个研究小组得出了两个关键见解。首先，他们意识到需要细粒度多线程来获得最大性能收益，因为在粗粒度方法中，线程切换和线程启动（例如，从新线程填充流水线）的开销抵消了大部分性能优势（参见 Laudon, Gupta, 和 Horowitz [1994]）。其次，几个小组意识到，为了有效利用大量功能单元，需要同时使用指令级并行（ILP）和线程级并行（TLP）。这些见解导致了几种结合多线程和多发射的架构。Wolfe 和 Shen [1991]描述了一种称为 XIMD 的架构，该架构静态交错为 VLIW 处理器调度的线程。Hirata 等人 [1992]描述了一种用于媒体的协处理器，该处理器结合了静态超标量流水线和多线程支持；他们报告了结合这两种并行形式所带来的加速效果。Keckler 和 Dally [1992]为具有多个功能单元的处理器结合了 ILP 的静态调度和线程的动态

调度。如何在 ILP 和 TLP 之间分配功能单元以及如何调度这两种并行形式的问题仍然没有解决。

当在20世纪90年代中期动态调度的超标量即将问世时，几个研究小组提议使用动态调度能力来实时混合来自多个线程的指令。Yamamoto 等人 [1994]似乎首次提出了这样的建议，尽管其多线程超标量架构的模拟结果使用了简化的假设。这项工作很快被 Tullsen, Eggers, 和 Levy [1995]跟进，他们提供了第一个现实的模拟评估，并创造了“同时多线程”这一术语。同一小组与工业合作者的后续工作解决了关于 SMT 的许多悬而未决的问题。例如，Tullsen 等人 [1996]解决了关于调度 ILP 与 TLP 的挑战问题。Lo 等人 [1997]对 SMT 概念进行了广泛讨论并评估了其性能潜力，Lo 等人 [1998]评估了 SMT 处理器上的数据库性能。Tuck 和 Tullsen [2003]回顾了 Pentium 4 上的 SMT 性能。

IBM Power4 引入了多线程（参见 Tendler 等人 [2002]），而 Power5 使用了同时多线程。Mathis 等人 [2005]探讨了 Power5 中 SMT 的性能，而 Sinharoy 等人 [2005]描述了系统架构。

参考文献

Agarwal, A., J. Kubiatowicz, D. Kranz, B.-H. Lim, D. Yeung, G. D’Souza, and M. Parkin [1993]. “Sparcle: An evolutionary processor design for large-scale multiprocessors,” IEEE Micro 13 (June), 48–61.

Agerwala, T., and J. Cocke [1987]. High Performance Reduced Instruction Set Processors, Tech. Rep. RC12434, IBM Thomas Watson Research Center, Yorktown Heights, N.Y.

Alverson, G., R. Alverson, D. Callahan, B. Koblenz, A. Porterfield, and B. Smith [1992]. “Exploiting heterogeneous parallelism on a multithreaded multiprocessor,” Proc. ACM/IEEE Conf. on Supercomputing, November 16–20, 1992, Minneapolis, Minn., 188–197.

Anderson, D. W., F. J. Sparacio, and R. M. Tomasulo [1967]. “The IBM 360 Model 91: Processor philosophy and instruction handling,” IBM J. Research and Development 11:1 (January), 8–24.

Austin, T. M., and G. Sohi [1992]. “Dynamic dependency analysis of ordinary programs,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 342–351.

Babbay, F., and A. Mendelson [1998]. “Using value prediction to increase the power of speculative execution hardware,” ACM Trans. on Computer Systems 16:3 (August), 234–270.

Bakoglu, H. B., G. F. Grohoski, L. E. Thatcher, J. A. Kaeli, C. R. Moore, D. P. Tattle, W. E. Male, W. R. Hardell, D. A. Hicks, M. Nguyen Phu, R. K. Montoye, W. T. Glover, and S. Dhawan [1989]. “IBM second-generation RISC processor organization,” Proc. IEEE Int’l. Conf. on Computer Design, October, Rye Brook, N.Y., 138–142.

Ball, T., and J. Larus [1993]. “Branch prediction for free,” Proc. ACM SIGPLAN’93 Conference on Programming Language Design and Implementation (PLDI), June 23–25, 1993, Albuquerque, N.M., 300–313.

Bhandarkar, D., and D. W. Clark [1991]. "Performance from architecture: Comparing a RISC and a CISC with similar hardware organizations," Proc. Fourth Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 8–11, 1991, Palo Alto, Calif., 310–319.

Bhandarkar, D., and J. Ding [1997]. "Performance characterization of the Pentium Pro processor," Proc. Third Int'l. Symposium on High Performance Computer Architecture, February 1–5, 1997, San Antonio, Tex., 288–297.

Bloch, E. [1959]. "The engineering design of the Stretch computer," Proc. Eastern Joint Computer Conf., December 1–3, 1959, Boston, Mass., 48–59.

Bucholtz, W. [1962]. Planning a Computer System: Project Stretch, McGraw-Hill, New York.

Calder, B., D. Grunwald, M. Jones, D. Lindsay, J. Martin, M. Mozer, and B. Zorn [1997]. "Evidence-based static branch prediction using machine learning," ACM Trans. Program. Lang. Syst. 19:1, 188–222.

Calder, B., G. Reinman, and D. M. Tullsen [1999]. "Selective value prediction," Proc. 26th Annual Int'l. Symposium on Computer Architecture (ISCA), May 2–4, 1999, Atlanta, Ga.

Chang, P. P., S. A. Mahlke, W. Y. Chen, N. J. Warter, and W. W. Hwu [1991]. "IMPACT: An architectural framework for multiple-instruction-issue processors," Proc. 18th Annual Int'l. Symposium on Computer Architecture (ISCA), May 27–30, 1991, Toronto, Canada, 266–275.

Charlesworth, A. E. [1981]. "An approach to scientific array processing: The architecture design of the AP-120B/FPS-164 family," Computer 14:9 (September), 18–27.

Chen, T. C. [1980]. "Overlap and parallel processing," in Introduction to Computer Architecture, H. Stone, ed., Science Research Associates, Chicago, 427–486.

Chrysos, G. Z., and J. S. Emer [1998]. "Memory dependence prediction using store sets," Proc. 25th Annual Int'l. Symposium on Computer Architecture (ISCA), July 3–14, 1998, Barcelona, Spain, 142–153.

Clark, D. W. [1987]. "Pipelining and performance in the VAX 8800 processor," Proc. Second Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 5–8, 1987, Palo Alto, Calif., 173–177.

Clark, W. A. [1957]. "The Lincoln TX-2 computer development," Proc. Western Joint Computer Conference, February 26–28, 1957, Los Angeles, 143–145.

Colwell, R. P., and R. Steck [1995]. "A 0.6 μm BiCMOS processor with dynamic execution." Proc. of IEEE Int'l. Symposium on Solid State Circuits (ISSCC), February 15–17, 1995, San Francisco, 176–177.

Colwell, R. P., R. P. Nix, J. J. O'Donnell, D. B. Papworth, and P. K. Rodman [1987]. "A VLIW architecture for a trace scheduling compiler," Proc. Second Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 5–8, 1987, Palo Alto, Calif., 180–192.

Cvetanovic, Z., and R. E. Kessler [2000]. "Performance analysis of the Alpha 21264-based Compaq ES40 system," 27th Annual Int'l. Symposium on

Computer Architecture (ISCA), June 10–14, 2000, Vancouver, Canada, 192–202.

Davidson, E. S. [1971]. "The design and control of pipelined function generators," Proc. IEEE Conf. on Systems, Networks, and Computers, January 19–21, 1971, Oaxtepec, Mexico, 19–21.

Davidson, E. S., A. T. Thomas, L. E. Shar, and J. H. Patel [1975]. "Effective control for pipelined processors," Proc. IEEE COMPCON, February 25–27, 1975, San Francisco, 181–184.

Dehnert, J. C., P. Y.-T. Hsu, and J. P. Bratt [1989]. "Overlapped loop support on the Cydra 5," Proc. Third Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 3–6, 1989, Boston, Mass., 26–39.

Diep, T. A., C. Nelson, and J. P. Shen [1995]. "Performance evaluation of the PowerPC 620 microarchitecture," Proc. 22nd Annual Int'l. Symposium on Computer Architecture (ISCA), June 22–24, 1995, Santa Margherita, Italy.

Ditzel, D. R., and H. R. McLellan [1987]. "Branch folding in the CRISP microprocessor: Reducing the branch delay to zero," Proc. 14th Annual Int'l. Symposium on Computer Architecture (ISCA), June 2–5, 1987, Pittsburgh, Penn., 2–7.

Douglas, J. [2005]. "Intel 8xx series and Paxville Xeon-MP Microprocessors," paper presented at Hot Chips 17, August 14–16, 2005, Stanford University, Palo Alto, Calif.

Eden, A., and T. Mudge [1998]. "The YAGS branch prediction scheme," Proc. of the 31st Annual ACM/IEEE Int'l. Symposium on Microarchitecture, November 30–December 2, 1998, Dallas, Tex., 69–80.

Edmondson, J. H., P. I. Rubinfield, R. Preston, and V. Rajagopalan [1995]. "Superscalar instruction execution in the 21164 Alpha microprocessor," IEEE Micro 15:2, 33–43.

Ellis, J. R. [1986]. Bulldog: A Compiler for VLIW Architectures, MIT Press, Cambridge, Mass.

Emer, J. S., and D. W. Clark [1984]. "A characterization of processor performance in the VAX-11/780," Proc. 11th Annual Int'l. Symposium on Computer Architecture (ISCA), June 5–7, 1984, Ann Arbor, Mich., 301–310.

Evers, M., S. J. Patel, R. S. Chappell, and Y. N. Patt [1998]. "An analysis of correlation and predictability: What makes two-level branch predictors work," Proc. 25th Annual Int'l. Symposium on Computer Architecture (ISCA), July 3–14, 1998, Barcelona, Spain, 52–61.

Fisher, J. A. [1981]. "Trace scheduling: A technique for global microcode compaction," IEEE Trans. on Computers 30:7 (July), 478–490.

Fisher, J. A. [1983]. "Very long instruction word architectures and ELI-512," 10th Annual Int'l. Symposium on Computer Architecture (ISCA), June 5–7, 1982, Stockholm, Sweden, 140–150.

Fisher, J. A., and S. M. Freudenberger [1992]. "Predicting conditional branches from previous runs of a program," Proc. Fifth Int'l. Conf. on Architectural

Support for Programming Languages and Operating Systems (ASPLOS), October 12–15, 1992, Boston, 85–95.

Fisher, J. A., and B. R. Rau [1993]. Journal of Supercomputing, January (special issue).

Fisher, J. A., J. R. Ellis, J. C. Ruttenberg, and A. Nicolau [1984]. "Parallel processing: A smart compiler and a dumb processor," Proc. SIGPLAN Conf. on Compiler Construction, June 17–22, 1984, Montreal, Canada, 11–16.

Forgie, J. W. [1957]. "The Lincoln TX-2 input-output system," Proc. Western Joint Computer Conference, February 26–28, 1957, Los Angeles, 156–160.

Foster, C. C., and E. M. Riseman [1972]. "Percolation of code to enhance parallel dispatching and execution," IEEE Trans. on Computers C-21:12 (December), 1411–1415.

Gallagher, D. M., W. Y. Chen, S. A. Mahlke, J. C. Gyllenhaal, and W.W. Hwu [1994]. "Dynamic memory disambiguation using the memory conflict buffer," Proc. Sixth Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 4–7, Santa Jose, Calif., 183–193.

González, J., and A. González [1998]. "Limits of instruction level parallelism with data speculation," Proc. Vector and Parallel Processing (VECPAR) Conf., June 21–23, 1998, Porto, Portugal, 585–598.

Heinrich, J. [1993]. MIPS R4000 User's Manual, Prentice Hall, Englewood Cliffs, N.J.

Hinton, G., D. Sager, M. Upton, D. Boggs, D. Carmean, A. Kyker, and P. Roussel [2001]. "The microarchitecture of the Pentium 4 processor," Intel Technology Journal, February.

Hirata, H., K. Kimura, S. Nagamine, Y. Mochizuki, A. Nishimura, Y. Nakase, and T. Nishizawa [1992]. "An elementary processor architecture with simultaneous instruction issuing from multiple threads," Proc. 19th Annual Int'l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 136–145.

Hopkins, M. [2000]. "A critical look at IA-64: Massive resources, massive ILP, but can it deliver?" Microprocessor Report, February.

Hsu, P. [1994]. "Designing the TFP microprocessor," IEEE Micro 18:2 (April), 2333.

Huck, J. et al. [2000]. "Introducing the IA-64 Architecture" IEEE Micro, 20:5 (September–October), 12–23.

Hwu, W.-M., and Y. Patt [1986]. "HPSm, a high performance restricted data flow architecture having minimum functionality," 13th Annual Int'l. Symposium on Computer Architecture (ISCA), June 2–5, 1986, Tokyo, 297–307.

Hwu, W. W., S. A. Mahlke, W. Y. Chen, P. P. Chang, N. J. Warter, R. A. Bringmann, R. O. Ouellette, R. E. Hank, T. Kiyohara, G. E. Haab, J. G. Holm, and D. M. Lavery [1993]. "The superblock: An effective technique for VLIW and superscalar compilation," J. Supercomputing 7:1, 2 (March), 229–248.

IBM. [1990]. "The IBMRISC System/6000 processor" (collection of papers), IBM J. Research and Development 34:1 (January).

Jimenez, D. A., and C. Lin [2002]. “Neural methods for dynamic branch prediction,” ACM Trans. Computer Sys 20:4 (November), 369–397.

Johnson, M. [1990]. Superscalar Microprocessor Design, Prentice Hall, Englewood Cliffs, N.J.

Jordan, H. F. [1983]. “Performance measurements on HEP—a pipelined MIMD computer,” Proc. 10th Annual Int’l. Symposium on Computer Architecture (ISCA), June 5–7, 1982, Stockholm, Sweden, 207–212.

Jouppi, N. P., and D. W. Wall [1989]. “Available instruction-level parallelism for superscalar and superpipelined processors,” Proc. Third Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 3–6, 1989, Boston, 272–282.

Kaeli, D. R., and P. G. Emma [1991]. “Branch history table prediction of moving target branches due to subroutine returns,” Proc. 18th Annual Int’l. Symposium on Computer Architecture (ISCA), May 27–30, 1991, Toronto, Canada, 34–42.

Keckler, S. W., and W. J. Dally [1992]. “Processor coupling: Integrating compile time and runtime scheduling for parallelism,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 202–213.

Keller, R. M. [1975]. “Look-ahead processors,” ACM Computing Surveys 7:4 (December), 177–195.

Keltcher, C. N., K. J. McGrath, A. Ahmed, and P. Conway [2003]. “The AMD Opteron processor for multiprocessor servers,” IEEE Micro 23:2 (March– April), 66–76.

Kessler, R. [1999]. “The Alpha 21264 microprocessor,” IEEE Micro 19:2 (March/April) 24–36.

Killian, E. [1991]. “MIPS R4000 technical overview–64 bits/100 MHz or bust,” Hot Chips III Symposium Record, August 26–27, 1991, Stanford University, Palo Alto, Calif., 1.6–1.19.

Kogge, P. M. [1981]. The Architecture of Pipelined Computers, McGraw-Hill, New York.

Kumar, A. [1997]. “The HP PA-8000 RISC CPU,” IEEE Micro 17:2 (March/April).

Kunkel, S. R., and J. E. Smith [1986]. “Optimal pipelining in supercomputers,” Proc. 13th Annual Int’l. Symposium on Computer Architecture (ISCA), June 2–5, 1986, Tokyo, 404–414.

Lam, M. [1988]. “Software pipelining: An effective scheduling technique for VLIW processors,” SIGPLAN Conf. on Programming Language Design and Implementation, June 22–24, 1988, Atlanta, Ga., 318–328.

Lam, M. S., and R. P. Wilson [1992]. “Limits of control flow on parallelism,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 46–57.

Laudon, J., A. Gupta, and M. Horowitz [1994]. “Interleaving: A multithreading technique targeting multiprocessors and workstations,” Proc. Sixth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 4–7, San Jose, Calif., 308–318.

Lauterbach, G., and T. Horel [1999]. "UltraSPARC-III: Designing third generation 64-bit performance," IEEE Micro 19:3 (May/June).

Lipasti, M. H., and J. P. Shen [1996]. "Exceeding the dataflow limit via value prediction," Proc. 29th Int'l. Symposium on Microarchitecture, December 2–4, 1996, Paris, France.

Lipasti, M. H., C. B. Wilkerson, and J. P. Shen [1996]. "Value locality and load value prediction," Proc. Seventh Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 1–5, 1996, Cambridge, Mass., 138–147.

Lo, J., L. Barroso, S. Eggers, K. Gharachorloo, H. Levy, and S. Parekh [1998]. "An analysis of database workload performance on simultaneous multithreaded processors," Proc. 25th Annual Int'l. Symposium on Computer Architecture (ISCA), July 3–14, 1998, Barcelona, Spain, 39–50.

Lo, J., S. Eggers, J. Emer, H. Levy, R. Stamm, and D. Tullsen [1997]. "Converting thread-level parallelism into instruction-level parallelism via simultaneous multithreading," ACM Trans. on Computer Systems 15:2 (August), 322–354.

Mahlke, S. A., W. Y. Chen, W.-M. Hwu, B. R. Rau, and M. S. Schlansker [1992]. "Sentinel scheduling for VLIW and superscalar processors," Proc. Fifth Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 12–15, 1992, Boston, 238–247.

Mahlke, S. A., R. E. Hank, J. E. McCormick, D. I. August, and W. W. Hwu [1995]. "A comparison of full and partial predicated execution support for ILP processors," Proc. 22nd Annual Int'l. Symposium on Computer Architecture (ISCA), June 22–24, 1995, Santa Margherita, Italy, 138–149.

Mathis, H. M., A. E. Mercias, J. D. McCalpin, R. J. Eickemeyer, and S. R. Kunkel [2005]. "Characterization of the multithreading (SMT) efficiency in Power5," IBM J. of Research and Development, 49:4/5 (July/September), 555–564.

McCormick, J., and A. Knies [2002]. "A brief analysis of the SPEC CPU2000 benchmarks on the Intel Itanium 2 processor," paper presented at Hot Chips 14, August 18–20, 2002, Stanford University, Palo Alto, Calif.

McFarling, S. [1993]. Combining Branch Predictors, WRL Technical Note TN-36, Digital Western Research Laboratory, Palo Alto, Calif.

McFarling, S., and J. Hennessy [1986]. "Reducing the cost of branches," Proc. 13th Annual Int'l. Symposium on Computer Architecture (ISCA), June 2–5, 1986, Tokyo, 396–403.

McNairy, C., and D. Soltis [2003]. "Itanium 2 processor microarchitecture," IEEE Micro 23:2 (March–April), 44–55.

Moshovos, A., and G. S. Sohi [1997]. "Streamlining inter-operation memory communication via data dependence prediction," Proc. 30th Annual Int'l. Symposium on Microarchitecture, December 1–3, Research Triangle Park, N.C., 235–245.

Moshovos, A., S. Breach, T. N. Vijaykumar, and G. S. Sohi [1997]. "Dynamic speculation and synchronization of data dependences," Proc. 24th Annual Int'l. Symposium on Computer Architecture (ISCA), June 2–4, 1997, Denver, Colo.

Nicolau, A., and J. A. Fisher [1984]. “Measuring the parallelism available for very long instruction word architectures,” IEEE Trans. on Computers C-33:11 (November), 968–976.

Pan, S.-T., K. So, and J. T. Rameh [1992]. “Improving the accuracy of dynamic branch prediction using branch correlation,” Proc. Fifth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 12–15, 1992, Boston, 76–84.

Postiff, M.A., D. A. Greene, G. S. Tyson, and T. N. Mudge [1999]. “The limits of instruction level parallelism in SPEC95 applications,” Computer Architecture News 27:1 (March), 31–40.

Ramamoorthy, C. V., and H. F. Li [1977]. “Pipeline architecture,” ACM Computing Surveys 9:1 (March), 61–102.

Rau, B. R. [1994]. “Iterative modulo scheduling: An algorithm for software pipelining loops,” Proc. 27th Annual Int’l. Symposium on Microarchitecture, November 30–December 2, 1994, San Jose, Calif., 63–74.

Rau, B. R., C. D. Glaeser, and R. L. Picard [1982]. “Efficient code generation for horizontal architectures: Compiler techniques and architectural support,” Proc. Ninth Annual Int’l. Symposium on Computer Architecture (ISCA), April 26–29, 1982, Austin, Tex., 131–139.

Rau, B. R., D. W. L. Yen, W. Yen, and R. A. Towle [1989]. “The Cydra 5 departmental supercomputer: Design philosophies, decisions, and trade-offs,” IEEE Computers 22:1 (January), 12–34.

Riseman, E. M., and C. C. Foster [1972]. “Percolation of code to enhance paralleled dispatching and execution,” IEEE Trans. on Computers C-21:12 (December), 1411–1415.

Rymarczyk, J. [1982]. “Coding guidelines for pipelined processors,” Proc. Symposium Architectural Support for Programming Languages and Operating Systems (ASPLOS), March 1–3, 1982, Palo Alto, Calif., 12–19.

Sharangpani, H., and K. Arora [2000]. “Itanium Processor Microarchitecture,” IEEE Micro, 20:5 (September–October), 24–43.

Sinharoy, B., R. N. Koala, J. M. Tendler, R. J. Eickemeyer, and J. B. Joyner [2005]. “POWER5 system microarchitecture,” IBM J. of Research and Development, 49:4–5, 505–521.

Sites, R. [1979]. Instruction Ordering for the CRAY-1 Computer, Tech. Rep. 78-CS-023, Dept. of Computer Science, University of California, San Diego.

Skadron, K., P. S. Ahuja, M. Martonosi, and D. W. Clark [1999]. “Branch prediction, instruction-window size, and cache size: Performance tradeoffs and simulation techniques,” IEEE Trans. on Computers, 48:11 (November).

Smith, A., and J. Lee [1984]. “Branch prediction strategies and branch-target buffer design,” Computer 17:1 (January), 6–22.

Smith, B. J. [1978]. “A pipelined, shared resource MIMD computer,” Proc. Int’l. Conf. on Parallel Processing (ICPP), August, Bellaire, Mich., 6–8.

Smith, J. E. [1981]. “A study of branch prediction strategies,” Proc. Eighth Annual Int’l. Symposium on Computer Architecture (ISCA), May 12–14, 1981, Minneapolis, Minn., 135–148.

Smith, J. E. [1984]. “Decoupled access/execute computer architectures,” ACM Trans. on Computer Systems 2:4 (November), 289–308.

Smith, J. E. [1989]. “Dynamic instruction scheduling and the Astronautics ZS-1,” Computer 22:7 (July), 21–35.

Smith, J. E., and A. R. Pleszkun [1988]. “Implementing precise interrupts in pipelined processors,” IEEE Trans. on Computers 37:5 (May), 562–573. (This paper is based on an earlier paper that appeared in Proc. 12th Annual Int’l. Symposium on Computer Architecture (ISCA), June 17–19, 1985, Boston, Mass.)

Smith, J. E., G. E. Dermer, B. D. Vanderwarn, S. D. Klinger, C. M. Rozewski, D. L. Fowler, K. R. Scidmore, and J. P. Laudon [1987]. “The ZS-1 central processor,” Proc. Second Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 5–8, 1987, Palo Alto, Calif., 199–204.

Smith, M. D., M. Horowitz, and M. S. Lam [1992]. “Efficient superscalar performance through boosting,” Proc. Fifth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 12–15, 1992, Boston, 248–259.

Smith, M. D., M. Johnson, and M. A. Horowitz [1989]. “Limits on multiple instruction issue,” Proc. Third Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 3–6, 1989, Boston, 290–302.

Sodani, A., and G. Sohi [1997]. “Dynamic instruction reuse,” Proc. 24th Annual Int’l. Symposium on Computer Architecture (ISCA), June 2–4, 1997, Denver, Colo.

Sohi, G. S. [1990]. “Instruction issue logic for high-performance, interruptible, multiple functional unit, pipelined computers,” IEEE Trans. on Computers 39:3 (March), 349–359.

Sohi, G. S., and S. Vajapeyam [1989]. “Tradeoffs in instruction format design for horizontal architectures,” Proc. Third Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 3–6, 1989, Boston, 15–25.

Sussenguth, E. [1999]. “IBM’s ACS-1 Machine,” IEEE Computer 22:11 (November).

Tendler, J. M., J. S. Dodson, J. S. Fields, Jr., H. Le, and B. Sinharoy [2002]. “Power4 system microarchitecture,” IBM J. of Research and Development, 46:1, 5–26.

Thorlin, J. F. [1967]. “Code generation for PIE (parallel instruction execution) computers,” Proc. Spring Joint Computer Conf., April 18–20, 1967, Atlantic City, N.J., 27.

Thornton, J. E. [1964]. “Parallel operation in the Control Data 6600,” Proc. AFIPS Fall Joint Computer Conf., Part II, October 27–29, 1964, San Francisco, 26, 33–40.

Thornton, J. E. [1970]. Design of a Computer, the Control Data 6600, Scott, Foresman, Glenview, Ill.

Tjaden, G. S., and M. J. Flynn [1970]. “Detection and parallel execution of independent instructions,” IEEE Trans. on Computers C-19:10 (October), 889–895.

Tomasulo, R. M. [1967]. “An efficient algorithm for exploiting multiple arithmetic units,” IBM J. Research and Development 11:1 (January), 25–33.

Tuck, N., and D. Tullsen [2003]. “Initial observations of the simultaneous multithreading Pentium 4 processor,” Proc. 12th Int. Conf. on Parallel Architectures and Compilation Techniques (PACT’03), September 27–October 1, New Orleans, La., 26–34.

Tullsen, D. M., S. J. Eggers, and H. M. Levy [1995]. “Simultaneous multithreading: Maximizing on-chip parallelism,” Proc. 22nd Annual Int’l. Symposium on Computer Architecture (ISCA), June 22–24, 1995, Santa Margherita, Italy, 392–403.

Tullsen, D. M., S. J. Eggers, J. S. Emer, H. M. Levy, J. L. Lo, and R. L. Stamm [1996]. “Exploiting choice: Instruction fetch and issue on an implementable simultaneous multithreading processor,” Proc. 23rd Annual Int’l. Symposium on Computer Architecture (ISCA), May 22–24, 1996, Philadelphia, Penn., 191–202.

Wall, D. W. [1991]. “Limits of instruction-level parallelism,” Proc. Fourth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 8–11, 1991, Palo Alto, Calif., 248–259.

Wall, D. W. [1993]. Limits of Instruction-Level Parallelism, Research Rep. 93/6, Western Research Laboratory, Digital Equipment Corp., Palo Alto, Calif.

Weiss, S., and J. E. Smith [1984]. “Instruction issue logic for pipelined supercomputers,” Proc. 11th Annual Int’l. Symposium on Computer Architecture (ISCA), June 5–7, 1984, Ann Arbor, Mich., 110–118.

Weiss, S., and J. E. Smith [1987]. “A study of scalar compilation techniques for pipelined supercomputers,” Proc. Second Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 5–8, 1987, Palo Alto, Calif., 105–109.

Wilson, R. P., and M. S. Lam [1995]. “Efficient context-sensitive pointer analysis for C programs,” Proc. ACM SIGPLAN’95 Conf. on Programming Language Design and Implementation, June 18–21, 1995, La Jolla, Calif., 1–12.

Wolfe, A., and J. P. Shen [1991]. “A variable instruction stream extension to the VLIW architecture,” Proc. Fourth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 8–11, 1991, Palo Alto, Calif., 2–14.

Yamamoto, W., M. J. Serrano, A. R. Talcott, R. C. Wood, and M. Nemirosky [1994]. “Performance estimation of multistreamed, superscalar processors,” Proc. 27th Annual Hawaii Int’l. Conf. on System Sciences, January 4–7, 1994, Maui, 195–204.

Yeager, K. [1996]. “The MIPS R10000 superscalar microprocessor,” IEEE Micro 16:2 (April), 28–40.

Yeh, T., and Y. N. Patt [1992]. “Alternative implementations of two-level adaptive branch prediction,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 124–134.

Yeh, T., and Y. N. Patt [1993]. "A comparison of dynamic branch predictors that use two levels of branch history," Proc. 20th Annual Int'l. Symposium on Computer Architecture (ISCA), May 16–19, 1993, San Diego, Calif., 257–266.

M.6 SIMD 超级计算机、向量计算机、多媒体 SIMD 指令扩展和图形处理器的发展

在这一历史部分，我们首先介绍也许是最臭名昭著的超级计算机——Illiac IV，它是早期 SIMD（单指令流多数据流）架构的代表。接着，我们转向也许是最著名的超级计算机——Cray-1，它是向量架构的代表。下一步是多媒体 SIMD 扩展，其名称部分来源于一场广告活动，该活动涉及"Bunny People"，一群在半导体生产线上穿着洁净服跳迪斯科舞的工人。最后，我们介绍 GPU 的历史，虽然没有前面的那么引人注目。

SIMD 超级计算机

然而，通用多处理器的成本非常高，因此考虑了进一步的设计选项，以在不严重降低系统性能或效率的情况下降低成本。这些选项包括重新集中三大主要组件之一……将[控制单元]集中化形成了基本的组织结构，比如 Illiac IV 这样的阵列处理器。

Bouknight 等[1972]

……对于 Illiac IV 来说，编程这台机器是非常困难的，而且这种架构可能并不太适合我们尝试运行的一些应用程序。关键的观点是，我认为在 Illiac IV 中，应用程序和架构之间并没有很好地匹配。

David Kuck

Illiac IV 的软件设计师和平行软件的早期先驱

David Kuck 在 1991 年接受 Andrew Goldstein 采访时的口述历史

IEEE 历史中心，新泽西州新不伦瑞克

SIMD 模型是最早的并行计算模型之一，可以追溯到第一个大规模多处理器 Illiac IV。与向量架构中数据计算流水线化不同，这些机器拥有功能单元阵列，因此可以被视为阵列处理器。关于 SIMD 风格计算机的最早想法来自 Unger [1958]和 Slotnick、Borck、McReynolds [1962]。Slotnick 设计的 Solomon 成为 Illiac IV 的基础，可能是超级计算机项目中最臭名昭著的一个。虽然在推动一些后来证明有用的技术方面取得了成功，但作为计算机项目却失败了。其成本从 1966 年的 800 万美元预算上涨到 1972 年的 3100 万美元，还只建造了计划中的四分之一的多处理器。（按 2011 年的美元计算，这一增长相当于从 5400 万美元到 1.52 亿美元。）实际性能最多只有 15 MFLOPS，而完整系统的初始预测为 1000 MFLOPS [Hord 1982]。1972 年交付给 NASA Ames 研究中心后，计算机还需要三年的工程改进才能投入使用。这些事件使得对 SIMD 的研究放缓，但 Danny Hillis [1985] 在 Connection Machine 中复兴了这种风格，该机器拥有 65,536 个 1 位处理器。

• 原文是：*The cost of a general multiprocessor is, however, very high and further design options were considered which would decrease the cost without seriously degrading the power or efficiency of the system. The options consist of recentralizing one of the three major components. …Centralizing the [control unit] gives rise to the basic organization of [an] … array processor such as the Illiac IV. Bouknight et al. [1972]*

• 原文是：*…with Iliac IV, programming the machine was very difficult and the architecture probably was not very well suited to some of the applications we were trying to run. The key idea was that I did not think we had a very good match in Iliac IV between applications and architecture.*

David Kuck

Software designer for the Illiac IV and early pioneer in parallel software

David Kuck

An oral history conducted in 1991 by Andrew Goldstein, IEEE History Center, New Brunswick, N.J.

SIMD 多处理器的基本权衡是处理器性能与处理器数量之间的权衡。1980 年代的 SIMD 超级计算机强调高度的并行性，而不是单个处理器的性能。例如，Connection Multiprocessor 2 提供了 65,536 个单位宽处理器，而 Illiac IV 计划使用 64 个 64 位处理器。大规模并行 SIMD 多处理器依赖互连或通信网络在处理单元之间交换数据。

在 1980 年代首先由 Thinking Machines 和 MasPar 复兴后，SIMD 模型作为超级计算机逐渐消失，主要有两个原因。首先，它过于僵化。许多重要问题并不是数据并行的，而且这种架构在缩小规模时不能以竞争的方式扩展，即小规模 SIMD 多处理器的成本性能往往比替代方案更差。其次，SIMD 无法利用 1980 年代 SISD（单指令单数据）微处理器技术的巨大性能和成本优势，该技术每 18 个月性能翻倍。SIMD 多处理器的设计者不得不为他们的多处理器构建定制处理器，而不是利用这种低成本技术。

向量计算机

我当然不是在发明向量处理器。我知道现在存在三种类型的向量处理器。它们分别是 Illiac-IV、(CDC) Star 处理器和 TI (ASC)处理器。这三者都是开创性的处理器……成为先锋的一个问题是你总会犯错，而我绝对不想成为先锋。最好是成为第二个，这样可以看到先锋们犯的错误。

Seymour Cray

在 Lawrence Livermore 实验室关于 Cray-1 介绍的公开演讲（1976 年）

◆ 原文：*I'm certainly not inventing vector processors. There are three kinds that I know of existing today. They are represented by the Illiac-IV, the (CDC) Star processor, and the TI (ASC) processor. Those three were all pioneering processors. … One of the problems of being a pioneer is you always make mistakes and I never, never want to be a pioneer. It's always best to come second when you can look at the mistakes the pioneers made.*

Seymour Cray

Public lecture at Lawrence Livermore Laboratories on the introduction of the Cray-1 (1976)

第一批向量处理器是由控制数据公司（Control Data Corporation, CDC）推出的 STAR-100（see Hintz and Tate [1972]）和德州仪器公司的 ASC（see Watson [1972]），两者均于 1972 年推出。这两者都是内存-内存向量处理器。它们的标量单元相对较慢——STAR 使用相同的单元处理标量和向量——使得标量流水线极其深。这两种处理器都有较高的启动开销，并处理几百到几千个元素的向量。标量和向量之间性能差异的交叉点可能超过 50 个元素。看来这两种处理器在设计时没有充分考虑到阿姆达尔定律的影响。

Seymour Cray 曾在 CDC 参与 6600 和 7600 的开发，他创立了 Cray Research，并在 1976 年推出了 Cray-1（see Russell [1978]）。Cray-1 采用向量寄存器架构，大幅降低启动开销并减少内存带宽需求。它还有效支持非单位步长并发明了向量链接操作。最重要的是，Cray-1 当时是世界上最快的标量处理器。这种良好的标量和向量性能的结合可能是 Cray-1 成功的最重要因素。一些客户购买该处理器主要是因为其出色的标量性能。许多后续的向量处理器都基于这一首个商业成功的向量处理器架构。Baskett 和 Keller [1977]对 Cray-1 进行了很好的评估。

1981 年，CDC 开始交付 CYBER 205（见 Lincoln [1982]）。205 具有与 STAR 相同的基本架构，但在各方面性能都有所提升，并且向量单元可以扩展至四条通道，每条通道都有多个功能单元和宽的 load-store 通道，能够每个时钟周期提供多个字。CYBER 205 的峰值性能远超 Cray-1；然而，在实际程序中，两者的性能差异要小得多。

1983 年，Cray Research 交付了首台 Cray X-MP（see Chen [1983]）。该处理器具有更快的时钟速率（9.5 纳秒对比 Cray-1 的 12.5 纳秒）、更好的向量链接操作

支持（允许具有 RAW 依赖的向量操作并行进行）和多条内存通道，保持了 Cray Research 在超级计算机领域的领先地位。随后推出的 Cray-2 是一种全新的设计，可配置最多四个处理器。Cray-2 的一个主要特点是使用了 DRAM，使得当时能够拥有非常大的内存。首台 Cray-2，其 256M 字（64 位字）的内存，比之前所有 Cray 机器的内存总和还要多！Cray-2 的时钟速度比 X-MP 快得多，但也有更深的流水线；然而，它缺乏向量链接操作，具有巨大的内存延迟，并且每个处理器只有一个内存通道。总体而言，Cray-2 只在需要其超大主内存的问题上比 Cray X-MP 更快。

在同一年，日本的处理器厂商进入了超级计算机市场。首先是富士通的 VP100 和 VP200（see Miura and Uchida [1983]），随后是日立 S810 和 NEC SX/2（see Watanabe [1987]）。这些处理器在性能上接近 Cray X-MP。总体而言，这三款处理器的峰值性能比 Cray X-MP 高得多。然而，由于启动开销较大，它们的典型性能通常低于 Cray X-MP。Cray X-MP 倾向于多处理器的方法，最初提供了两处理器版本，后来推出了四处理器版本。相比之下，这三款日本处理器具有可扩展的向量能力。

1988 年，Cray Research 推出了 Cray Y-MP，这是 X-MP 的更大更快版本。Y-MP 最多可支持八个处理器，并将周期时间降低到 6 纳秒。配备完整的八个处理器时，Y-MP 应当是最快的超级计算机，尽管单处理器的日本超级计算机可能比单处理器的 Y-MP 更快。1989 年底，Cray Research 分裂成两家公司，目标是在 1990 年代初推出高端处理器。Seymour Cray 领导了分拆后的 Cray Computer Corporation，直到 1995 年公司关闭。他们的初始处理器 Cray-3 计划采用砷化镓技术，但未能开发出可靠且成本效益高的实现技术。在 1996 年他因车祸不幸去世前不久，Seymour Cray 又创立了另一家公司，使用商品组件开发高性能系统。

Cray Research 专注于 C90，一款新的高端处理器，最多可配备 16 个处理器，时钟速率为 240 MHz。该处理器于 1991 年交付。1993 年，Cray Research 推出了他们的首款高度并行处理器 T3D，采用最多 2048 个 Digital Alpha21064 微处理器。1995 年，他们宣布推出一款新的低端向量机 J90 和一款高端机 T90。T90 与 C90 非常相似，但时钟速度提高了一倍（460 MHz），采用三维封装和光学时钟分配◆。

◆ 译者注：光学时钟分配是指以光信号（部分）取代电信号进行时钟信号传递的技术。

1995 年，Cray Research 被 Silicon Graphics 公司收购。1998 年，它发布了 SV1 系统，该系统在 J90 内存系统上集成了速度更快的 CMOS 处理器。它还在每个 CPU 上增加了向量数据缓存，以帮助满足增加的内存带宽需求。Silicon Graphics 在 2000 年将 Cray Research 出售给 Tera Computer，合并后的公司更名为 Cray Inc。

日本的超级计算机制造商继续改进他们的设计。2001 年，NEC SX/5 被认为是当时最快的向量超级计算机，拥有 16 条通道，时钟速率为 312 MHz，最多可配备 16 个处理器共享同一内存。2001 年发布的 NEC SX/6 是首款商用单芯片向量微处理器，集成了乱序四发射超标量处理器、标量指令和数据缓存，以及单芯片上的八通道向量单元[Kitagawa et al, 2003]。地球模拟器（Earth Simulator）由 640 个节点构成，每个节点由八个 SX-6 向量微处理器共享局部内存。2004 年发布的 SX-8 将通道数减少到 4 个，但将向量时钟速率提高到 2 GHz。标量单元

以较慢的 1 GHz 时钟速率运行，这是向量机中的常见模式，因为缺乏数据冒险，向量单元中可以使用更深的流水线。

2002 年，Cray Inc.发布了基于全新向量 ISA 的 X1。X1 SSP 处理器芯片集成了一种乱序超标量流水线，标量缓存运行在 400 MHz，双通道向量单元运行在 800 MHz。当四个 SSP 芯片结合在一起形成 MSP 时，产生的峰值向量性能为 12.8 GFLOPS，能够与当时的 NEC SX 机器相竞争。2004 年交付的 X1E 增强版将时钟速率分别提高到 565 和 1130 MHz。许多想法借鉴了 Cray T3E 设计，它是一种使用现成微处理器的 MIMD（多指令多数据）计算机。X1 具有新的指令集，拥有更多的寄存器，并在共享地址空间中将内存就近分布在处理器附近。乱序标量单元和向量单元是解耦的，使得标量单元的频率可以比向量单元更快。当数据被阻塞以利用 MSP 缓存时，向量长度较短，这与八通道向量单元不太匹配。为了处理这些较短的向量，每个只有两个向量通道的处理器可以在不同的循环上工作。

Cray X2 于 2007 年宣布，可能是最后一款 Cray 向量架构，因为考虑到市场规模，很难证明在新硅技术上的投资是合理的。处理器时钟速率为 1.3 GHz，拥有 8 个向量通道，单精度处理器峰值性能为 42 GFLOP/秒。它包括 L1 和 L2 缓存。每个节点是一个四路 SMP，最多可配备 128 GBytes 的 DRAM，最大规模为 8K 节点。

NEC SX-9 每个节点最多可配备 16 个处理器，每个处理器拥有 8 个通道，时钟速率为 3.2 GHz。它于 2008 年推出。峰值双精度向量性能为 102 GFLOP/秒。16 处理器 SMP 可以拥有 1024 GB 的 DRAM。最大规模为 512 个节点。

现代向量化编译器技术的基础和数据依赖的概念由 Kuck 及其同事[1974]在伊利诺伊大学开发。Padua 和 Wolfe [1986]对向量化编译器技术进行了很好的概述。

多媒体 SIMD 指令扩展

一家计算机硬件公司与迪斯科舞可能有什么共同之处呢？如果参考世界上最大的微处理器公司——Intel 在 1997 年发布的一则广告活动，那它们有很多共同点。

IBS 管理研究中心

《舞动领导力之路》，2002 年。

◆ 原文：*What could a computer hardware company … possibly have in common with disco dancing. A lot, if one goes by an advertisement campaign released by the world's largest microprocessor company … Intel, in 1997.*

IBS Center for Management Research "Dancing Its Way Towards Leadership," 2002

回顾历史，1957 年的 TX-2 计算机拥有分区的算术逻辑单元（ALUs），以支持当时的媒体需求，但这些理念逐渐淡出，直到 30 年后在个人计算机时代被重新发现。由于每个桌面微处理器都需要图形显示，随着晶体管预算的增加，不可避免地需要加入支持图形操作的功能。许多图形系统使用 8 位来表示 3 种原色中的每一种，再加上 8 位来表示像素的透明度。随着远程会议和电子游戏的出现，扬声器和麦克风的加入也提示了声音支持的必要性。音频采样需要超过 8 位的精度，但 16 位就足够了。

每个微处理器都有特殊支持，以便字节和半字在存储时占用更少空间，但由于典型整数程序中对这些数据大小进行算术操作的频率较低，除了数据传输之外几乎没有其他支持。Intel i860 在公司内部被视为图形加速器。其架构师认

识到许多图形和音频应用会对这些数据向量执行相同的操作[Atkins 1991; Kohn 1989]。虽然在 1989 年 i860 的晶体管预算无法支持一个完整的向量单元，但通过在 64 位 ALU 中分区进位链，它可以同时对八个 8 位操作数、四个 16 位操作数或两个 32 位操作数的短向量进行操作。这种分区 ALUs 的成本很低。适合这种支持的应用包括 MPEG（视频）、电子游戏（3D 图形）、数字摄影和远程会议（音频和图像处理）。

随着时间的推移，这种多媒体支持像病毒一样扩散到几乎所有桌面微处理器。HP 是第一个成功将这种支持引入桌面 RISC 的公司，但很快在 1990 年代，其他制造商也纷纷推出自己的版本。这些扩展最初被称为子字并行或向量。由于 Intel 在 1996 年发布 80x86 的 MMX 扩展时使用 SIMD 来描述，这个名称因成功的电视广告活动而变得流行，该广告活动涉及穿着模仿半导体制造线洁净服的迪斯科舞者。

图形处理器

自 2007 年 9 月 NVIDIA 推出 CUDA API 以来，GPU 计算进入 HPC 主流领域已经将近三年。自那时起，这项技术的采用以令人惊讶的强劲和稳定速度推进。许多一两年前开始进行小型试点项目的组织现已转向企业部署，GPU 加速机器如今在 TOP500 榜单上从第二位开始出现。对于一个通常不以快速采用新技术而闻名的社区来说，CUDA 的相对快速普及是一个值得注意的信号。与普遍认为的 GPU 计算更困难的观点相反，我认为其成功表明它并不比优秀的 CPU 编程更复杂。此外，它更清晰简洁地表达了一大类问题的并行性，从而使代码更易于维护、更具可扩展性，并更好地适应未来多核架构的映射。

Vincent Natol

《为 CUDA 喝彩》，HPCwire（2010）。

3D 图形流水线硬件从 20 世纪 80 年代初期的大型昂贵系统，逐步演变为小型工作站，并在 90 年代中后期发展为 PC 加速器。在这一过程中，发生了三个主要转变：

- 领先的图形子系统价格从 50,000 美元下降到 200 美元。
- 性能从每秒处理 5000 万像素提高到每秒处理 10 亿像素，以及从每秒处理 100,000 个顶点提高到每秒处理 1000 万个顶点。
- 原生硬件能力从线框（多边形轮廓）发展到平面着色（恒定颜色）填充多边形，再到光滑着色（插值颜色）填充多边形，最终实现了带纹理映射和基本多重纹理的全场景抗锯齿。

可扩展 GPU

从一开始，可扩展性就是图形系统的一大吸引力。工作站图形系统通过改变安装的像素处理器电路板数量，为客户提供了像素处理能力的选择。在 90 年代中期之前，PC 的图形扩展几乎不存在，只有一个选择——VGA 控制器。随着具备 3D 功能的加速器出现，市场上开始有多种选择。3dfx 在他们的 Voodoo2 上推出了原始 SLI（Scan Line Interleave）多板扩展，成为当时（1998 年）的性能王者。同样在 1998 年，NVIDIA 通过速度分级和封装，然后通过独立芯片设计

原文：*It's been almost three years since GPU computing broke into the mainstream of HPC with the introduction of NVIDIA's CUDA API in September 2007. Adoption of the technology since then has proceeded at a surprisingly strong and steady pace. Many organizations that began with small pilot projects a year or two ago have moved on to enterprise deployment, and GPU accelerated machines are now represented on the TOP500 list starting at position two. The relatively rapid adoption of CUDA by a community not known for the rapid adoption of much of anything is a noteworthy signal. Contrary to the accepted wisdom that GPU computing is more difficult, I believe its success thus far signals that it is no more complicated than good CPU programming. Further, it more clearly and succinctly expresses the parallelism of a large class of problems leading to code that is easier to maintain, more scalable and better positioned to map to future many-core architectures.*

Vincent Natol

"Kudos for CUDA," HPCwire (2010)

（GeForce 2 GTS 和 GeForce 2MX），在单一架构上推出了不同产品系列，如 Riva TNT Ultra（高性能）和 Vanta（低成本）。目前，对于特定架构代，通常需要四到五个独立的 GPU 芯片设计来覆盖桌面 PC 的性能和价格范围。此外，还有笔记本和工作站系统的独立市场。在收购 3dfx 后，NVIDIA 在 2004 年继续了多 GPU 的 SLI 概念，从 GeForce 6800 开始，为程序员和用户提供透明的多 GPU 可扩展性。在扩展范围内，功能行为是完全一致的；一个应用程序可以在任何架构系列的实现上不变地运行。

图形流水线

早期的图形硬件是可配置的，但应用开发者无法对其进行编程。每一代产品都提供了渐进的改进；然而，开发者变得越来越成熟，并要求更多的新功能，而这些功能无法合理地作为内置固定功能提供。NVIDIA 的 GeForce 3，由 Lindholm 等人[2001]描述，迈出了实现真正通用着色器可编程性的第一步。它向应用开发者开放了浮点顶点引擎的内部指令集。这与微软的 DirectX 8 和 OpenGL 的顶点着色器扩展的发布相吻合。后来的 GPU，在 DirectX 9 时代，将通用可编程性和浮点能力扩展到像素片段阶段，并在顶点阶段提供纹理。2002 年推出的 ATI Radeon 9700，具备可编程的 24 位浮点像素片段处理器，通过 DirectX 9 和 OpenGL 进行编程。GeForce FX 增加了 32 位浮点像素处理器。这是一个将不同阶段功能统一化的总体趋势，至少就应用程序员而言。NVIDIA 的 GeForce 6800 和 7800 系列采用了独立的处理器设计和专用于顶点和片段处理的独立硬件。XBox 360 在 2005 年推出了早期统一处理器 GPU，允许顶点和像素着色器在同一处理器上执行。

GPGPU：承上启下

随着支持 DirectX 9 的 GPU 问世，一些研究人员注意到 GPU 的原始性能增长路径，并开始探索使用 GPU 解决复杂的并行问题。DirectX 9 GPU 设计仅为了满足图形 API 所需的功能。要访问计算资源，程序员必须将他们的问题转化为原生图形操作。例如，要运行多个像素着色器实例，必须向 GPU 发出一个三角形（如果需要，则剪裁成矩形形状）。着色器没有执行任意内存散射操作的手段。写入内存的唯一方法是将结果作为像素颜色值发出，并配置帧缓冲操作阶段以将结果写入（或混合，如果需要）到二维帧缓冲。此外，将一个计算阶段的结果传递到下一个阶段的唯一方法是将所有并行结果写入像素帧缓冲，然后将该帧缓冲作为纹理图输入到下一个计算阶段的像素片段着色器。将通用计算映射到这一时代的 GPU 相当尴尬。然而，勇敢的研究人员通过艰苦的努力展示了一些有用的应用。这一领域被称为“GPGPU”，即在 GPU 上进行通用计算。

GPU 计算

在为 GeForce 8800 开发 Tesla 架构时，NVIDIA 意识到如果程序员能将 GPU 视为处理器，其潜在用途将大大增加。NVIDIA 选择了一种编程方法，程序员可以明确声明其工作负载的数据并行的方式。

对于 DirectX 10 代，NVIDIA 已经开始研发一种高效的浮点和整数处理器，可以运行多种并行工作负载以支持逻辑图形流水线。该处理器旨在利用执行相同代码路径的线程组的常见情况。NVIDIA 添加了具有整数字节寻址的内存加载和存储指令，以支持编译 C 程序的需求。它引入了线程块（协作线程数组）、线程块网格和屏障同步，以调度和管理高度并行的计算工作。添加了原子内存操作。NVIDIA 开发了 CUDA C/C++编译器、库和运行时软件，使程序员能够轻松访问新的数据并行计算模型并开发应用程序。

为了创建一个与供应商无关的 GPU 编程语言，许多公司正在为 OpenCL 语言创建编译器，该语言具有许多 CUDA 的特性，但可以在更多的平台上运行。截至 2011 年，如果为 GPU 编写 CUDA 代码，性能要比编写 OpenCL 代码高得多。AMD 收购了第二大 GPU 供应商 ATI，表明了 GPU 计算的扩展。AMD Fusion 架构在本版完成时宣布，是传统 GPU 与传统 CPU 的初步融合。NVIDIA 也宣布了 Project Denver，将 ARM 标量处理器与 NVIDIA GPU 结合在一个地址空间中。当这些系统出货时，了解它们的集成程度以及集成对数据并行和图形应用的性能和能耗的影响将会很有趣。

参考文献

Bouknight,W. J., S. A. Deneberg, D. E. McIntyre, J. M. Randall, A. H. Sameh, and D. L. Slotnick [1972]. “The Illiac IV system,” Proc. IEEE 60:4, 369–379. Also appears in D. P. Siewiorek, C. G. Bell, and A. Newell, Computer Structures: Principles and Examples, McGraw-Hill, New York, 1982, 306–316.

Hillis, W. D. [1985]. The Connection Multiprocessor, MIT Press, Cambridge, Mass. Hord, R. M. [1982]. The Illiac-IV, The First Supercomputer, Computer Science Press, Rockville, Md.

Slotnick, D. L., W. C. Borck, and R. C. McReynolds [1962]. “The Solomon computer,” Proc. AFIPS Fall Joint Computer Conf., December 4–6, 1962, Philadelphia, Penn., 97–107.

Unger, S. H. [1958]. “A computer oriented towards spatial problems,” Proc. Institute of Radio Engineers 46:10 (October), 1744–1750.

Asanovic, K. [1998]. “Vector Microprocessors,” Ph.D. thesis, Computer Science Division, University of California, Berkeley.

Baskett, F., and T. W. Keller [1977]. “An Evaluation of the Cray-1 Processor,” in High Speed Computer and Algorithm Organization, D. J. Kuck, D. H. Lawrie, and A. H. Sameh, eds., Academic Press, San Diego, Calif., 71–84.

Chen, S. [1983]. “Large-scale and high-speed multiprocessor system for scientific applications,” Proc. NATO Advanced Research Workshop on High Speed Computing, June 20–22, Julich, West Germany. Also in K. Hwang, ed., “Superprocessors: Design and applications,” IEEE, August, 59–73, 1984.

Flynn, M. J. [1966]. “Very high-speed computing systems,” Proc. IEEE 54:12 (December), 1901–1909.

Gebis, J. and Patterson, D. [2007]. “Embracing and extending 20th-century instruction set architectures,” IEEE Computer, 40:4 (April), 68–75.

Hintz, R. G., and D. P. Tate [1972]. “Control data STAR-100 processor design,” Proc. IEEE COMPCON, September 12–14, 1972, San Francisco, 1–4.

Kitagawa, K., S. Tagaya, Y. Hagihara, and Y. Kanoh [2003]. “A hardware overview of SX-6 and SX-7 supercomputer,” NEC Research and Development Journal 44:1 (January), 2–7.

Kozyrakis, C., and D. Patterson [2002]. “Vector vs. superscalar and VLIW architectures for embedded multimedia benchmarks,” Proc. 35th Annual Intl. Symposium on Microarchitecture (MICRO), November 18–22, 2002, Istanbul, Turkey. Kuck, D., P. P. Budnik, S.-C. Chen, D. H. Lawrie, R. A. Towle, R. E. Strebendt, E.

W. Davis, Jr., J. Han, P. W. Kraska, and Y. Muraoka [1974]. “Measurements of parallelism in ordinary Fortran programs,” Computer 7:1 (January), 37–46.

Lincoln, N. R. [1982]. “Technology and design trade offs in the creation of a modern supercomputer,” IEEE Trans. on Computers C-31:5 (May), 363–376.

Miura, K., and K. Uchida [1983]. “FACOM vector processing system: VP100/200,” Proc. NATO Advanced Research Workshop on High Speed Computing, June 20–22, Julich, West Germany. Also in K. Hwang, ed., “Superprocessors: Design and applications,” IEEE, August, 59–73, 1984.

Padua, D., and M. Wolfe [1986]. “Advanced compiler optimizations for supercomputers,” Communications of the ACM 29:12 (December), 1184–1201.

Russell, R. M. [1978]. “The Cray-1 processor system,” Communications of the ACM 21:1 (January), 63–72.

Vajapeyam, S. [1991]. “Instruction-Level Characterization of the Cray Y-MP Processor,” Ph.D. thesis, Computer Sciences Department, University of Wisconsin– Madison.

Watanabe, T. [1987]. “Architecture and performance of the NEC supercomputer SX system,” Parallel Computing 5, 247–255.

Watson, W. J. [1972]. “The TI ASC—a highly modular and flexible super processor architecture,” Proc. AFIPS Fall Joint Computer Conf., December 5–7, 1972, Anaheim, Calif., 221–228.

Atkins, M. [1991]. “Performance and the i860 Microprocessor,” IEEE Micro, 11:5 (September), 24–27, 72–78.

Kohn, L., and N. Margulis [1989]. “Introducing the Intel i860 64-Bit Microprocessor,” IEEE Micro, 9:4 (July), 15–30.

Akeley, K., and T. Jermoluk [1988]. “High-performance polygon rendering,” Proc. SIGGRAPH 88, August 1–5, 1988, Atlanta, Ga., 239–46.

Hillis, W. D., and G. L. Steele [1986]. “Data parallel algorithms,” Communications of the ACM 29:12 (December), 1170–1183 (http://doi.acm.org/10.1145/7902.7903).

IEEE 754-2008 Working Group. [2006]. DRAFT Standard for Floating-Point Arithmetic, 754-2008 (https://doi.org/10.1109/IEEESTD.2008.4610935).

Lee, W. V., et al. [2010]. “Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU,” Proc. ISCA ’10, June 19–23, 2010, Saint-Malo, France.

Lindholm, E., M. J. Kligard, and H. Moreton [2001]. A user-programmable vertex engine. In SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 149–158.

Moore, G. E. [1965]. "Cramming more components onto integrated circuits," Electronics 38:8 (April 19), 114–117.

Williams, S., A. Waterman, and D. Patterson [2009]. "Roofline: An insightful visual performance model for multicore architectures," Communications of the ACM, 52:4 (April), 65–76.

M.7 多处理器和并行处理的历史

多处理器领域有着丰富的历史;在本节中,我们将按时间段和架构来讨论这一主题。我们从 SIMD(单指令多数据流)方法和 Illiac IV 开始。然后,我们简要讨论一些其他早期实验性多处理器,并进而讨论并行处理中的一些重大争论。接下来,我们探讨当前多处理器的历史根源,最后讨论最近的进展。

SIMD 计算机:诱人的构想,长期的尝试,却无持久成功

然而,通用多处理器的成本非常高,因此考虑了进一步的设计选项,以在不严重降低系统功率或效率的情况下降低成本。这些选项包括重新集中三大主要组件之一。……集中[控制单元]形成了[一种]基本组织,例如 Illiac IV 这样的阵列处理器。

Bouknight 等 [1972]◆

◆ 原文:*The cost of a general multiprocessor is, however, very high and further design options were considered which would decrease the cost without seriously degrading the power or efficiency of the system. The options consist of recentralizing one of the three major components. … Centralizing the [control unit] gives rise to the basic organization of [an] … array processor such as the Illiac IV.*

Bouknight et al. [1972]

SIMD 模型是最早的并行计算模型之一,可以追溯到第一个大规模多处理器 Illiac IV。在该多处理器中,与最近的 SIMD 多处理器一样,其核心思想是使用单一指令同时操作多个数据项,并利用多个功能单元。最早关于 SIMD 风格计算机的想法来自 Unger [1958]和 Slotnick, Borck, 和 McReynolds [1962]。Slotnick 的 Solomon 设计构成了 Illiac IV 的基础,可能是最臭名昭著的超级计算机项目之一。虽然在推动后来项目中有用的几项技术方面取得了成功,但作为计算机项目本而言却失败了。成本从 1966 年估计的 800 万美元飙升至 1972 年的 3100 万美元,尽管仅建造了计划中的四分之一多处理器。实际性能却最多达到 15 MFLOPS,而最初对完整系统的预测为 1000 MFLOPS [Hord 1982]。1972 年交付给 NASA Ames 研究中心,该计算机又经过了三年的工程改造才可用。这些事件减缓了对 SIMD 的研究,但 Danny Hillis [1985]在 Connection Machine 中复兴了这种风格,该机器拥有 65,636 个 1 位处理器。

真正的 SIMD 计算机需要混合使用 SISD 和 SIMD 指令。有一个 SISD 主机计算机来执行不需要并行操作的分支和地址计算等操作。SIMD 指令广播到所有执行单元,每个单元都有自己的寄存器集。为了灵活性,可以在 SIMD 指令期间禁用单个执行单元。此外,极大规模并行 SIMD 多处理器依赖互连或通信网络在处理元素之间交换数据。SIMD 在处理 for 循环中的数组时效果最佳;因此,要在 SIMD 中实现大规模并行性,必须有大量的数据或数据并行性。在 case 语句中,SIMD 表现最弱,因为每个执行单元必须根据其数据执行不同的操作。拥有

错误数据的执行单元被禁用，以便正确的单元可以继续。这种情况下的性能基本上是1/n，其中n是情况的数量。

SIMD 多处理器的基本权衡是处理器性能与处理器数量之间的关系。最近的多处理器强调个体处理器性能之外的大规模并行性。例如，Connection Multiprocessor 2 提供了 65,536 个单比特宽度的处理器，而 Illiac IV 有 64 个 64 比特处理器。

在 1980 年代，由 Thinking Machines 首先复兴，然后由 MasPar 复兴的 SIMD 模型再次作为通用多处理器架构被搁置，主要有两个原因。首先，它过于僵化。许多重要问题无法使用这种风格的多处理器，并且该架构无法以竞争性方式缩小规模；即小规模 SIMD 多处理器的成本性能通常比替代方案差。其次，SIMD 无法利用微处理器技术的巨大性能和成本优势。设计者必须为其多处理器构建定制处理器，而不是利用这种低成本技术。

虽然 SIMD 计算机作为通用替代方案已经退出舞台，但这种架构风格将在特殊用途设计中继续发挥作用。许多特殊任务是高度数据并行的，并且需要有限的功能单元。因此，设计者可以内置对某些操作的支持，以及功能单元之间的硬连线互连路径。这样的组织通常被称为阵列处理器，它们在图像和信号处理等任务中非常有用。

其他的早期尝试

要搞清楚第一个 MIMD（多指令多数据流）多处理器并不容易。令人惊讶的是，例如，Eckert-Mauchly 公司制造的第一台计算机就具有重复单元以提高可用性。Holland [1959] 提出了关于多处理器的早期论点。在 20 世纪 70 年代，卡内基梅隆大学进行了两个记录良好的多处理器项目。第一个是 C.mmp [Wulf and Bell 1972; Wulf and Harbison 1978]，它由 16 个 PDP-11 通过交叉开关连接到 16 个内存单元。它是最早拥有多个处理器的多处理器之一，并采用共享内存编程模型。C.mmp 项目的研究重点主要在软件方面，特别是操作系统领域。后来的一个多处理器 Cm* [Swan et al. 1977] 是一个基于集群的多处理器，具有分布式内存和不均匀的访问时间。由于缺乏缓存和较长的远程访问延迟，数据放置变得至关重要。Gehringer、Siewiorek 和 Segall [1987] 对该多处理器及许多应用实验进行了详细描述。这些多处理器中的许多想法在 20 世纪 80 年代被再次使用，当时微处理器使得构建多处理器变得便宜得多。

关于并行处理的大讨论

在 20 世纪 60 年代中期，人们开始背离传统的计算机组织结构，因为在提高计算机运行速度的努力中，收益递减规律开始显现。……电子电路的运行速度最终受光速的限制……许多电路已经在纳秒范围内运行。

Bouknight 等 [1972]

……顺序计算机在其潜在计算能力上正在接近一个基本的物理极限。这样的极限是光速……

Angel L. DeCegama

《并行处理技术》，第一卷 (1989)

• 原文：*The turning away from the conventional organization came in the middle 1960s, when the law of diminishing returns began to take effect in the effort to increase the operational speed of a computer....Electronic circuits are ultimately limited in their speed of operation by the speed of light ... and many of the circuits were already operating in the nanosecond range.*

Bouknight et al. [1972]

• 原文：*... sequential computers are approaching a fundamental physical limit on their potential computational power. Such a limit is the speed of light ...*

Angel L. DeCegama

The Technology of Parallel Processing, Vol. I (1989)

……当今的多处理器……随着技术接近光速，正接近一个瓶颈。即使顺序处理器的组件能够以如此快的速度工作，所能期望的最佳情况也不过是每秒几百万条指令。

David Mitchell

《Transputer：时机已至》(1989)

◆ 原文：*... today's multiprocessors ... are nearing an impasse as technologies approach the speed of light. Even if the components of a sequential processor could bemade to work this fast, the best that could be expected is no more than a few million instructions per second.*

David Mitchell

The Transputer: The Time Is Now (1989)

上述引文给出了放弃当前计算模式的经典论据，而 Amdahl [1967] 则以支持继续专注于 IBM 360 架构的经典回应进行了反驳。关于并行执行优势的讨论可以追溯到 19 世纪 [Menabrea 1842]！然而，多处理器在减少单个重要程序延迟方面的有效性至今仍在探索之中。除了关于并行性优势与局限的争论外，围绕如何构建多处理器系统也有许多激烈的讨论。

虽然很难预测未来，但在 1989 年，Gordon Bell 对 1995 年做出了两项预测。在本书第一版出版时，我们收录了这些预测，当时结果尚不明朗。本节将讨论这些预测，并对其准确性进行评估。

第一个预测是，到 1995 年，将会建成一台可持续运行每秒一万亿次浮点运算（teraFLOPS，即一百万 MFLOPS）的计算机，这台计算机可能是由 4K 到 32K 节点的多计算机，或拥有数百万处理元件的连接多处理器 [Bell 1989]。为了理解这个预测的意义，每年 Gordon Bell 奖都会表彰并行计算的进步，包括最快实际程序（最高 MFLOPS）。1989 年，获奖者使用八处理器的 Cray Y-MP，实现了 1680 MFLOPS。以这些数据为基础，为了让最快的程序在 1995 年达到 1 TFLOPS，多处理器和程序每年都需要提升 3.6 倍。1999 年，第一个 Gordon Bell 奖得主突破了 1 TFLOPS 大关。他们使用专为利弗莫尔实验室设计的 5832 处理器 IBM RS/6000 SST 系统，在冲击波模拟中实现了 1.18 TFLOPS。这意味着年均提升 1.93 倍，依然相当可观。

自 1990 年代以来，人们逐渐认识到，尽管我们有技术能力制造 TFLOPS 级别的多处理器，但这种机器是否具备成本效益仍不明确，除非用于极少数与国家安全密切相关的专门应用。我们在 1990 年估算，实现 1 TFLOPS 需要约 5000 个处理器，成本约为 1 亿美元。而 Livermore 实验室的 5832 处理器 IBM 系统花费了 1.1 亿美元。可以预见，单个微处理器的性能提升（无论是成本还是性能）都会直接影响大规模多处理器的成本和性能，但一个 5000 处理器的系统仍将远高于使用同款处理器的台式机价格的 5000 倍。自那时起，更快的多处理器不断被制造出来，但近五年来最主要的提升来自处理器本身，而非并行架构的根本性突破。

Bell 的第二个预测涉及 1995 年出货的超级计算机中的数据流数量。Danny Hillis 认为，虽然拥有较少数据流的超级计算机可能是最畅销的，但最大型的多处理器将会是拥有大量数据流的多处理器，并承担大部分计算任务。Bell 与 Hillis 打赌，在 1995 年最后一个季度，使用少量数据流（≤100）的多处理器所交付的持续 MFLOPS 会超过使用大量数据流（≥1000）的多处理器。这个赌注只涉及超级计算机，即成本超过 100 万美元、用于科学应用的多处理器。对于这场赌约，持续 MFLOPS 被定义为每月的浮点运算次数，因此多处理器的可用性也影响其评分。

1989 年打赌时，谁会胜出尚无定论。1995 年，对当时已知的超级计算机的调查显示，全球仅有六台多处理器拥有超过 1000 个数据流，因此 Bell 的预测明

显获胜。事实上，1995 年，基于微处理器的小型多处理器（≤20 个处理器）已逐渐占据主导地位。1995 年，对全球在用的 500 台最高性能多处理器（基于 Linpack 评分）的调查，即 TOP500，显示最多的多处理器是基于总线的共享内存多处理器！到 2005 年，各类集群或多计算机扮演了重要角色。例如，在前 25 个系统中，有 11 个是定制集群，如 IBM Blue Gene 系统或 Cray XT3；10 个是共享内存多处理器集群（包括分布式和集中式内存）；其余 4 个则是采用现成互连的 PC 集群。

近期的进展与发展

除并行向量多处理器（参见附录 G）和最近的 IBM Blue Gene 设计外，其他所有新近的 MIMD 计算机都是基于现成的微处理器，通过总线和逻辑上的中心内存，或通过互连网络和分布式内存构建的。20 世纪 80 年代建造的一些实验性多处理器进一步完善和增强了许多今天多处理器系统的基础概念。

基于总线的缓存一致性多处理器的发展

虽然在 20 世纪 60、70 年代已经有大型主机采用了多处理器架构，但多处理器系统直到 80 年代才真正取得巨大成功。Bell [1985] 指出，关键在于微处理器体积更小，使得内存总线可以替代互连网络硬件，而可移植操作系统的出现意味着多处理器项目不再需要开发新的操作系统。在他的论文中，Bell 定义了多处理器和多计算机的术语，并为构建更大规模多处理器的两种不同方法奠定了基础。

名称	协议类型	内存写策略	独特特征	典型多处理器
Write Once	写失效	首次写入后写回内存	文献中首次介绍的监听协议	
Synapse N+1	写失效	写回	显式标明内存的所有者	Synapse 多处理器；首批商用缓存一致性多处理器
Berkeley (MOESI)	写失效	写回	有共享状态	Berkeley SPUR 多处理器；Sun Enterprise 服务器
Illinois (MESI)	写失效	写回	独立干净的私有状态；可从任何带有干净副本的缓存提供数据	SGI Power 和 Challenge 系列
Firefly	写广播	私有时写回，共享时写穿透	广播时更新内存	当前无多处理器使用；最接近的是 SPARCCenter 2000

表 M.1 对五种监听协议的总结。Archibald 和 Baer [1986] 使用这些名称来描述这五种协议，而 Eggers [1989] 则如图所示总结了它们的相同点和不同点。Firefly 协议因其首次出现的实验性 DEC Firefly 多处理器而得名。这些协议的别称基于它们所支持的状态：M=修改（Modified），E=独占（Exclusive）（私有干净），S=共享（Shared），I=无效（Invalid），O=拥有者（Owner）（共享肮脏）。

第一个采用监听缓存（snooping caches）的基于总线的多处理器是 Frank [1984] 描述的 Synapse N+1。Goodman [1983] 撰写了最早描述监听缓存的论文之一。20 世纪 80 年代末，许多商业化的基于总线、监听缓存架构相继问世，包括 Silicon Graphics 4D/240 [Baskett, Jermoluk, and Solomon 1988]、Encore Multimax [Wilson 1987] 和 Sequent Symmetry [Lovett and Thakkar 1988]。80 年代中期，出现了多种一致性协议的开发热潮，Archibald 和 Baer [1986] 对这些协议进行了很

好的综述和分析，并引用了原始论文。表 M.1 总结了几种嗅探缓存一致性协议，并展示了一些使用这些协议的多处理器系统。

90 年代初，随着采用超宽、高速总线（如 SGI Challenge 系统采用 256 位、分组型总线，支持多达 8 块处理器板和 32 个处理器）以及后来的多总线和交叉开关互连（如 Sun SPARCCenter 和 Enterprise 系统，Charlesworth [1998] 讨论了这些多处理器的互连架构）的出现，这类系统开始扩展。到 2001 年，Sun Enterprise 服务器成为当时大规模（>16 处理器）对称多处理器的主要代表。如今，大多数基于总线的机器只支持大约四个处理器，并采用交换机或其他设计来支持八个或更多处理器。

迈向大规模多处理器

在构建大规模多处理器的努力中，探索了两种不同的方向：消息传递多计算机和可扩展的共享内存多处理器。尽管曾有许多尝试构建网格和超立方体互连的多处理器，但最早成功整合所有关键要素的多处理器之一是加州理工学院开发的 Cosmic Cube [Seitz 1985]。它在路由和互连技术上取得了重要进步，并大幅降低了互连成本，使多计算机成为可行方案。基于这些思想，Intel 开发了 iPSC 860，这是一种超立方体互连的 i860 处理器集合。更新的多处理器，如 Intel Paragon，则采用了维度更低、单链路带宽更高的网络。Paragon 还在每个节点中配备了独立的 i860 作为通信控制器，虽然不少用户发现将两个 i860 处理器同时用于计算和通信效果更好。Thinking Machines 公司的 CM-5 则采用了现成的微处理器和 fat tree 互连结构（参见附录 F），并为用户提供了通信通道的用户级访问，从而显著降低了通信延迟。到 1995 年，这两种多处理器代表了消息传递多计算机领域的最高水平。

早期可扩展共享内存多处理器的尝试包括 IBM RP3 [Pfister et al. 1985]，NYU Ultracomputer [Elder 等，1985；Schwartz 1980]，伊利诺伊大学的 Cedar 项目 [Gajski et al. 1983]，以及 BBN Butterfly 和 Monarch [BBN Laboratories 1986; Rettberg et al. 1990]。这些多处理器都采用了非均匀分布式内存模型的变体，因此属于分布式共享内存（distributed shared memory, DSM）多处理器，但它们并不支持缓存一致性，这大大增加了编程难度。

RP3 和 Ultracomputer 项目都探索了同步（fetch-and-operate）以及在网络中合并引用的新思想。在这四种多处理器中，互连网络的成本最终都高于处理节点本身，这给小型多处理器带来了难题。Cray T3D/E（参见 Arpaci 等[1995]对 T3D 的评估，以及 Scott[1996]对 T3E 增强的描述）则是在这些思想基础上发展而来，采用了非一致性共享地址空间，但借鉴了多计算机领域互连技术的进步（see Scott and Thorson [1996]）。

通过结合多种思想，实现了可扩展缓存一致性的共享内存模型。实际上，基于目录的缓存一致性技术早于嗅探缓存技术。事实上，最早的缓存一致性协议就是采用目录机制的，如 Tang [1976] 所描述并在 IBM 3081 中实现。Censier 和 Feautrier [1978] 描述了一种带有内存标签的目录一致性方案。将目录分布到各个内存中以实现可扩展缓存一致性的想法，最早由 Agarwal 等人 [1988] 提出，并成为斯坦福 DASH 多处理器的基础（参见 Lenoski 等 [1990, 1992]），DASH 是首

个实际运行的缓存一致性 DSM 多处理器。DASH 是一种“胖节点”cc-NUMA 机器，使用四处理器的 SMP 作为节点，以类似 Wildfire 的方式互连，但采用了更具可扩展性的二维网格结构，而不是交叉开关互连。

Kendall Square Research 的 KSR-1 [Burkhardt et al. 1992] 是首个商用可扩展一致性共享内存的实现。它在基本 DSM 方法上进行了扩展，实现了一种称为“仅缓存内存架构”（cache-only memory architecture, COMA）的概念，即将主内存作为缓存。在 KSR-1 中，内存块可以在每个节点的主内存中复制，并通过硬件支持来处理这些复制块带来的额外一致性需求。（KSR-1 并不是严格意义上的纯 COMA，因为它没有迁移数据项的主位置，而是始终在主位置保留一份副本，实质上只实现了复制。）许多其他研究方案 [Falsafi and Wood 1997; Hagersten, Landin, and Haridi 1992; Saulsbury et al. 1995; Stenström, Joe, and Gupta 1992] 针对 COMA 风格架构及类似方法进行了探索，这些方法通过迁移 [Chandra et al. 1994; Soundararajan et al. 1998] 来减轻非均匀内存访问的负担，但没有进一步的商业实现。

Convex Exemplar 采用了两级架构来实现可扩展一致性共享内存：在最低层，采用交叉开关构建八处理器模块；然后通过环形结构连接最多 32 个这样的模块，总共可支持 256 个处理器（相关评估参见 Thekkath 等 [1997]）。Laudon 和 Lenoski [1997] 描述了 SGI Origin，它于 1996 年首次推出，基础架构与斯坦福 DASH 机器高度相似，但在可扩展性和易编程性方面有多项创新。Origin 采用位向量作为目录结构，长度为 16 或 32 位，每一位代表一个节点（每个节点包含两个处理器）；粗粒度的位向量表示允许每一位代表最多 8 个节点，总共可支持 1024 个处理器。正如 Galles [1996] 所描述，Origin 采用高性能的 fat hypercube 进行全局互连。Hristea, Lenoski 和 Keen [1997] 对 Origin 内存系统的性能进行了全面评估。

为探索带或不带多线程的可扩展一致性，多个研究原型被提出，包括 MIT 的 Alewife 机器 [Agarwal et al. 1995] 和斯坦福的 FLASH 多处理器 [Gibson et al. 2000；Kuskin et al. 1994]。

集群

集群（Clusters）很可能是在 20 世纪 60 年代由那些无法将所有工作都放在一台计算机上，或需要主机故障时备用计算机的用户“发明”的 [Pfister 1998]。Tandem 公司于 1975 年推出了一个 16 节点的集群。Digital 公司随后推出了 VAXclusters，首次亮相于 1984 年。这些集群最初是彼此独立的计算机，共享 I/O 设备，因此需要分布式操作系统来协调各自的活动。不久之后，计算机之间建立了通信链路，部分原因是为了让计算机可以分布在不同地理位置，从而在单点故障时提高可用性。用户登录到集群时，并不知道自己运行在哪台机器上。到 1993 年，DEC（现为 HP）已经售出了超过 25,000 套集群。其他早期涉足集群的公司有 Tandem（现为 HP）和 IBM（至今仍在）。如今，几乎每家公司都有集群产品。大多数集群产品的主要目标是提高可用性，性能扩展只是次要收益。

集群在科学计算领域也逐渐成为 MPP（大规模并行处理机）的竞争对手。1993 年，Beowulf 项目启动，旨在满足 NASA 提出的“用不到 5 万美元实现 1

GFLOPS 计算能力”的目标。1994 年，一个由 16 台基于 80486 处理器的现成 PC 组成的集群实现了这一目标 [Bell and Gray 2001]。这种需求推动了各种软件接口的发展，使得提交、协调和调试大型程序或大量独立程序变得更加容易。

人们也努力降低集群通信的延迟并提升带宽，多个研究项目致力于解决这一问题。(低延迟研究的一个商业成果是 VI 接口标准，后被 Infiniband 采纳，相关内容将在下文讨论。)低延迟随后在其他应用中也发挥了作用。例如，1997 年，加州大学伯克利分校用 100 台 Ultra-SPARC 台式机组成的集群，每个节点通过 160 MB/秒的 Myrinet 交换机互连，创造了数据库排序和破解加密信息的世界纪录——将最初存储在磁盘上的 8.6GB 数据在 1 分钟内完成排序，仅用 3.5 小时就破解了一个 40 位 DES 密钥。

这个名为“工作站网络”（Network of Workstations）的研究项目 [Anderson, Culler, and Patterson 1995]，还开发了 Inktomi 搜索引擎，并由此成立了一家同名初创公司。Google 借鉴 Inktomi 的做法，用台式机集群而不是大型 SMP 来构建搜索引擎，这与当时领先的搜索引擎 Alta Vista（Google 后来超越了它）采用的策略不同 [Brin and Page 1998]。到 2011 年，几乎所有互联网服务都依赖集群为数百万用户提供服务。

集群在科学家中也非常受欢迎。原因之一是成本低廉，使得个人科学家或小团队可以拥有专属于自己程序的集群。这样的集群能比在超级计算中心共享 MPP 的长队中等待（有时需等上数周）更快获得计算结果。对集群有兴趣的读者，可以参阅 Pfister [1998] 所著的趣味性集群专著。

大规模多处理器的最新趋势

在 1990 年代中后期，人们逐渐认识到，超大规模并行计算市场预期中的增长不太可能实现。由于市场未能扩展，越来越明显的是，高端并行计算市场无法承担为小众市场量身定制硬件和软件的高昂成本。也许由这一现象引发的最重要趋势就是：集群将被用来实现最高性能水平。

目前大规模多处理器大致分为四类：

- 利用 Myrinet 或 Infiniband 等互连技术集成标准桌面主板的集群。
- 由标准微处理器构建、配置为处理单元并通过定制互连连接的多计算机系统。这类系统包括 Cray XT3（采用早期 Cray 互连和简单集群架构）和 IBM Blue Gene（稍后会详细介绍这一独特机器）。
- 由小规模共享内存计算机（可能支持向量运算）组成的集群，包括 Earth Simulator（地球模拟器，拥有自己的在线期刊）。
- 大规模共享内存多处理器，如 Cray X1 [Dunigan 等，2005] 以及 SGI 的 Origin 和 Altix 系统。SGI 系统也可以配置为集群，支持超过 512 个处理器，尽管集群间只支持消息传递。

在这些设计中，IBM Blue Gene 最为引人注目，因为其设计理念与最近单处理器架构趋向多核的根本原因相似。Blue Gene 最初是 IBM 内部的一个研究项目，旨在解决蛋白质测序和折叠问题。Blue Gene 的设计者们发现，在大规模多处理器系统中，功耗日益成为关注重点，而嵌入式领域处理器的性能/功耗比要

远优于高端单处理器领域。如果高性能的实现途径是并行化，为什么不用最节能的构建模块，只是数量更多呢?

因此，Blue Gene 采用了定制芯片，集成了嵌入式 PowerPC 微处理器，其性能仅为高端 PowerPC 的一半，但功耗和芯片面积却远小于后者。这使得更多的系统功能，包括全局互连，都可以集成到同一芯片上。最终形成了高度可复制且高效的构建模块，使 Blue Gene 能够更高效地扩展到更大规模的处理器数量。Blue Gene 不是采用独立微处理器或标准桌面主板作为构建单元，而是采用处理器核心。这种方式无疑带来了更高的效率。至于市场是否能够支撑定制设计和专用软件的成本，目前仍是一个悬而未决的问题。

2006 年，美国劳伦斯利弗莫尔国家实验室的一台 Blue Gene 处理器系统拥有 32,000 个处理器（计划在 2005 年底扩展到 65,000 个），其 Linpack 性能比第三名系统（由 20 台 SGI Altix 512 处理器系统通过 Infiniband 集群互连组成）高出 2.6 倍。

Blue Gene 的前身是一台实验性机器 QCDOD，它率先采用低功耗嵌入式微处理器和高度集成的互连结构，从而大幅降低了节点的成本和功耗。

同步与一致性模型的发展

针对共享内存多处理器，已经提出了多种同步原语。Mellor-Crummey 和 Scott [1991] 对相关问题进行了综述，并给出了诸如锁和屏障等重要原语的高效实现。文献中还提供了大量参考资料，涵盖了自旋锁、队列锁和屏障等方面的重要进展。Lamport [1979] 首次提出了顺序一致性（sequential consistency）的概念，并阐释了并行程序正确执行的意义。Dubois、Scheurich 和 Briggs [1988] 提出了弱排序（weak ordering）的思想（最早在 1986 年提出）。1990 年，Adve 和 Hill 对弱排序进行了更好的定义，并提出了数据竞争自由（data-race-free）的概念；在同一会议上，Gharachorloo 及其同事 [1990] 提出了释放一致性（release consistency），并首次给出了宽松一致性模型的性能数据。更宽松的一致性模型已被广泛应用于微处理器架构，包括 Sun SPARC、Alpha 和 IA-64。Adve 和 Gharachorloo [1996] 对内存一致性及这些模型之间的差异进行了出色的教程式讲解。

其他参考资料

利用虚拟内存实现不同计算机之间共享地址空间的思想，最早由 Kai Li 在 1988 年的 Ivy 系统中提出。之后有许多论文探讨了硬件支持、软件机制和编程相关的问题。Amza 等 [1996] 描述了在工作站上构建的新一致性模型系统，Kontothanassis 等 [1997] 描述了使用远程写操作的软件共享内存方案，Erlichson 等 [1996] 则介绍了利用共享虚拟内存，通过 SMP 作为节点来构建大规模多处理器系统的方法。

关于多处理器和多计算机的信息几乎是无穷无尽的：会议、期刊论文，甚至书籍的出版速度都远超个人的学习吸收能力。毫无疑问，许多论文会被忽视——这在过去也常见。大多数主要的体系结构会议都包含多处理器相关论文。每年举办的 Supercomputing XY 会议（X 和 Y 为年份的最后两位数字）汇集了用户、体系结构师、软件开发者和厂商，会议论文集以书籍、CD-ROM 和在线形式（见

www.scXY.org）出版。两大期刊 Journal of Parallel and Distributed Computing 和 IEEE Transactions on Parallel and Distributed Systems 涵盖了并行处理的各个方面。以下参考资料中还包括了几本专注于并行处理的书籍，其中 Culler、Singh 和 Gupta [1999] 是最新的大型著作。多年来，NASA Ames 研究中心的 Eugene Miya 一直在收集并行处理论文的在线文献，目前已收录超过 35,000 条，在线地址为：liinwww.ira.uka.de/bibliography/Parallel/Eugene/index.html。

这些参考资料不仅记录了如今已被广泛应用的概念的发现过程，还描述了许多被探索但未被采纳的想法，以及那些尚未到达成熟阶段的理念。随着多核和多处理器成为高性能计算机体系结构的未来，我们预计在未来几年还会涌现许多新方法。其中少数将成功解决过去 40 年来多处理器应用中硬件和软件的关键难题！

参考文献

Adve, S. V., and K. Gharachorloo [1996]. “Shared memory consistency models: A tutorial,” IEEE Computer 29:12 (December), 66–76.

Adve, S. V., and M. D. Hill [1990]. “Weak ordering—a new definition,” Proc. 17th Annual Int’l. Symposium on Computer Architecture (ISCA), May 28–31, 1990, Seattle, Wash., 2–14.

Agarwal, A., R. Bianchini, D. Chaiken, K. Johnson, and D. Kranz [1995]. “The MIT Alewife machine: Architecture and performance,” 22nd Annual Int’l. Symposium on Computer Architecture (ISCA), June 22–24, 1995, Santa Margherita, Italy, 2–13.

Agarwal, A., J. L. Hennessy, R. Simoni, and M. A. Horowitz [1988]. “An evaluation of directory schemes for cache coherence,” Proc. 15th Annual Int’l. Symposium on Computer Architecture, May 30–June 2, 1988, Honolulu, Hawaii, 280–289.

Agarwal, A., J. Kubiatowicz, D. Kranz, B.-H. Lim, D. Yeung, G. D’Souza, and M. Parkin [1993]. “Sparcle: An evolutionary processor design for large-scale multiprocessors,” IEEE Micro 13 (June), 48–61.

Alles, A. [1995]. “ATM Internetworking,” White Paper (May), Cisco Systems, Inc., San Jose, Calif. (www.cisco.com/warp/public/614/12.html).

Almasi, G. S., and A. Gottlieb [1989]. Highly Parallel Computing, Benjamin/Cummings, Redwood City, Calif.

Alverson, G., R. Alverson, D. Callahan, B. Koblenz, A. Porterfield, and B. Smith [1992]. “Exploiting heterogeneous parallelism on a multithreaded multiprocessor,” Proc. ACM/IEEE Conf. on Supercomputing, November 16–20, 1992, Minneapolis, Minn., 188–197.

Amdahl, G. M. [1967]. “Validity of the single processor approach to achieving large scale computing capabilities,” Proc. AFIPS Spring Joint Computer Conf., April 18–20, 1967, Atlantic City, N.J., 483–485.

Amza C., A. L. Cox, S. Dwarkadas, P. Keleher, H. Lu, R. Rajamony, W. Yu, and W. Zwaenepoel [1996]. “Treadmarks: Shared memory computing on networks of workstations,” IEEE Computer 29:2 (February), 18–28.

Anderson, T. E., D. E. Culler, and D. Patterson [1995]. “A case for NOW (networks of workstations),” IEEE Micro 15:1 (February), 54–64.

Ang, B., D. Chiou, D. Rosenband, M. Ehrlich, L. Rudolph, and Arvind [1998]. “StarT-Voyager: A flexible platform for exploring scalable SMP issues,” Proc. ACM/IEEE Conf. on Supercomputing, November 7–13, 1998, Orlando, FL.

Archibald, J., and J.-L. Baer [1986]. “Cache coherence protocols: Evaluation using a multiprocessor simulation model,” ACM Trans. on Computer Systems 4:4 (November), 273–298.

Arpaci, R. H., D. E. Culler, A. Krishnamurthy, S. G. Steinberg, and K. Yelick [1995]. “Empirical evaluation of the CRAY-T3D: A compiler perspective,” Proc. 22nd Annual Int’l. Symposium on Computer Architecture (ISCA), June 22–24, 1995, Santa Margherita, Italy.

Baer, J.-L., and W.-H. Wang [1988]. “On the inclusion properties for multi-level cache hierarchies,” Proc. 15th Annual Int’l. Symposium on Computer Architecture, May 30–June 2, 1988, Honolulu, Hawaii, 73–80.

Balakrishnan, H. V., N. Padmanabhan, S. Seshan, and R. H. Katz [1997]. “A comparison of mechanisms for improving TCP performance over wireless links,” IEEE/ACM Trans. on Networking 5:6 (December), 756–769.

Barroso, L. A., K. Gharachorloo, and E. Bugnion [1998]. “Memory system characterization of commercial workloads,” Proc. 25th Annual Int’l. Symposium on Computer Architecture (ISCA), July 3–14, 1998, Barcelona, Spain, 3–14.

Baskett, F., T. Jermoluk, and D. Solomon [1988]. “The 4D-MP graphics superworkstation: Computing+graphics=40 MIPS+40 MFLOPS and 10,000 lighted polygons per second,” Proc. IEEE COMPCON, February 29–March 4, 1988, San Francisco, 468–471.

BBN Laboratories. [1986]. Butterfly Parallel Processor Overview, Tech. Rep. 6148, BBN Laboratories, Cambridge, Mass.

Bell, C. G. [1985]. “Multis: A new class of multiprocessor computers,” Science 228 (April 26), 462–467.

Bell, C. G. [1989]. “The future of high performance computers in science and engineering,” Communications of the ACM 32:9 (September), 1091–1101.

Bell, C. G., and J. Gray [2001]. Crays, Clusters and Centers, Tech. Rep. MSR-TR-2001-76, Microsoft Research, Redmond, Wash.

Bell, C. G., and J. Gray [2002]. “What’s next in high performance computing,” CACM, 45:2 (February), 91–95.

Bouknight,W. J., S. A. Deneberg, D. E. McIntyre, J. M. Randall, A. H. Sameh, and D. L. Slotnick [1972]. “The Illiac IV system,” Proc. IEEE 60:4, 369–379. Also appears in D. P. Siewiorek, C. G. Bell, and A. Newell, Computer Structures: Principles and Examples, McGraw-Hill, New York, 1982, 306–316.

Brain, M. [2000]. Inside a Digital Cell Phone, www.howstuffworks.com/insidecell- phone.htm.

Brewer, E. A., and B. C. Kuszmaul [1994]. “How to get good performance from the CM-5 data network,” Proc. Eighth Int’l. Parallel Processing Symposium (IPPS), April 26–29, 1994, Cancun, Mexico.

Brin, S., and L. Page [1998]. “The anatomy of a large-scale hypertextual Web search engine,” Proc. 7th Int’l. World Wide Web Conf., April 14–18, 1998, Brisbane, Queensland, Australia, 107–117.

Burkhardt III, H., S. Frank, B. Knobe, and J. Rothnie [1992]. Overview of the KSR1 Computer System, Tech. Rep. KSR-TR-9202001, Kendall Square Research, Boston.

Censier, L., and P. Feautrier [1978]. “A new solution to coherence problems in multicache systems,” IEEE Trans. on Computers C-27:12 (December), 1112–1118.

Chandra, R., S. Devine, B. Verghese, A. Gupta, and M. Rosenblum [1994]. “Scheduling and page migration for multiprocessor compute servers,” Proc. Sixth Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 4–7, 1994, San Jose, Calif., 12–24.

Charlesworth, A. [1998]. “Starfire: Extending the SMP envelope,” IEEE Micro 18:1 (January/February), 39–49.

Clark, W. A. [1957]. “The Lincoln TX-2 computer development,” Proc. Western Joint Computer Conference, February 26–28, 1957, Los Angeles, 143–145.

Comer, D. [1993]. Internetworking with TCP/IP, 2nd ed., Prentice Hall, Englewood Cliffs, N.J.

Culler, D. E., J. P. Singh, and A. Gupta [1999]. Parallel Computer Architecture: A Hardware/Software Approach, Morgan Kaufmann, San Francisco.

Dally, W. J., and C. I. Seitz [1986]. “The torus routing chip,” Distributed Computing 1:4, 187–196.

Davie, B. S., L. L. Peterson, and D. Clark [1999]. Computer Networks: A Systems Approach, 2nd ed., Morgan Kaufmann, San Francisco.

Desurvire, E. [1992]. “Lightwave communications: The fifth generation,” Scientific American (International Edition) 266:1 (January), 96–103.

Dongarra, J., T. Sterling, H. Simon, and E. Strohmaier [2005]. “High-performance computing: Clusters, constellations, MPPs, and future directions,” Computing in Science & Engineering, 7:2 (March/April), 51–59.

Dubois, M., C. Scheurich, and F. Briggs [1988]. “Synchronization, coherence, and event ordering,” IEEE Computer 21:2 (February), 9–21.

Dunigan, W., K. Vetter, K. White, and P. Worley [2005]. “Performance evaluation of the Cray X1 distributed shared memory architecture,” IEEE Micro, January/February, 30–40.

Eggers, S. [1989]. “Simulation Analysis of Data Sharing in Shared Memory Multiprocessors,” Ph.D. thesis, Computer Science Division, University of California, Berkeley.

Elder, J., A. Gottlieb, C. K. Kruskal, K. P. McAuliffe, L. Randolph, M. Snir, P. Teller, and J. Wilson [1985]. “Issues related to MIMDshared-memory computers: The NYU Ultracomputer approach,” Proc. 12th Annual Int’l. Symposium on Computer Architecture (ISCA), June 17–19, 1985, Boston, Mass., 126–135.

Erlichson, A., N. Nuckolls, G. Chesson, and J. L. Hennessy [1996]. “SoftFLASH: Analyzing the performance of clustered distributed virtual shared memory,” Proc.

Seventh Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 1–5, 1996, Cambridge, Mass., 210–220.

Falsafi, B., and D. A. Wood [1997]. "Reactive NUMA: A design for unifying SCOMA and CC-NUMA," Proc. 24th Annual Int'l. Symposium on Computer Architecture (ISCA), June 2–4, 1997, Denver, Colo., 229–240.

Flynn, M. J. [1966]. "Very high-speed computing systems," Proc. IEEE 54:12 (December), 1901–1909.

Forgie, J. W. [1957]. "The Lincoln TX-2 input-output system," Proc. Western Joint Computer Conference, February 26–28, 1957, Los Angeles, 156–160.

Frank, S. J. [1984]. "Tightly coupled multiprocessor systems speed memory access time," Electronics 57:1 (January), 164–169.

Gajski, D., D. Kuck, D. Lawrie, and A. Sameh [1983]. "CEDAR—a large scale multiprocessor," Proc. Int'l. Conf. on Parallel Processing (ICPP), August, Columbus, Ohio, 524–529.

Galles, M. [1996]. "Scalable pipelined interconnect for distributed endpoint routing: The SGI SPIDER chip," Proc. IEEE HOT Interconnects '96, August 15–17, 1996, Stanford University, Palo Alto, Calif.

Gehringer, E. F., D. P. Siewiorek, and Z. Segall [1987]. Parallel Processing: The Cm* Experience, Digital Press, Bedford, Mass.

Gharachorloo, K., A. Gupta, and J. L. Hennessy [1992]. "Hiding memory latency using dynamic scheduling in shared-memory multiprocessors," Proc. 19th Annual Int'l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia.

Gharachorloo, K., D. Lenoski, J. Laudon, P. Gibbons, A. Gupta, and J. L. Hennessy [1990]. "Memory consistency and event ordering in scalable sharedmemory multiprocessors," Proc. 17th Annual Int'l. Symposium on Computer Architecture (ISCA), May 28–31, 1990, Seattle, Wash., 15–26.

Gibson, J., R. Kunz, D. Ofelt, M. Horowitz, J. Hennessy, and M. Heinrich [2000]. "FLASH vs. (simulated) FLASH: Closing the simulation loop," Proc. Ninth Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), November 12–15, Cambridge, Mass., 49–58.

Goodman, J. R. [1983]. "Using cache memory to reduce processor memory traffic," Proc. 10th Annual Int'l. Symposium on Computer Architecture (ISCA), June 5–7, 1982, Stockholm, Sweden, 124–131.

Goralski, W. [1997]. SONET: A Guide to Synchronous Optical Network, McGraw-Hill, New York.

Grice, C., and M. Kanellos [2000]. "Cell phone industry at crossroads: Go high or low?" CNET News (August 31), technews.netscape.com/news/0-1004-201-2518386-0.html?tag=st.ne.1002.tgif.sf.

Groe, J. B., and L. E. Larson [2000]. CDMA Mobile Radio Design, Artech House, Boston.

Hagersten E., and M. Koster [1998]. "WildFire: A scalable path for SMPs," Proc. Fifth Int'l. Symposium on High-Performance Computer Architecture, January 9–12, 1999, Orlando, Fla.

Hagersten, E., A. Landin, and S. Haridi [1992]. “DDM—a cache-only memory architecture,” IEEE Computer 25:9 (September), 44–54.

Hill, M. D. [1998]. “Multiprocessors should support simple memory consistency models,” IEEE Computer 31:8 (August), 28–34.

Hillis, W. D. [1985]. The Connection Multiprocessor, MIT Press, Cambridge, Mass.

Hirata, H., K. Kimura, S. Nagamine, Y. Mochizuki, A. Nishimura, Y. Nakase, and T. Nishizawa [1992]. “An elementary processor architecture with simultaneous instruction issuing from multiple threads,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 136–145.

Hockney, R.W., and C. R. Jesshope [1988]. Parallel Computers 2: Architectures, Programming and Algorithms, Adam Hilger, Ltd., Bristol, England.

Holland, J. H. [1959]. “A universal computer capable of executing an arbitrary number of subprograms simultaneously,” Proc. East Joint Computer Conf. 16, 108–113.

Hord, R. M. [1982]. The Illiac-IV, The First Supercomputer, Computer Science Press, Rockville, Md.

Hristea, C., D. Lenoski, and J. Keen [1997]. “Measuring memory hierarchy performance of cache-coherent multiprocessors using micro benchmarks,” Proc. ACM/IEEE Conf. on Supercomputing, November 15–21, 1997, San Jose, Calif.

Hwang, K. [1993]. Advanced Computer Architecture and Parallel Programming, McGraw-Hill, New York.

IBM. [2005]. “Blue Gene,” IBM J. of Research and Development, 49:2/3 (special issue).

Infiniband Trade Association. [2001]. InfiniBand Architecture Specifications Release 1.0.a, www.infinibandta.org.

Jordan, H. F. [1983]. “Performance measurements on HEP—a pipelined MIMD computer,” Proc. 10th Annual Int’l. Symposium on Computer Architecture (ISCA), June 5–7, 1982, Stockholm, Sweden, 207–212.

Kahn, R. E. [1972]. “Resource-sharing computer communication networks,” Proc. IEEE 60:11 (November), 1397–1407.

Keckler, S. W., and W. J. Dally [1992]. “Processor coupling: Integrating compile time and runtime scheduling for parallelism,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 202–213.

Kontothanassis, L., G. Hunt, R. Stets, N. Hardavellas, M. Cierniak, S. Parthasarathy, W. Meira, S. Dwarkadas, and M. Scott [1997]. “VM-based shared memory on low-latency, remotememory-access networks,” Proc. 24th Annual Int’l. Symposium on Computer Architecture (ISCA), June 2–4, 1997, Denver, Colo.

Kurose, J. F., and K. W. Ross [2001]. Computer Networking: A Top-Down Approach Featuring the Internet, Addison-Wesley, Boston.

Kuskin, J., D. Ofelt, M. Heinrich, J. Heinlein, R. Simoni, K. Gharachorloo, J. Chapin, D. Nakahira, J.Baxter, M. Horowitz, A. Gupta, M. Rosenblum, and J. L. Hennessy [1994]. “The Stanford FLASH multiprocessor,” Proc. 21st

Annual Int'l. Symposium on Computer Architecture (ISCA), April 18–21, 1994, Chicago.

Lamport, L. [1979]. "How to make a multiprocessor computer that correctly executes multiprocess programs," IEEE Trans. on Computers C-28:9 (September), 241–248.

Laudon, J., A. Gupta, and M. Horowitz [1994]. "Interleaving: A multithreading technique targeting multiprocessors and workstations," Proc. Sixth Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 4–7, 1994, San Jose, Calif., 308–318.

Laudon, J., and D. Lenoski [1997]. "The SGI Origin: A ccNUMA highly scalable server," Proc. 24th Annual Int'l. Symposium on Computer Architecture (ISCA), June 2–4, 1997, Denver, Colo., 241–251.

Lenoski, D., J. Laudon, K. Gharachorloo, A. Gupta, and J. L. Hennessy [1990]. "The Stanford DASH multiprocessor," Proc. 17th Annual Int'l. Symposium on Computer Architecture (ISCA), May 28–31, 1990, Seattle, Wash., 148–159.

Lenoski, D., J. Laudon, K. Gharachorloo, W.-D. Weber, A. Gupta, J. L. Hennessy, M. A. Horowitz, and M. Lam [1992]. "The Stanford DASH multiprocessor," IEEE Computer 25:3 (March), 63–79.

Li, K. [1988]. "IVY: A shared virtual memory system for parallel computing," Proc. Int'l. Conf. on Parallel Processing (ICCP), August, The Pennsylvania State University, University Park, Penn.

Lo, J., L. Barroso, S. Eggers, K. Gharachorloo, H. Levy, and S. Parekh [1998]. "An analysis of database workload performance on simultaneous multithreaded processors," Proc. 25th Annual Int'l. Symposium on Computer Architecture (ISCA), July 3–14, 1998, Barcelona, Spain, 39–50.

Lo, J., S. Eggers, J. Emer, H. Levy, R. Stamm, and D. Tullsen [1997]. "Converting thread-level parallelism into instruction-level parallelism via simultaneous multithreading," ACM Trans. on Computer Systems 15:2 (August), 322–354.

Lovett, T., and S. Thakkar [1988]. "The Symmetry multiprocessor system," Proc. Int'l. Conf. on Parallel Processing (ICCP), August, The Pennsylvania State University, University Park, Penn., 303–310.

Mellor-Crummey, J. M., and M. L. Scott [1991]. "Algorithms for scalable synchronization on shared-memory multiprocessors," ACM Trans. on Computer Systems 9:1 (February), 21–65.

Menabrea, L. F. [1842]. "Sketch of the analytical engine invented by Charles Babbage," Bibliothèque Universelle de Genève, 82 (October).

Metcalfe, R. M. [1993]. "Computer/network interface design: Lessons from Arpanet and Ethernet." IEEE J. on Selected Areas in Communications 11:2 (February), 173–180.

Metcalfe,R. M., and D. R. Boggs [1976]. "Ethernet: Distributed packet switching for local computer networks," Communications of the ACM 19:7 (July), 395–404.

Mitchell, D. [1989]. "The Transputer: The time is now," Computer Design (RISC suppl.), 40–41.

Miya, E. N. [1985]. “Multiprocessor/distributed processing bibliography,” Computer Architecture News 13:1, 27–29.

National Research Council. [1997]. The Evolution of Untethered Communications, Computer Science and Telecommunications Board, National Academy Press, Washington, D.C.

Nikhil, R. S., G. M. Papadopoulos, and Arvind [1992]. “*T: A multithreaded massively parallel architecture,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 156–167.

Noordergraaf, L., and R. van der Pas [1999]. “Performance experiences on Sun’s WildFire prototype,” Proc. ACM/IEEE Conf. on Supercomputing, November 13–19, 1999, Portland, Ore.

Partridge, C. [1994]. Gigabit Networking, Addison-Wesley, Reading, Mass.

Pfister, G. F. [1998]. In Search of Clusters, 2nd ed., Prentice Hall, Upper Saddle River, N.J.

Pfister, G. F., W. C. Brantley, D. A. George, S. L. Harvey, W. J. Kleinfekder, K. P. McAuliffe, E. A. Melton, V. A. Norton, and J. Weiss [1985]. “The IBM research parallel processor prototype (RP3): Introduction and architecture,” Proc. 12th Annual Int’l. Symposium on Computer Architecture (ISCA), June 17–19, 1985, Boston, Mass., 764–771.

Reinhardt, S. K., J. R. Larus, and D. A. Wood [1994]. “Tempest and Typhoon: User-level shared memory,” Proc. 21st Annual Int’l. Symposium on Computer Architecture (ISCA), April 18–21, 1994, Chicago, 325–336.

Rettberg, R. D., W. R. Crowther, P. P. Carvey, and R. S. Towlinson [1990]. “The Monarch parallel processor hardware design,” IEEE Computer 23:4 (April), 18–30.

Rosenblum, M., S. A. Herrod, E. Witchel, and A. Gupta [1995]. “Complete computer simulation: The SimOS approach,” in IEEE Parallel and Distributed Technology (now called Concurrency) 4:3, 34–43.

Saltzer, J. H., D. P. Reed, and D. D. Clark [1984]. “End-to-end arguments in system design,” ACM Trans. on Computer Systems 2:4 (November), 277–288.

Satran, J., D. Smith, K. Meth, C. Sapuntzakis, M. Wakeley, P. Von Stamwitz, R. Haagens, E. Zeidner, L. Dalle Ore, and Y. Klein [2001]. “iSCSI,” IPS Working Group of IETF, Internet draft www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-07.txt.

Saulsbury, A., T. Wilkinson, J. Carter, and A. Landin [1995]. “An argument for Simple COMA,” Proc. First IEEE Symposium on High-Performance Computer Architectures, January 22–25, 1995, Raleigh, N.C., 276–285.

Schwartz, J. T. [1980]. “Ultracomputers,” ACM Trans. on Programming Languages and Systems 4:2, 484–521.

Scott, S. L. [1996]. “Synchronization and communication in the T3E multiprocessor,” Seventh Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 1–5, 1996, Cambridge, Mass., 26–
36.

Scott, S. L., and G. M. Thorson [1996]. “The Cray T3E network: Adaptive routing in a high-performance 3D torus,” Proc. IEEE HOT Interconnects ’96, August 15–17, 1996, Stanford University, Palo Alto, Calif., 14–156.

Seitz, C. L. [1985]. “The Cosmic Cube (concurrent computing),” Communications of the ACM 28:1 (January), 22–33.

Singh, J. P., J. L. Hennessy, and A. Gupta [1993]. “Scaling parallel programs for multiprocessors: Methodology and examples,” Computer 26:7 (July), 22–33.

Slotnick, D. L., W. C. Borck, and R. C. McReynolds [1962]. “The Solomon computer,” Proc. AFIPS Fall Joint Computer Conf., December 4–6, 1962, Philadelphia, Penn., 97–107.

Smith, B. J. [1978]. “A pipelined, shared resource MIMD computer,” Proc. Int’l. Conf. on Parallel Processing (ICPP), August, Bellaire, Mich., 6–8. Soundararajan, V., M. Heinrich, B. Verghese, K. Gharachorloo, A. Gupta, and J. L.

Hennessy [1998]. “Flexible use of memory for replication/migration in cache-coherent DSM multiprocessors,” Proc. 25th Annual Int’l. Symposium on Computer Architecture (ISCA), July 3–14, 1998, Barcelona, Spain, 342–355.

Spurgeon, C. [2001]. “Charles Spurgeon’s Ethernet Web site,” www.host.ots.utexas.edu/ethernet/ethernet-home.html.

Stenström, P., T. Joe, and A. Gupta [1992]. “Comparative performance evaluation of cachecoherent NUMA and COMA architectures,” Proc. 19th Annual Int’l. Symposium on Computer Architecture (ISCA), May 19–21, 1992, Gold Coast, Australia, 80–91.

Sterling, T. [2001]. Beowulf PC Cluster Computing with Windows and Beowulf PC Cluster Computing with Linux, MIT Press, Cambridge, Mass.

Stevens, W. R. [1994–1996]. TCP/IP Illustrated (three volumes), Addison-Wesley, Reading, Mass.

Stone, H. [1991]. High Performance Computers, Addison-Wesley, New York.

Swan, R. J., A. Bechtolsheim, K. W. Lai, and J. K. Ousterhout [1977]. “The implementation of the Cm* multi-microprocessor,” Proc. AFIPS National Computing Conf., June 13–16, 1977, Dallas, Tex., 645–654. Swan, R. J., S. H. Fuller, and D. P. Siewiorek [1977]. “Cm*—a modular, multimicroprocessor,” Proc. AFIPS National Computing Conf., June 13–16, 1977, Dallas, Tex., 637–644.

Tanenbaum, A. S. [1988]. Computer Networks, 2nd ed., Prentice Hall, Englewood Cliffs, N.J.

Tang, C. K. [1976]. “Cache design in the tightly coupled multiprocessor system,” Proc. AFIPS National Computer Conf., June 7–10, 1976, New York, 749–753.

Thacker, C. P., E. M. McCreight, B. W. Lampson, R. F. Sproull, and D. R. Boggs [1982]. “Alto: A personal computer,” in D. P. Siewiorek, C. G. Bell, and A. Newell, eds., Computer Structures: Principles and Examples, McGraw-Hill, New York, 549–572.

Thekkath, R., A. P. Singh, J. P. Singh, S. John, and J. L. Hennessy [1997]. “An evaluation of a commercial CC-NUMA architecture—the CONVEX Exemplar

SPP1200," Proc. 11th Int'l. Parallel Processing Symposium (IPPS), April 1–7, 1997, Geneva, Switzerland.

Tullsen, D. M., S. J. Eggers, J. S. Emer, H. M. Levy, J. L. Lo, and R. L. Stamm [1996]. "Exploiting choice: Instruction fetch and issue on an implementable simultaneous multithreading processor," Proc. 23rd Annual Int'l. Symposium on Computer Architecture (ISCA), May 22–24, 1996, Philadelphia, Penn., 191–202.

Tullsen, D. M., S. J. Eggers, and H. M. Levy [1995]. "Simultaneous multithreading: Maximizing on-chip parallelism," Proc. 22nd Annual Int'l. Symposium on Computer Architecture (ISCA), June 22–24, 1995, Santa Margherita, Italy, 392–403.

Unger, S. H. [1958]. "A computer oriented towards spatial problems," Proc. Institute of Radio Engineers 46:10 (October), 1744–1750.

Walrand, J. [1991]. Communication Networks: A First Course, Aksen Associates: Irwin, Homewood, Ill.

Wilson, A. W., Jr. [1987]. "Hierarchical cache/bus architecture for shared-memory multiprocessors," Proc. 14th Annual Int'l. Symposium on Computer Architecture (ISCA), June 2–5, 1987, Pittsburgh, Penn., 244–252.

Wolfe, A., and J. P. Shen [1991]. "A variable instruction stream extension to the VLIW architecture." Proc. Fourth Int'l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 8–11, 1991, Palo Alto, Calif., 2–14.

Wood, D. A., and M. D. Hill [1995]. "Cost-effective parallel computing," IEEE Computer 28:2 (February), 69–72.

Wulf, W., and C. G. Bell [1972]. "C.mmp—A multi-mini-processor," Proc. AFIPS Fall Joint Computer Conf., December 5–7, 1972, Anaheim, Calif., 765–777.

Wulf, W., and S. P. Harbison [1978]. "Reflections in a pool of processors—an experience report on C.mmp/Hydra," Proc. AFIPS National Computing Conf. June 5–8, 1978, Anaheim, Calif., 939–951.

Yamamoto, W., M. J. Serrano, A.R. Talcott, R. C.Wood, and M. Nemirosky [1994]. "Performance estimation of multistreamed, superscalar processors," Proc. 27th Hawaii Int'l. Conf. on System Sciences, January 4–7, 1994, Wailea, 195–204.

M.8 集群的发展

本节将介绍作为仓库级计算机（WSC）和公用计算基础的集群的发展历程。（如需了解更多信息，建议阅读 Barroso 和 Hölzle [2009] 的著作，以及访问 James Hamilton 在 http://perspectives.mvdirona.com 上的博客和演讲。）

集群：仓库级计算机的前身

集群的"发明"可以追溯到 20 世纪 60 年代，当时一些用户无法将所有工作负载都放在一台计算机上，或者需要备用机器以防主机故障 [Pfister 1998]。Tandem 公司于 1975 年推出了一个 16 节点集群，Digital 公司随后在 1984 年推出了 VAXcluster。这些集群最初是独立的计算机，共享 I/O 设备，需要分布式操作系统来协调各自的活动。不久之后，集群中的计算机之间建立了通信链路，

部分原因是为了让计算机能够分布在不同地域，从而在某个站点发生灾难时提高系统的可用性。用户登录到集群后，并不知晓自己具体运行在哪台机器上。到1993 年，DEC（现为 HP）已售出超过 25,000 个集群。其他早期参与的公司包括 Tandem（现为 HP）和 IBM（仍为 IBM）。如今，几乎所有公司都有自己的集群产品，这些产品大多以提升可用性为主要目标，而性能扩展则是次要收益。

科学计算领域，集群逐渐成为与 MPP（大规模并行处理器）竞争的选择。1993 年，Beowulf 项目启动，旨在满足 NASA 希望以低于 50,000 美元的成本获得 1 GFLOPS 计算机的需求。1994 年，利用现成的 80486 芯片组装的 16 节点 PC 集群实现了这一目标 [Bell and Gray 2001]。这种需求推动了各种软件接口的发展，使得提交、协调和调试大型程序或大量独立程序变得更加容易。

人们也不断努力降低集群内部通信的延迟并提升带宽，多个研究项目都在攻克这一难题。（低延迟研究的一个商业成果是 VI 接口标准，后被 Infiniband 采纳，后文会提到。）低延迟随后在其他应用中也展现出价值。例如，1997 年，加州大学伯克利分校利用由 100 台 UltraSPARC 桌面计算机组成的集群，每台之间通过 160MB/秒的 Myrinet 交换机连接，创造了数据库排序和破解加密消息的世界纪录——用 1 分钟排序 8.6GB 磁盘数据，破解 40 位 DES 密钥仅需 3.5 小时。

这个名为 Network of Workstations（NOW）的研究项目 [Anderson, Culler, and Patterson 1995] 还开发了 Inktomi 搜索引擎，后来成立了同名创业公司。Eric Brewer 在伯克利和公司带领 Inktomi 团队，展示了利用商品硬件构建互联网服务计算基础设施的可行性。通过在机架内采用标准化网络连接 PC 服务器，Inktomi 实现了更好的扩展性。相比之下，之前领先的搜索引擎 Alta Vista 则采用大型 SMP 架构。与科学计算领域的高性能集群相比，这一方向强调大量低成本节点和清晰的编程模型。因此，NOW 项目和 Inktomi 被认为是仓库级计算机和云计算的基础。Google 在成为主流搜索引擎时也借鉴了 Inktomi 的技术，就像 Inktomi 取代 Alta Vista 一样 [Brin and Page 1998]。（Google 最初的创新在于搜索质量，仓库级计算机的创新则是在后期才出现。）多年来，所有互联网服务都依赖集群技术来为数以百万计的用户提供服务。

效用计算：云计算的前身

如前文所述，最早的效用计算（utility computing）形式是分时系统。虽然随着更小、更便宜的个人计算机的出现，分时系统逐渐被淘汰，但在过去十年中，业界曾多次尝试让效用计算“重生”，不过都未能取得完全成功。Sun 公司于 2000 年开始在 Sun Cloud 上以每小时 1 美元的价格出售计算时间，HP 在 2001 年推出了 Utility Data Center，Intel 也在 2000 年代初尝试出售其内部超级计算机的计算时间。尽管这些服务已商业化，但实际用户寥寥无几。

相关的话题还有网格计算（grid computing），其最初目的是让科学计算程序能够跨地域分布的计算设施运行。当时，有人就质疑这一目标的合理性，更不用说实现起来有多难。网格计算通常需要非常大的系统来运行极其庞大的程序，任务往往需要多个数据中心协同完成。由于长距离通信本身存在较高延迟，单个应用在地理分布广泛的环境下并不能很好地运行。网格计算的探索最终推动了一

◆ 译者注：也有译为“公用计算”，表示共享某些资源进行计算。

些数据访问规范的形成，但网格计算社区并未开发出在高性能计算领域之外也有用的 API，因此云计算与网格计算在代码和发展历史上几乎没有交集。

Armbrust 等人[2009]指出，一旦互联网服务公司解决了大规模运行的运维难题，他们实现的规模经济效应使得运营成本低于小型数据中心。Amazon 意识到，如果这种成本优势属实，那么他们就有可能通过出售此类服务获利。2006 年，Amazon 以每实例每小时 0.10 美元的价格推出了弹性云计算（EC2）。EC2 的流行促使其他互联网公司也纷纷推出云计算服务，例如 Google App Engine 和 Microsoft Azure，虽然它们所提供的抽象层次高于 Amazon Web Services 的 x86 虚拟机。因此，如今“按需付费”计算的流行，并不是因为最近才有人提出了这个想法，而是由于技术和商业模式的结合，使得公司能够通过向大量用户提供他们真正需要的服务来获利。未来是否会出现多种成功的效用计算模式，还是整个行业会向某一标准趋同，还有待时间检验。无论如何，这一进程都值得关注。

集装箱

2003 年秋季，许多人开始思考用集装箱来容纳服务器的可行性。Internet Archive（互联网档案馆）的创始人兼馆长 Brewster Kahle 曾多次演讲，介绍如何将整个档案馆装进一个 40 英尺的集装箱。他的初衷是制作档案馆的多个副本，并分布到世界各地，以确保其长期存续，从而避免像公元前 48 年被大火焚毁的亚历山大图书馆那样的命运。与 Kahle 合作的人士在 2003 年 11 月根据他的演讲撰写了一份白皮书，更详细地探讨了集装箱化设计的方案。

同年，Google 的工程师们也在考虑用集装箱建设数据中心，并于 2003 年 12 月提交了相关专利申请。首个用于数据中心的集装箱于 2005 年 1 月交付，Google 最终于 2007 年 10 月获得该专利。2009 年 4 月，Google 首次公开披露了集装箱的应用。

Sun Microsystems 的 Greg Papadopolous 和 Applied Minds 的 Danny Hillis 在听了 Kahle 的演讲后，设计了一款名为 Sun Modular Datacenter 的产品，并于 2006 年 10 月首次亮相。（该项目的代号为“Black Box”，至今仍被许多人使用。）这种半长（20 英尺）的集装箱可容纳 280 台服务器。该产品的发布与微软宣布其正在建设可容纳 220 个 40 英尺集装箱的数据中心的消息相结合，激发了众多公司纷纷推出适用于集装箱的数据中心和服务器产品。

有趣的是，2009 年，Internet Archive 将其数据迁移到了 Sun Modular Datacenter。如今，Internet Archive 的一个副本被永久保存在埃及新亚历山大图书馆，距离古老的亚历山大图书馆遗址不远。

参考文献

Anderson, T. E., D. E. Culler, and D. Patterson [1995]. “A case for NOW (networks of workstations),” IEEE Micro 15:1 (February), 54–64.

Apache Software Foundation. [2011]. Apache Hadoop project, http://hadoop.apache.org. Armbrust, M., A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D.

Patterson, A. Rabkin, I. Stoica, and M. Zaharia [2009]. Above the Clouds: A Berkeley View of Cloud Computing, Tech. Rep. UCB/EECS-2009-28, University of California, Berkeley (http://www.eecs.berkeley.edu/Pubs/TechRpts/ 2009/EECS-2009-28.html).

Barroso, L. A. [2010]. "Warehouse scale computing [keynote address]," Proc. ACM SIG-MOD, June 8–10, 2010, Indianapolis, Ind.

Barroso, L. A., and U. Hölzle [2007]. "The case for energy-proportional computing," IEEE Computer 40:12 (December), 33–37.

Barroso, L.A., and U. Hölzle [2009]. "The datacenter as a computer: An introduction to the design of warehouse-scale machines," in M. D. Hill, ed., Synthesis Lectures on Computer Architecture, Morgan & Claypool, San Rafael, Calif. Barroso, L.A., Clidaras, J. and Hölzle, U., 2013. The datacenter as a computer:An introduction to the design of warehouse-scale machines. Synthesis lectures on computer architecture, 8(3), pp.1–154.

Barroso, L.A., Marty, M., Patterson, D., and Ranganathan, P. 2017. Attack of the Killer Microseconds. Communications of the ACM, 56(2).

Bell, C. G., and J. Gray [2002]. "What's next in high performance computing," Communications of the ACM 45:2 (February), 91–95.

Brady, J.T., 1986. A theory of productivity in the creative process. IEEE Computer Graphics and Applications, 6(5), pp.25–34.

Brin, S., and L. Page [1998]. "The anatomy of a large-scale hypertextual Web search engine," Proc. 7th Int'l. World Wide Web Conf., April 14–18, 1998, Brisbane, Queensland, Australia, 107–117.

Carter, J., and K. Rajamani [2010]. "Designing energy-efficient servers and data centers," IEEE Computer 43:7 (July), 76–78.

Chang, F., J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber [2006]. "Bigtable: A distributed storage system for structured data," in Proc. Operating Systems Design and Implementation (OSDI '06), November 6–8, 2006, Seattle, Wash.

Chang, J., J. Meza, P. Ranganathan, C. Bash, and A. Shah [2010]. "Green server design: Beyond operational energy to sustainability," Workshop on Power Aware Computing and Systems (HotPower '10), October 4–6, 2010, Vancouver, British Columbia.

Clark, J., 2014 Five Numbers That Illustrate the Mind-Bending Size of Amazon's Cloud, Bloomberg, https://www.bloomberg.com/news/2014-11-14/5-numbersthat-illustrate-the-mind-bending-size-of-amazon-s-cloud.html.

Clidaras, J., C. Johnson, and B. Felderman [2010]. Private communication.

Climate Savers Computing. [2007]. Efficiency specs, http://www.climatesaverscomputing.org/.

Clos, C., 1953. A Study of Non-Blocking Switching Networks. Bell Labs Technical Journal, 32(2), pp.406-424.

Dean, J. [2009]. "Designs, lessons and advice from building large distributed systems [keynote address]," Proc. 3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware, Co-located with the 22nd

ACM Symposium on Operating Systems Principles (SOSP 2009), October 10–11, 2009, Big Sky, Mont.

Dean, J. and Barroso, L.A., 2013. The tail at scale. Communications of the ACM, 56(2), pp.74-80.

Dean, J., and S. Ghemawat [2004]. "MapReduce: Simplified data processing on large clusters." In Proc. Operating Systems Design and Implementation (OSDI '04), December 6–8, 2004, San Francisco, 137–150.

Dean, J., and S. Ghemawat [2008]. "MapReduce: simplified data processing on large clusters," Communications of the ACM 51:1, 107–113.

DeCandia, G., D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels [2007]. "Dynamo: Amazon's highly available key-value store," in Proc. 21st ACM Symposium on Operating Systems Principles, October 14–17, 2007, Stevenson, Wash.

Doherty, W.J. and Thadhani, A.J., 1982. The economic value of rapid response time. IBM Report.

Fan, X., W. Weber, and L. A. Barroso [2007]. "Power provisioning for a warehouse-sized computer," in Proc. 34th Annual Int'l. Symposium on Computer Architecture (ISCA), June 9–13, 2007, San Diego, Calif.

A. Fikes, "Storage architecture and challenges," in Google Faculty Summit, 2010. Ghemawat, S., H. Gobioff, and S.-T. Leung (2003). "The Google file system," in Proc. 19th ACM Symposium on Operating Systems Principles, October 19–22, 2003, Lake George, N.Y.

Greenberg, A., N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta [2009]. "VL2: A scalable and flexible data center network," in Proc. SIGCOMM, August 17–21, Barcelona, Spain.

González, A.and Day, M. April 27, 2016, "Amazon, Microsoft invest billions as computing shifts to cloud," The Seattle Times. http://www.seattletimes.com/business/technology/amazon-microsoft-invest-billions-as-computing-shifts-tocloud/

Hamilton, J. [2009]. "Data center networks are in my way," Stanford Clean Slate CTO Summit, October 23, 2009, http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_CleanSlateCTO2009.pdf.

Hamilton, J. [2010]. "Cloud computing economies of scale," Proc. AWS Workshop on Genomics & Cloud Computing, June 8, 2010, Seattle, Wash. (http://mvdirona.com/jrh/TalksAndPapers/JamesHamilton_GenomicsCloud 20100608.pdf).

Hamilton, J., 2014. AWS Innovation at Scale, AWS Re-invent conference. https://www.youtube.com/watch?v=JIQETrFC_SQ

Hamilton, J., May 2015. The Return to the Cloud, http://perspectives.mvdirona.com/2015/05/the-return-to-the-cloud/

Hamilton, J., April 2017. How Many Data Centers Needed World-Wide, http://perspectives.mvdirona.com/2017/04/how-many-data-centers-needed-worldwide/

Hölzle, U. [2010]. "Brawny cores still beat wimpy cores, most of the time," IEEE Micro, July/August.

Kanev, S., Darago, J.P., Hazelwood, K., Ranganathan, P., Moseley, T., Wei, G.Y. and Brooks, D., 2015, June. Profiling a warehouse-scale computer. ACM/ IEEE 42nd Annual International Symposium on Computer Architecture (ISCA).

Lang, W., J. M. Patel, and S. Shankar [2010]. "Wimpy node clusters: What about non-wimpy workloads?" Proc. Sixth Int'l. Workshop on Data Management on New Hardware, June 7, 2010, Indianapolis, Ind.

Lim, K., P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt [2008]. "Understanding and designing new system architectures for emerging warehouse-computing environments," Proc. 35th Annual Int'l. Symposium on Computer Architecture (ISCA), June 21–25, 2008, Beijing, China.

Narayanan, D., E. Thereska, A. Donnelly, S. Elnikety, and A. Rowstron [2009]. "Migrating server storage to SSDs: Analysis of trade-offs," Proc. 4th ACM European Conf. on Computer Systems, April 1–3, 2009, Nuremberg, Germany. Pfister, G. F. [1998]. In Search of Clusters, 2nd ed., Prentice Hall, Upper Saddle River, N.J.

Pinheiro, E., W.-D. Weber, and L. A. Barroso [2007]. "Failure trends in a large disk drive population," Proc. 5th USENIX Conference on File and Storage Technologies (FAST '07), February 13–16, 2007, San Jose, Calif.

Ranganathan, P., P. Leech, D. Irwin, and J. Chase [2006]. "Ensemble-level power management for dense blade servers," Proc. 33rd Annual Int'l. Symposium on Computer Architecture (ISCA), June 17–21, 2006, Boston, Mass., 66–77.

Reddi, V. J., B. C. Lee, T. Chilimbi, and K. Vaid [2010]. "Web search using mobile cores: Quantifying and mitigating the price of efficiency," Proc. 37th Annual Int'l. Symposium on Computer Architecture (ISCA), June 19–23, 2010, Saint-Malo, France.

Schroeder, B., and G. A. Gibson [2007]. "Understanding failures in petascale computers," Journal of Physics: Conference Series 78, 188–198.

Schroeder, B., E. Pinheiro, and W.-D. Weber [2009]. "DRAMerrors in the wild: A large-scale field study," Proc. Eleventh Int'l. Joint Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS), June 15–19, 2009, Seattle, Wash.

Schurman, E. and J. Brutlag [2009]. "The User and Business Impact of Server Delays," Proc. Velocity: Web Performance and Operations Conf., June 22– 24, 2009, San Jose, Calif.

Tezzaron Semiconductor. [2004]. "Soft Errors in Electronic Memory—A White Paper, Tezzaron Semiconductor, Naperville, Ill. (http://www.tezzaron.com/about/papers/soft_errors_1_1_secure.pdf).

Vahdat, A., M. Al-Fares, N. Farrington, R. N. Mysore, G. Porter, and S. Radhakrishnan [2010]. "Scale-out networking in the data center," IEEE Micro July/ August 2010.

M.9 历史视角和参考

随着架构师们对 DSA（领域专用架构）进行探索，了解体系结构的历史可能会有所帮助。某些曾经未能在通用计算中取得成功的老旧架构思想，反而可能

非常适合领域专用架构。毕竟，这些架构在某些方面可能表现突出，要么正好契合你的应用领域，要么你的领域恰好不包含那些曾经对这些架构构成挑战的特性。例如，1960 年代的 Illiac IV（Barnes et al., 1968）和 1970 年代的 FPS 120a（Charlesworth, 1981）都采用了二维处理单元阵列，因此它们可以被视为 TPU 和 Paintbox 的前身。同样，Multiflow（Rau and Fisher, 1993）和 Itanium（Sharangpani and Arora, 2000）这类 VLIW（超长指令字）架构虽然未能在通用计算领域获得商业成功，但 Paintbox 并不存在 VLIW 架构难以应对的数据缓存未命中、不可预测的分支或庞大的代码体积等问题。

有两篇综述文章表明，定制神经网络 ASIC 的历史至少可以追溯到 25 年前（Ienne et al., 1996; Asanović, 2002）。例如，CNAPS 芯片包含一个由 64 个 16 位 ×8 位乘法器组成的 SIMD 阵列，并且多个 CNAPS 芯片可以通过一个控制器连接起来（Hammerstrom, 1990）。Synapse-1 系统则基于一种定制的脉动阵列乘加芯片 MA-16，每次可以执行 16 个 16 位乘法运算（Ramacher et al., 1991）。该系统通过串联 MA-16 芯片，并配备了专用硬件来实现激活函数。

自 1995 年起，25 台由 T0 定制 ASIC 加速的 SPERT-II 工作站被部署于语音识别领域，用于神经网络的训练和推理（Asanović et al., 1998）。40MHz 的 T0 在 MIPS 指令集架构上增加了矢量指令。其八通道矢量单元可基于 8 位和 16 位输入，每个时钟周期最多产生 16 个 32 位算术结果，使其在推理时比 SPARC-20 工作站快 25 倍，在训练时快 20 倍。他们发现 16 位精度对于训练来说不够，因此采用了两个 16 位字，这使训练时间翻倍。为克服这一缺点，他们引入了“bunches”（批处理），即每次处理 32 到 1000 个数据集，从而减少了权重更新所需时间，使训练速度超过了单字但无批处理的情况。

我们使用“图像处理单元（Image Processing Unit）”来称呼 Paintbox，以标识这一新兴的处理器类别，但这并不是该术语的首次使用。据我们所知，最早的用法是在 1999 年，当时索尼 Playstation 将其一款本质上为 MPEG2 解码器的芯片命名为 IPU（Sony/Toshiba, 1999）。2006 年，Freescale 也将 IPU 作为 i.MX31 应用处理器的一部分命名，这与我们对该术语更通用的理解更为接近（Freescale, 2006，作为 i.MX31 应用处理器的一部分）。

参考文献

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., 2016. Tensor-flow: largescale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Adolf, R., Rama, S., Reagen, B., Wei, G.Y., Brooks, D., 2016. Fathom: reference workloads for modern deep learning methods. In: IEEE International Symposium on Workload Characterization (IISWC).

Amodei, D., et al., 2015. Deep speech 2: end-to-end speech recognition in English and mandarin, arXiv:1512.02595.

Asanović, K., 2002. Programmable neurocomputing. In: Arbib, M.A. (Ed.), The Handbook of Brain Theory and Neural Networks, second ed. MIT Press,

Cambridge, MA. ISBN: 0-262-01197-2. https://people.eecs.berkeley.edu/#krste/papers/neurocomputing.pdf.

Asanović, K., Beck, A., Johnson, J., Wawrzynek, J., Kingsbury, B., Morgan, N., 1998. Training neural networks with Spert-II. In: Sundararajan, N., Saratchandran,

P. (Eds.), Parallel Architectures for Artificial Networks: Paradigms and Implementations. IEEE Computer Society Press. ISBN: 0-8186-8399-6. (Chapter 11) https://people.eecs.berkeley.edu/#krste/papers/annbook.pdf.

Bachrach, J., Vo, H., Richards, B., Lee, Y., Waterman, A., Avižienis, R., Wawrzynek, J., Asanović, K., 2012. Chisel: constructing hardware in a Scala embedded language. In: Proceedings of the 49th Annual Design Automation Conference, pp. 1216–1225.

Barnes, G.H., Brown, R.M., Kato, M., Kuck, D.J., Slotnick, D.L., Stokes, R., 1968. The ILLIAC IV computer. IEEE Trans. Comput. 100 (8), 746–757.

Bhattacharya, S., Lane, N.D., 2016. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In: Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, pp. 176–189.

Brunhaver, J., 2014. PhD thesis. Stanford.

Canis, A., Choi, J., Aldham, M., Zhang, V., Kammoona, A., Czajkowski, T., Brown, S.D., Anderson, J.H., 2013. LegUp: an open-source high-level synthesis tool for FPGA-based processor/accelerator systems. ACM Trans. Embed. Comput. Syst. 13 (2).

Canny, J., et al., 2015. Machine learning at the limit. In: IEEE International Conference on Big Data.

Caulfield, A.M., Chung, E.S., Putnam, A., Haselman, H.A.J.F.M., Humphrey, S.H.M., Daniel, P.K.J.Y.K., Ovtcharov, L.T.M.K., Lanka, M.P.L.W.S., Burger, D.C.D., 2016. A cloud-scale acceleration architecture. In: MICRO Conference.

Charlesworth, A.E., 1981. An approach to scientific array processing: the architectural design of the AP-120B/FPS-164 family. Computer 9, 18–27.

Clark, J., October 26, 2015. Google Turning Its Lucrative Web Search Over to AI Machines. Bloomberg Technology, www.bloomberg.com.

Dally, W.J., 2002. Computer architecture is all about interconnect. In: Proceedings of the 8th International Symposium High Performance Computer Architecture.

Freescale as part of i.MX31 Applications Processor, 2006. http://cache.freescale.com/files/32bit/doc/white_paper/IMX31MULTIWP.pdf.

Galal, S., Shacham, O., Brunhaver II, J.S., Pu, J., Vassiliev, A., Horowitz, M., 2013. FPU generator for design space exploration. In: 21st IEEE Symposium on Computer Arithmetic (ARITH).

Hameed, R., Qadeer, W., Wachs, M., Azizi, O., Solomatnikov, A., Lee, B.C., Richardson, S., Kozyrakis, C., Horowitz, M., 2010. Understanding sources of inefficiency in general-purpose chips. ACM SIGARCH Comput. Architect. News 38 (3), 37–47.

Hammerstrom, D., 1990. A VLSI architecture for high-performance, low-cost, onchip learning. In: IJCNN International Joint Conference on Neural Networks.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity mappings in deep residual networks. Also in arXiv preprint arXiv:1603.05027.

Huang, M., Wu, D., Yu, C.H., Fang, Z., Interlandi, M., Condie, T., Cong, J., 2016. Programming and runtime support to blaze FPGA accelerator deployment at datacenter scale. In: Proceedings of the Seventh ACM Symposium on Cloud Computing. ACM, pp. 456–469.

Iandola, F., 2016. Exploring the Design Space of Deep Convolutional Neural Networks at Large Scale (Ph.D. dissertation). UC Berkeley.

Ienne, P., Cornu, T., Kuhn, G., 1996. Special-purpose digital hardware for neural networks: an architectural survey. J. VLSI Signal Process. Syst. Signal Image Video Technol. 13 (1).

Jouppi, N., 2016. Google supercharges machine learning tasks with TPU custom chip. https://cloudplatform.googleblog.com.

Jouppi, N., Young, C., Patil, N., Patterson, D., Agrawal, G., et al., 2017. Datacenter performance analysis of a matrix processing unit. In: 44th International Symposium on Computer Architecture.

Karpathy, A., et al., 2014. Large-scale video classification with convolutional neural networks. CVPR.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst.

Kung, H.T., Leiserson, C.E., 1980. Algorithms for VLSI processor arrays. Introduction to VLSI systems.

Lee, Y., Waterman, A., Cook, H., Zimmer, B., Keller, B., Puggelli, A., Kwak, J., Jevtic, R., Bailey, S., Blagojevic, M., Chiu, P.-F., Avizienis, R., Richards, B., Bachrach, J., Patterson, D., Alon, E., Nikolic, B., Asanovic, K., 2016. An agile approach to building RISC-V microprocessors. IEEE Micro 36 (2), 8–20.

Lewis-Kraus, G., 2016. The Great A.I. Awakening. New York Times Magazine. Nielsen, M., 2016. Neural Networks and Deep Learning. http://neuralnetworks anddeeplearning.com/.

Nvidia, 2016. Tesla GPU Accelerators For Servers. http://www.nvidia.com/object/teslaservers.html.

Olofsson, A., 2011. Debunking the myth of the $100 M ASIC. EE Times. http://www.eetimes.com/author.asp?section_id=36&doc_id=1266014.

Ovtcharov, K., Ruwase, O., Kim, J.Y., Fowers, J., Strauss, K., Chung, E.S., 2015a. Accelerating deep convolutional neural networks using specialized hardware. Microsoft Research Whitepaper. https://www.microsoft.com/en-us/research/publication/accelerating-deepconvolutional-neural-networks-using-specializedhardware/.

Ovtcharov, K., Ruwase, O., Kim, J.Y., Fowers, J., Strauss, K., Chung, E.S., 2015b. Toward accelerating deep learning at scale using specialized hardware in the datacenter. In: 2015 IEEE Hot Chips 27 Symposium.

Patterson, D., Nikolić, B., 7/25/2015, Agile Design for Hardware, Parts I, II, and III. EE Times, http://www.eetimes.com/author.asp?doc_id=1327239.

Patterson, D.A., Ditzel, D.R., 1980. The case for the reduced instruction set computer. ACM SIGARCH Comput. Architect. News 8 (6), 25–33.

Prabhakar, R., Koeplinger, D., Brown, K.J., Lee, H., De Sa, C., Kozyrakis, C., Olukotun, K., 2016. Generating configurable hardware from parallel patterns. In:Proceedings of theTwenty-First InternationalConference onArchitecturalSupport for Programming Languages and Operating Systems. ACM, pp. 651–665.

Putnam, A., Caulfield, A.M., Chung, E.S., Chiou, D., Constantinides, K., Demme, J., Esmaeilzadeh, H., Fowers, J., Gopal, G.P., Gray, J., Haselman, M., Hauck, S., Heil, S., Hormati, A., Kim, J.-Y., Lanka, S., Larus, J., Peterson, E., Pope, S., Smith, A., Thong, J., Xiao, P.Y., Burger, D., 2014. A reconfigurable fabric for accelerating large-scale datacenter services. In: 41st International Symposium on Computer Architecture.

Putnam, A., Caulfield, A.M., Chung, E.S., Chiou, D., Constantinides, K., Demme, J., Esmaeilzadeh, H., Fowers, J., Gopal, G.P., Gray, J., Haselman, M., Hauck, S., Heil, S., Hormati, A., Kim, J.-Y., Lanka, S., Larus, J., Peterson, E., Pope, S., Smith, A., Thong, J., Xiao, P.Y., Burger, D., 2015. A reconfigurable fabric for accelerating large-scale datacenter services. IEEE Micro. 35 (3).

Putnam, A., Caulfield, A.M., Chung, E.S., Chiou, D., Constantinides, K., Demme, J., Esmaeilzadeh, H., Fowers, J., Gopal, G.P., Gray, J., Haselman, M., Hauck, S., Heil, S., Hormati, A., Kim, J.-Y., Lanka, S., Larus, J., Peterson, E., Pope, S., Smith, A., Thong, J., Xiao, P.Y., Burger, D., 2016. A reconfigurable fabric for accelerating large-scale datacenter services. Commun. ACM.

Qadeer, W., Hameed, R., Shacham, O., Venkatesan, P., Kozyrakis, C., Horowitz, M.A., 2015. Convolution engine: balancing efficiency & flexibility in specialized computing. Commun. ACM 58 (4).

Ragan-Kelley, J., Barnes, C., Adams, A., Paris, S., Durand, F., Amarasinghe, S., 2013. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. ACM SIGPLAN Not. 48 (6), 519–530.

Ramacher, U., Beichter, J., Raab, W., Anlauf, J., Bruels, N., Hachmann, A., Wesseling, M., 1991. Design of a 1st generation neurocomputer. VLSI Design of Neural Networks. Springer, USA.

Rau, B.R., Fisher, J.A., 1993. Instruction-level parallelism. J. Supercomput. 235, Springer Science & Business Media.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115 (3).

Sergio Guadarrama, 2015. BVLC googlenet. https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet.

Shao, Y.S., Brooks, D., 2015. Research infrastructures for hardware accelerators. Synth. Lect. Comput. Architect. 10 (4), 1–99.

Sharangpani, H., Arora, K., 2000. Itanium processor microarchitecture. IEEE Micro 20 (5), 24–43.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman,

S., 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529 (7587).

Smith, J.E., 1982. Decoupled access/execute computer architectures. In: Proceedings of the 11th International Symposium on Computer Architecture.

Sony/Toshiba, 1999. 'Emotion Engine' in PS2 ("IPU is basically an MPEG2 decoder…"). http://www.cpu-collection.de/?l0=co&l1=Sony&l2=Emotion+Engine, http://arstechnica.com/gadgets/2000/02/ee/3/.

Steinberg, D., 2015. Full-Chip Simulations, Keys to Success. In: Proceedings of the Synopsys Users Group (SNUG) Silicon Valley 2015.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

TensorFlow Tutorials, 2016. https://www.tensorflow.org/versions/r0.12/tutorials/index.html.

Tung, L., 2016. Google Translate: 'This landmark update is our biggest single leap in 10 years', ZDNet. http://www.zdnet.com/article/google-translate-thislandmarkupdate-is-our-biggest-single-leap-in-10years/.

Vanhoucke, V., Senior, A., Mao, M.Z., 2011. Improving the speed of neural networks on CPUs. https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/37631.pdf.

Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J., 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. http://arxiv.org/abs/1609.08144.

M.10 磁存储、RAID 与 I/O 总线的历史

"大容量存储"这个术语在此处用来表示容量超过一百万个字母数字字符的存储单元……

Hoagland [1963]

• 原文：*Mass storage is a term used there to imply a unit capacity in excess of one million alphanumeric characters …*

Hoagland [1963]

存储 I/O 的多样性及其相关问题，使得后续的发展历史也变得丰富多彩。（Smotherman [1989] 更深入地探讨了 I/O 的历史。）本节将讨论磁存储、RAID 以及 I/O 总线和控制器。对于那些有兴趣进一步了解排队论的读者，可以参考 Jain [1991] 和 Lazowska 等 [1984] 的著作。

磁记录最初是为了记录声音而发明的，到 1941 年时，磁带已经能够与其他存储设备竞争。1947 年 ENIAC 的成功推动了将磁带用于记录数字信息的进程。直到 1970 年代，卷式磁带一直主导着可移动存储设备。20 世纪 80 年代，IBM 3480 磁带盒成为事实上的标准（至少在大型机领域），它可以通过并行读取 18 条磁道，以每秒 3 MB 的速度传输数据。这种 1/2 英寸磁带的容量仅为 200 MB。

StorageTek 在 Powder-Horn 系统中使用的 9840 磁带盒传输速度可达 10 MB/秒，容量为 20,000 MB。该设备采用“蛇形”记录方式，即磁头来回反向而不是单一纵向记录，以此提高存储密度。这种技术称为蛇形记录（serpentine recording）。另一种 1/2 英寸磁带是数字线性磁带（DLT），其中 DLT7000 的容量为 35,000 MB，传输速度为 5 MB/秒。其竞争对手是螺旋扫描（helical scan），通过旋转磁头来提高记录密度。2001 年时，8 毫米螺旋扫描磁带容量为 20,000 MB，传输速率大约 3 MB/秒。无论密度和价格如何，磁带顺序访问的本质总是激发出对具备随机访问特性的存储设备的需求。

1953 年，IBM 的 Reynold B. Johnson 召集了 15 名科学家，目标是建造一种比磁带更快的随机存取存储系统。设计要求是能存储相当于 50,000 张 IBM 标准穿孔卡的数据，并在一秒钟内取出所需数据。Johnson 提出的磁盘驱动器设计简单却前所未有：磁头需要以千分之几英寸的高度悬浮在高速旋转的盘片表面上。24 个月后，团队成功制造出功能性原型。这台设备重达 1 吨，占地约 300 立方英尺。它就是 RAMAC-350（随机存取会计控制方法，Random Access Method of Accounting Control），拥有 50 张直径 24 英寸的盘片，转速 1200 RPM，总容量 5 MB，访问时间仅为 1 秒。

自 RAMAC 开始，IBM 在磁盘领域保持了领导地位，其存储研发总部位于加州圣何塞，即 Johnson 团队当年的工作地点。许多未来竞争对手的磁盘制造商领导者都在 IBM 开始了职业生涯，很多磁盘公司也聚集在圣何塞附近。

虽然 RAMAC 是第一个使用磁盘的系统，但真正的重大突破出现在后来使用气浮磁头的磁盘上。磁头依靠高速转动的盘片表面产生的气垫悬浮在极其接近盘面的高度。这种气垫既能让磁头跟随表面微小的不平整，又能保持极低的悬浮高度。后续的进步主要来自零部件质量和精度的提升。到 2001 年时，磁头悬浮高度仅为 2 到 3 微英寸，而在 RAMAC 驱动器中，这个高度为 1000 微英寸。

可移动磁头硬盘很快成为主流高速磁性存储，但由于成本高昂，磁带在 1970 年代之前仍被广泛使用。下一项重要发展是 IBM 于 1962 年推出的可移动硬盘，允许共享昂贵的控制电子设备，推动了硬盘超越磁带，成为更受欢迎的存储介质。1962 年的 IBM 1311 硬盘面密度为每平方英寸 50,000 位，存储成本大约为每兆字节 800 美元。IBM 也在 1970 年发明了软盘驱动器，最初用于存放 IBM 370 系列的微代码。软盘在大约十年后因个人电脑而流行起来。

硬盘的第二个重大突破是大约 1973 年的“温彻斯特硬盘”设计。它有两个关键优势：首先，集成电路降低了 CPU，以及磁盘控制器和磁头控制电子元件的成本，使得硬盘无需再共享电子设备，固定硬盘经济可行。其次，由于固定硬盘可以密封，这极大减少了环境和控制问题。密封设计使磁头能贴近磁盘表面，从而提高了面密度。IBM 发布的首个密封硬盘有两个主轴，每个容量 30 MB，因为绰号“30-30”，人们就将其称为 Winchester（温彻斯特），名字来源于美国最受欢迎的步枪 Winchester 94，它的子弹口径同样叫“30-30”。温彻斯特硬盘在 1980 年代迅速普及，并在中期完全取代了可移动硬盘。在这之前，控制电子设备的成本太高，迫使存储介质必须是可更换的。

正如附录 D 提到的，当 DRAM 在面密度上逐渐逼近硬盘时，IBM 内部会议讨论了硬盘的未来。硬盘设计师们得出结论：硬盘必须以每年 60% 的提升速度进步，才能挫败 DRAM 的威胁，这相比历史上每年 29% 的提升是一个巨大飞跃。实现这一点的关键是磁阻头，特别是巨磁阻（GMR）磁头的应用，使得高密度成为可能。由于竞争激烈，从实验室达到密度新纪录到实际出货的时间差已经大幅缩短。

个人电脑带来了对小型硬盘（SFF，小尺寸规格）的市场需求，因为大型机使用的 14 英寸硬盘对于 PC 来说过大。到 2006 年，3.5 英寸硬盘是市场主流，但 2.5 英寸硬盘因笔记本电脑而在出货量上占有重要份额。至于 iPOD 或摄像机等手持设备是否会持续推动更小硬盘的需求尚未确定。例如，20 世纪 90 年代早期曾开发出 1.8 英寸硬盘用于掌上电脑，但该市场最终选择了闪存，1.8 英寸硬盘也因此很快消失。

RAID

20 世纪 80 年代 PC 所使用的小型硬盘（SFF hard disk）促使加州大学伯克利分校的一组研究人员提出了“廉价磁盘冗余阵列”（RAID）的概念。这支团队曾参与过精简指令集计算机（RISC）的研究，他们预计未来 CPU 的速度会大幅提升，于是提出问题：这些随 PC 配备的小硬盘能做些什么？在 I/O 领域能做些什么来跟上更快处理器的步伐？他们主张用 50 块小硬盘替换一台大型机硬盘，从而利用多个独立磁头大幅提升性能。许多小硬盘甚至比一块大型机硬盘更省电、占地空间更小。然而，问题在于：硬盘数量增多会显著降低平均故障间隔时间（MTTF）。于是，他们独立提出了磁盘冗余和“旋转校验”的思想，以便在保持像大型机硬盘一样高可靠性的同时，获得更高性能。

在介绍他们想法时遇到的困难是：一些研究人员听说过包含某种冗余技术的磁盘阵列，但并不理解伯克利的提案。因此，首篇 RAID 论文 [Patterson, Gibson, and Katz 1987] 不仅论证了采用小型硬盘阵列的可行性，也起到了入门教材和对已有磁盘阵列工作的分类总结作用。比如镜像（RAID 1）早已应用于 Tandem 等容错计算机。1987 年时，Thinking Machines 已经采用包含 32 块数据盘和 7 块校验盘、并用 ECC 做纠错的阵列（RAID 2）；Honeywell Bull 公司更早就有 RAID 2 产品。同时，在科学计算机中，也曾使用单一校验盘的磁盘阵列（RAID 3）。伯克利论文进一步描述了支持扇区访问的单一校验盘（RAID 4）和旋转校验（RAID 5）。Chen 等人 [1994] 提供了一份综述，涵盖 RAID 最初的思想、商业产品以及后续的发展。

伯克利团队并不知道，IBM 在研发 AS/400 计算机时，工程师们也独立提出了旋转校验，用于提升大容量硬盘阵列的可靠性。IBM 在伯克利团队发表论文之前就申请了 RAID 5 的专利。而 RAID 1、RAID 2、RAID 3 的专利则由多家公司更早申请，这成为后来大量法律诉讼的根源。

伯克利的 RAID 论文诞生于万维网问世之前，但却激发了全球工程师的想象力——论文被传真到世界各地。例如，现在属于 Seagate 的一家公司中，一位工程师就先后收到朋友和客户寄来的 7 份论文复印件。EMC 原本是为 IBM 提供 DRAM 内存板的厂商，但 1988 年 IBM 的新政策几乎让 EMC 无法继续维持

这项业务。据推测，伯克利的 RAID 论文也被送到了 EMC 高管的案头，他们于是决定转向由 IBM 主导的硬盘存储市场。他们采纳了论文中的思路——用大量小硬盘来对抗大型机硬盘，并在 1990 年推出 RAID 产品。为了可靠性，这款产品采用镜像模式（RAID 1），而 RAID 5 产品则过了很久才在 EMC 推出。在接下来一年里，Micropolis 推出了 RAID 3 产品，Compaq 提供了 RAID 4 产品，而 Data General、IBM 和 NCR 则推出了 RAID 5 产品。

RAID 的思想很快传播到工作站和服务器产业。Byte 杂志上一篇介绍 RAID 的文章（参见 Anderson [1990]）更促使 RAID 产品进入台式机 PC，这让伯克利团队感到意外。他们关注的重点是性能和可靠性兼顾，但对 PC 市场而言，更高的可用性也颇具吸引力。

另一个令人惊讶的地方是磁盘阵列的成本。由于需要冗余电源和风扇、支持“热插拔”的磁盘驱动器、RAID 专用硬件控制器，以及冗余硬盘等因素，首批磁盘阵列的价格远高于单纯的硬盘成本。也许正是因为这样，RAID 的“廉价”（Inexpensive）逐渐被人们解释为“独立”（Independent）。如今，许多市场营销部门和技术文档都只将 RAID 解释为“独立磁盘冗余阵列”。

EMC 的转型非常成功；到 2006 年，EMC 已成为存储系统的全球领先供应商，而 NetApp 则成为网络存储系统的领导者。那一年，RAID 产业的规模达到 300 亿美元，超过 80% 的非 PC 硬盘销售都用于 RAID 系统。为了表彰他们的贡献，1999 年 Garth Gibson、Randy Katz 和 David Patterson 获得了 IEEE Reynold B. Johnson 信息存储奖，以表彰他们提出了“廉价磁盘冗余阵列（RAID）”的成果。

I/O Buses and Controllers

无处不在的微处理器不仅催生了 20 世纪 70 年代的个人计算机，也推动了 80 年代末和 90 年代将控制功能逐步移入 I/O 设备的趋势。如今，I/O 设备继续延伸这一方向，把控制器直接嵌入设备本身。这类设备被称为智能设备，一些总线标准（如 SCSI）正是为它们而设计的。智能设备能够自己处理许多底层任务，并将结果排队，从而放宽了主机的时序约束。举例来说，许多兼容 SCSI 的硬盘内部都配有磁道缓冲区，支持预读（read ahead）和连接/断开操作。因此，在一条 SCSI 链上，当某个磁盘正从缓冲区向总线传输数据时，其他磁盘可以同时进行寻道或填充其缓冲区。

相比之下，最初的 RAMAC 控制器由真空管构成，它仅需完成几个简单的功能：将磁头移动到目标磁道，等待数据经过磁头，再带着奇偶校验传输数据。而 SCSI（Small Computer Systems Interface，小型计算机系统接口）则是一个鲜明例子，它最初由 Shugart 发明，起名为 SASI，后来在 IEEE 推动下成为标准。其特点是：一家公司发明总线后，积极鼓励其他厂家生产兼容设备，大家都能即插即用。

SCSI 曾经面对多种“接班人”候选标准，其中当时的领先者是 光纤通道仲裁环（Fibre Channel Arbitrated Loop, FC-AL）。然而，SCSI 委员会不断提升总线时钟频率，从而延长了 SCSI 的生命力。随着 串行 SCSI（SAS） 和 串行 ATA（SATA） 的诞生，SCSI 和 ATA 可能都会拥有非常长的生命周期。

也许最早的多厂商总线是 DEC 在 1970 年 PDP-11 计算机上引入的 Unibus。不过，像 DEC 这样开放的“总线开放政策”与一些公司形成了鲜明对比。那些公司坚持专有总线并申请接口专利，从而阻止了其他厂家生产兼容设备。使总线专有化不仅抬高了成本，还限制了可用 I/O 设备的数量，因为这些设备必须专门为该总线开发接口。与此相对，Intel 推动的 PCI 总线则代表了 PC 内部重回开放标准化 I/O 总线的潮流。其后继者为 PCI-X，而在 2000 年时正在研发的 Infiniband 则进一步推进了标准化。这些标准都是跨公司行业协会共同推动的成果。

RAMAC 时代的机器不仅带来了存储设备，还带来了 I/O 中断机制。第一个将中断从“只用于检测算术错误”扩展到“检测异步 I/O 事件”的机器被认为是 1954 年的 NBS DYSEAC [Leiner and Alexander 1954]。次年，IBM 的 SAGE 系统成为首台实际运行 DMA 的计算机。和今天的 DMA 一样，SAGE 配备了地址计数器，可以在 CPU 运算的同时执行块传输。

早期的 IBM 360 计算机在 I/O 系统方面开创了许多至今仍在使用的概念。IBM 360 是第一台广泛使用 DMA 的商用计算机，并引入了 I/O 程序 的概念，能由设备直接解释执行。还能“链式调用”I/O 程序，这是当时的重要创新。360 中提出的“通道”概念实际上就对应于今天的 I/O 总线。

Myer 和 Sutherland [1968] 曾写过一篇经典论文，探讨 I/O 控制器在复杂性与性能间的权衡。他们借用宗教中的“轮回（wheel of reincarnation）”概念，形象描述出这样一个循环：不断增强 I/O 处理器的能力，最后却需要再配套一个更简单的协处理器来辅助。附录 D 中引用的他们的话正是一则警世寓言。IBM 大型机的 I/O 通道及其独立的 I/O 处理器，可以视为 Infiniband 架构的灵感来源之一——今天的 Infiniband 主机通道适配器（HCA）卡上，就内置了自己的处理器。

参考文献

Anderson, D. [2003]. “You don’t know jack about disks,” Queue 1:4 (June), 20–30. Anderson, D., J. Dykes, and E. Riedel [2003]. “SCSI vs. ATA—more than an interface,” Proc. 2nd USENIX Conf. on File and Storage Technology (FAST ’03), March 31–April 2, 2003, San Francisco.

Anderson, M. H. [1990]. “Strength (and safety) in numbers (RAID, disk storage technology),” Byte 15:13 (December), 337–339.

Anon. et al. [1985]. A Measure of Transaction Processing Power, Tandem Tech. Rep. TR 85.2. Also appeared in Datamation, 31:7 (April), 112–118.

Bashe, C. J., W. Buchholz, G. V. Hawkins, J. L. Ingram, and N. Rochester [1981]. “The architecture of IBM’s early computers,” IBM J. Research and Development 25:5 (September), 363–375.

Bashe, C. J., L. R. Johnson, J. H. Palmer, and E. W. Pugh [1986]. IBM’s Early Computers, MIT Press, Cambridge, Mass.

Blaum, M., J. Brady, J. Bruck, and J. Menon [1994]. “EVENODD: An optimal scheme for tolerating double disk failures in RAID architectures,” Proc. 21st Annual Int’l. Symposium on Computer Architecture (ISCA), April 18–21, 1994, Chicago, 245–254.

Blaum, M., J. Brady, J. Bruck, and J. Menon [1995]. “EVENODD: An optimal scheme for tolerating double disk failures in RAID architectures,” IEEE Trans. on Computers 44:2 (February), 192–202.

Blaum, M., J. Brady, J., Bruck, J. Menon, and A. Vardy [2001]. “The EVENODD code and its generalization,” in H. Jin, T. Cortes, and R. Buyya, eds., High Performance Mass Storage and Parallel I/O: Technologies and Applications, IEEE & Wiley Press, New York, 187–208.

Blaum, M., J. Bruck, and A. Vardy [1996]. “MDS array codes with independent parity symbols,” IEEE Trans. on Information Theory, IT-42 (March), 529–542.

Brady, J. T. [1986]. “A theory of productivity in the creative process,” IEEE CG&A (May), 25–34.

Brown, A., and D. A. Patterson [2000]. “Towards maintainability, availability, and growth benchmarks: A case study of software RAID systems.” Proc. 2000 USENIX Annual Technical Conf., June 18–23, San Diego, Calif.

Bucher, I. V., and A. H. Hayes [1980]. “I/O performance measurement on Cray-1 and CDC 7000 computers,” Proc. Computer Performance Evaluation Users Group, 16th Meeting, October 20–23, 1980, Orlando, Fl., 245–254.

Chen, P. M., G. A. Gibson, R. H. Katz, and D. A. Patterson [1990]. “An evaluation of redundant arrays of inexpensive disks using an Amdahl 5890,” Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, May 22–25, 1990, Boulder, Colo.

Chen, P. M., and E. K. Lee [1995]. “Striping in a RAID level 5 disk array,” Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, May 15–19, 1995, Ottawa, Canada, 136–145.

Chen, P. M., E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson [1994]. “RAID: High-performance, reliable secondary storage,” ACM Computing Surveys 26:2 (June), 145–188.

Corbett, P., B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar [2004]. “Row-diagonal parity for double disk failure correction,” Proc. 3rd USENIX Conf. on File and Storage Technology (FAST ’04), March 31–April 2, 2004, San Francisco.

Denehy, T. E., J. Bent, F. I. Popovici, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau [2004]. “Deconstructing storage arrays,” Proc. 11th Int’l. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 7–13, 2004, Boston, Mass., 59–71.

Doherty, W. J., and R. P. Kelisky [1979]. “Managing VM/CMS systems for user effectiveness,” IBM Systems J. 18:1, 143–166.

Douceur, J. R., and W. J. Bolosky [1999]. “A large scale study of file-system contents,” Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, May 1–9, 1999, Atlanta, Ga., 59–69.

Enriquez, P. [2001]. “What happened to my dial tone? A study of FCC service disruption reports,” poster, Richard Tapia Symposium on the Celebration of Diversity in Computing, October 18–20, 2001, Houston, Tex.

Friesenborg, S. E., and R. J. Wicks [1985]. DASD Expectations: The 3380, 3380-23, and MVS/XA, Tech. Bulletin GG22-9363-02, IBM Washington Systems Center, Gaithersburg, Md.

Gibson, G. A. [1992]. Redundant Disk Arrays: Reliable, Parallel Secondary Storage, ACM Distinguished Dissertation Series, MIT Press, Cambridge, Mass.

Goldstein, S. [1987]. Storage Performance—An Eight Year Outlook, Tech. Rep. TR 03.308-1, IBM Santa Teresa Laboratory, San Jose, Calif.

Gray, J. [1990]. "A census of Tandem system availability between 1985 and 1990," IEEE Trans. on Reliability, 39:4 (October), 409–418.

Gray, J. (ed.) [1993]. The Benchmark Handbook for Database and Transaction Processing Systems, 2nd ed., Morgan Kaufmann, San Francisco.

Gray, J., and A. Reuter [1993]. Transaction Processing: Concepts and Techniques, Morgan Kaufmann, San Francisco.

Gray, J., and D. P. Siewiorek [1991]. "High-availability computer systems." Computer 24:9 (September), 39–48.

Gray, J., and C. van Ingen [2005]. Empirical Measurements of Disk Failure Rates and Error Rates," MSR-TR-2005-166, Microsoft Research, Redmond, Wash.

Gurumurthi, S., A. Sivasubramaniam, and V. Natarajan [2005]. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management, Proceedings of the International Symposium on Computer Architecture (ISCA), June, 38–49.

Henly, M., and B. McNutt [1989]. DASD I/O Characteristics: A Comparison of MVS to VM, Tech. Rep. TR 02.1550, IBM General Products Division, San Jose, Calif.

Hewlett-Packard. [1998]. "HP's '5NINES:5MINUTES' vision extends leadership and re-defines high availability in mission-critical environments," February 10, www.future.enterprisecomputing.hp.com/ia64/news/5nines_vision_pr.html.

Hoagland, A. S. [1963]. Digital Magnetic Recording, Wiley, New York.

Hospodor, A. D., and A. S. Hoagland [1993]. "The changing nature of disk controllers." Proc. IEEE 81:4 (April), 586–594.

IBM. [1982]. The Economic Value of Rapid Response Time, GE20-0752-0, IBM, White Plains, N.Y., 11–82.

Imprimis. [1989]. Imprimis Product Specification, 97209 Sabre Disk Drive IPI-2 Interface 1.2 GB, Document No. 64402302, Imprimis, Dallas, Tex.

Jain, R. [1991]. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley, NewYork.

Katz, R. H., D. A. Patterson, and G. A. Gibson [1989]. "Disk system architectures for high performance computing," Proc. IEEE 77:12 (December), 1842–1858.

Kim, M. Y. [1986]. "Synchronized disk interleaving," IEEE Trans. on Computers C-35:11 (November), 978–988.

Kuhn, D. R. [1997]. "Sources of failure in the public switched telephone network," IEEE Computer 30:4 (April), 31–36.

Lambright, D. [2000]. "Experiences in measuring the reliability of a cache-based storage system," Proc. of First Workshop on Industrial Experiences with Systems

Software (WIESS 2000), Co-Located with the 4th Symposium on Operating Systems Design and Implementation (OSDI), October 22, 2000, San Diego, Calif.

Laprie, J.-C. [1985]. "Dependable computing and fault tolerance: Concepts and terminology," Proc. 15th Annual Int'l. Symposium on Fault-Tolerant Computing, June 19–21, 1985, Ann Arbor, Mich., 2–11.

Lazowska, E. D., J. Zahorjan, G. S. Graham, and K. C. Sevcik [1984]. Quantitative System Performance: Computer System Analysis Using Queueing Network Models, Prentice Hall, Englewood Cliffs, N.J. (Although out of print, it is available online at www.cs.washington.edu/homes/lazowska/qsp/.)

Leiner, A. L. [1954]. "System specifications for the DYSEAC," J. ACM 1:2 (April), 57–81.

Leiner, A. L., and S. N. Alexander [1954]. "System organization of the DYSEAC," IRE Trans. of Electronic Computers EC-3:1 (March), 1–10.

Maberly, N. C. [1966]. Mastering Speed Reading, New American Library, New York.

Major, J. B. [1989]. "Are queuing models within the grasp of the unwashed?" Proc. Int'l. Conf. on Management and Performance Evaluation of Computer Systems, December 11–15, 1989, Reno, Nev., 831–839.

Mueller, M., L. C. Alves, W. Fischer, M. L. Fair, and I. Modi [1999]. "RAS strategy for IBM S/390 G5 and G6," IBM J. Research and Development, 43:5–6 (September–November), 875–888.

Murphy, B., and T. Gent [1995]. "Measuring system and software reliability using an automated data collection process," Quality and Reliability Engineering International, 11:5 (September–October), 341–353.

Myer, T. H., and I. E. Sutherland [1968]. "On the design of display processors," Communications of the ACM, 11:6 (June), 410–414.

National Storage Industry Consortium. [1998]. "Tape Roadmap," www.nsic.org.

Nelson, V. P. [1990]. "Fault-tolerant computing: Fundamental concepts," Computer 23:7 (July), 19–25.

Nyberg, C. R., T. Barclay, Z. Cvetanovic, J. Gray, and D. Lomet [1994]. "Alpha- Sort: A RISC machine sort," Proc. ACM SIGMOD, May 24–27, 1994, Minneapolis, Minn.

Okada, S., S. Okada, Y. Matsuda, T. Yamada, and A. Kobayashi [1999]. "System on a chip for digital still camera," IEEE Trans. on Consumer Electronics 45:3 (August), 584–590.

Patterson, D. A., G. A. Gibson, and R. H. Katz [1987]. A Case for Redundant Arrays of Inexpensive Disks (RAID), Tech. Rep. UCB/CSD 87/391, University of California, Berkeley. Also appeared in Proc. ACM SIGMOD, June 1–3, 1988, Chicago, 109–116.

Pavan, P., R. Bez, P. Olivo, and E. Zanoni [1997]. "Flash memory cells—an overview," Proc. IEEE 85:8 (August), 1248–1271.

Robinson, B., and L. Blount [1986]. The VM/HPO 3880-23 Performance Results, IBM Tech. Bulletin GG66-0247-00, IBM Washington Systems Center, Gaithersburg, Md.

Salem, K., and H. Garcia-Molina [1986]. “Disk striping,” Proc. 2nd Int’l. IEEE Conf. on Data Engineering, February 5–7, 1986, Washington, D.C., 249–259.

Scranton, R. A., D. A. Thompson, and D. W. Hunter [1983]. The Access Time Myth, Tech. Rep. RC 10197 (45223), IBM, Yorktown Heights, N.Y.

Seagate. [2000]. Seagate Cheetah 73 Family: ST173404LW/LWV/LC/LCV Product Manual, Vol. 1, Seagate, Scotts Valley, Calif. (www.seagate.com/support/disc/manuals/scsi/29478b.pdf).

Smotherman, M. [1989]. “A sequencing-based taxonomy of I/O systems and review of historical machines,” Computer Architecture News 17:5 (September), 5–15. Reprinted in Computer Architecture Readings, M. D. Hill, N. P. Jouppi, and G. S. Sohi, eds., Morgan Kaufmann, San Francisco, 1999, 451–461.

Talagala, N. [2000]. “Characterizing Large Storage Systems: Error Behavior and Performance Benchmarks,” Ph.D. dissertation, Computer Science Division, University of California, Berkeley.

Talagala, N., and D. Patterson [1999]. An Analysis of Error Behavior in a Large Storage System, Tech. Report UCB//CSD-99-1042, Computer Science Division, University of California, Berkeley.

Talagala, N., R. Arpaci-Dusseau, and D. Patterson [2000]. Micro-Benchmark Based Extraction of Local and Global Disk Characteristics, CSD-99-1063, Computer Science Division, University of California, Berkeley.

Talagala, N., S. Asami, D. Patterson, R. Futernick, and D. Hart [2000]. “The art of massive storage: A case study of a Web image archive,” IEEE Computer (November), 22–28.

Thadhani, A. J. [1981]. “Interactive user productivity,” IBM Systems J. 20:4, 407–423.