

C-AMAT：大数据时代的数据存取模型

孙贤和

美国伊利诺理工大学

特邀专栏作家

关键词：并行存储 孙-倪定律 存储系统测量与分析工具

在冯诺依曼体系结构下，数据的移动（即数据的访问能力）已严重制约计算机运算速度的提高。目前，这一难题有望得到突破。C-AMAT 并发式平均存储访问时间模型将大幅度提高计算机数据存取速度^[1]。

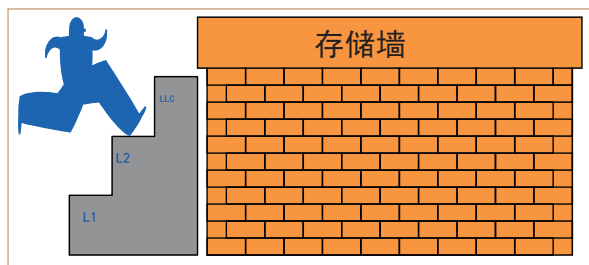


图1 存储墙导致存储访问能力制约运算速度

性能是计算机追求的目标，数据是计算机处理的对象。计算机的数据运算能力和数据访问能力都在逐年提升，但前者的提升速度远远大于后者，使得两者之间形成了逐年增加的相对差距。因此数据访问能力制约了最终性能的进步。这就是著名的存储墙问题 (Memory Wall Problem)。

我们是最早开展相关研究的人。在1990年我们首次提出的内存制约加速比模型被业界称为孙-倪定律 (Sun-Ni's Law)^[2]。孙-倪定律揭示了计算加速比将受限于存储容量的规律，是与阿姆达尔 (Amdahl) 定律和古斯塔夫森 (Gustafson) 定律并列的可扩

展计算的著名三大定律，已被收入多本并行计算机教科书，成为计算机学科研究生的必修内容。1994年，学术界通过平均存储访问时间 (AMAT) 模型正式提出了存储墙问题^[3]，从单次存储访问延迟的角度再次揭示了计算机性能受限于存储延迟的规律。20年过去了，在孙-倪定律和存储墙问题的指导下，现代计算机设计对存储访问的性能极为重视。今天的一个处理器芯片，像 Pentium Pro, Alpha21164, Strong Arm SA110, 及 Loongson 3A 等，80% 的晶体管都用于片上高速缓存而不是计算，并采用了许多新存储技术，包括并发存储技术。根据惯例，存储性能的测试由 AMAT 来完成。但是，50 年前提出的 AMAT 只考虑了存储访问的局部性，没有考虑存储的并发性，对现代存储系统的评估严重不足。尽管缺乏表征并发性能力，但因为没有更好的方法，所以 AMAT 至今仍是工业界和学术界表征存储系统性能的通用工具。轩尼诗 (Hennessy) 和帕特森 (Patterson) 在其所著的经典计算机体系结构教科书中也是这样介绍 AMAT 的^[4]。近年来，学术界也逐渐认识到 AMAT 的不足，提出了内存级并行 (memory level parallelism, MLP) 的新方法^[5]。MLP 表征了存储访问的并发性，但没有表征存储访问的局部性，在实践中需要和 AMAT 配合使用。如何在实践中将它们配合好是一个变数。更严重的问题是，MLP 是一个测量尺度，AMAT 是一个分析工具。经过 MLP

处理后, AMAT 中的三个参数 H 、 MR 、 AMP 的物理含义不再明确, 不能再直接用于指导分析。数学表达式也不再规整。虽然 MLP 在一定程度上对存储并行度进行了测量, 但无法用于处理器数据停顿的原因分析, 也无法帮助 AMAT 进行分析。轩尼诗和帕特森在他们最新出版的书中仍然使用 AMAT 作为表征、分析存储系统性能的基本工具, 但同时指出了 AMAT 的不足^[5]。

我们在 1990 年工作的基础上, 对存储系统性能进行了长期潜心研究和深入思考, 特别是从 2004 年开始研究存储系统的访问模型问题。十年磨一剑, 我们终于完成了存储系统测量工具、周期平均访问数 (Access Per memory active Cycle, APC)^[6] 和存储系统分析工具、并发式平均存储访问时间模型 (Concurrent-AMAT, C-AMAT)^[1] 这两大重要工作。APC 侧重于测量方法的研究, 给出了并发存储的测量方法和尺度。C-AMAT 则给出了严格规整的数学表达式和逻辑证明, 将 AMAT 推广到可同时考虑存储访问的局部性和并行性。

C-AMAT 包括 5 个参数, 分别表征命中存储请求的并发度、缺失存储请求的并发度、存储请求的命中时间、缺失率和平均缺失代价。C-AMAT 以最有限的参数同时表征了存储访问的局部性和并行性, 更好地表达了 AMAT 和 MLP 要表达的信息, 并将这些信息的互动关系统一在一个公式之下。研究表明, C-AMAT 能显式表达存储停顿时间, 并与最终的应用性能联系在一起^[7]。这 5 个参数中有 4 个是新的。通过这些新参数, C-AMAT 提供了一个找到最佳参数组合的工具。最新研究表明, 找到最佳并发存储参数可将现有 C-AMAT 速度提高 6 倍, 找到系统整体最佳参数组合, C-AMAT 的并发存储可将 AMAT 的存储速度提高两百多倍。如果硬件和软件再作相应的改进, 那上限就会更高了。

下面给出了 AMAT 和 C-AMAT 的公式:

$$AMAT = H + MR \times AMP \quad (1)$$

$$C-AMAT = \frac{H}{C_H} + pMR \times \frac{pAMP}{C_M} \quad (2)$$

在传统的 AMAT 公式 (1) 中, H 表示高速缓存命中时间、 MR 表示缓存的缺失率、 AMP 表示平均缺失代价。 AMP 是所有存储访问缺失代价的算术平均值。

在并发式 C-AMAT 的公式 (2) 中, C_H 表示命中并发度、 C_M 表示“纯粹”缺失并发度。现代计算机采用的多端口高速缓存、多 Bank 高速缓存、流水高速缓存等并发存储技术就是通过改善 C_H 起作用。非阻塞高速缓存、超前执行 (run-ahead)、同时多线程 (SMT) 等并发存储技术就是通过改善 C_M 起到提升存储访问性能的作用。在 C-AMAT 模型里, 我们首次引入了“纯粹缺失”这一非常重要的概念。并不是所有的缺失都是纯粹缺失, 只有缺失期间包括至少一个纯粹缺失周期, 在这个周期里整个存储系统中没有命中发生, 这样的缺失才称为纯粹缺失。之所以引入这一新概念, 是基于我们对处理器数据停顿原因的深刻认识: 在并发存储系统中, 缺失不

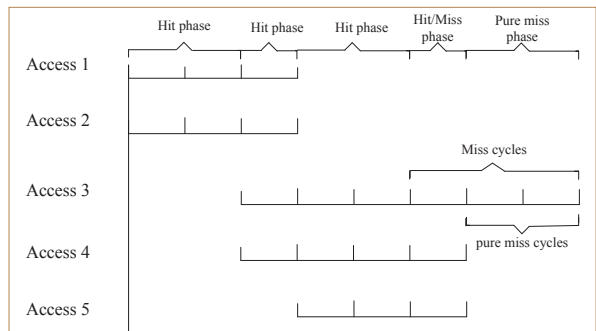


图2 C-AMAT原理示例

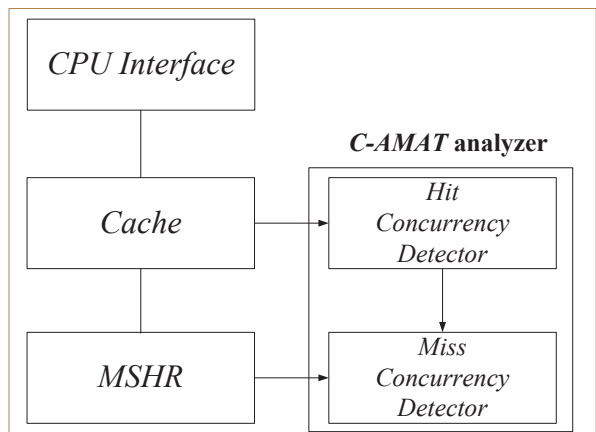


图3 C-AMAT 测量装置

一定引发处理器停顿，只有纯粹缺失才会引发处理器停顿。纯粹缺失率 pMR 定义为纯粹缺失访问的次数与全部存储访问的次数之比。纯粹平均缺失代价 $pAMP$ 是平均每个纯粹缺失访问中的纯粹缺失周期的数量。纯粹缺失给出了并发性和局部性的关系，并直接挑战“局部性永远是好的”这一现代计算机和计算机算法设计的基本原则。纯粹缺失和 C-AMAT 给现代计算机和计算机算法的设计带来了新思路。它们将影响或改变新一代计算机和计算机算法设计。

图 2 给出了一个示例来解释纯粹缺失和命中周期的概念。图中有 5 个不同的存储访问发生。每个访问首先要历经一个三周期的命中阶段。如果访问没有在高速缓存中命中，则会有一个大小不确定的缺失代价。缺失代价的大小取决于最终在何处获得了数据。访问 1、2、5 是命中访问。访问 3 和 4 是缺失访问。访问 3 有三个周期的缺失代价，访问 4 只有一个周期的缺失代价。如果从并发存储的角度去看，只有访问 3 包括两个纯粹缺失周期。尽管访问 4 有一个缺失周期，但是由于该周期与访问 5 的命中周期重叠在一起，因此这个周期不是纯粹缺失周期。因此，这 5 个存储访问的纯粹缺失率为 0.2，远远小于传统的缺失率 0.4。之所以要忽略与命中访问完全重叠的缺失，是因为这些缺失不会引发处理器的停顿。在有其他数据得到命中的情况下，处理器可以继续计算。由公式 (2) 可以得到，C-AMAT 是 $8/5=1.6$ 周期 / 访问；而由公式 (1) 得到的 AMAT 是 $3+0.4 \times 2=3.8$ 周期 / 访问。C-AMAT 与 AMAT 的差就是并发存储的贡献。在这个简单例子里，并发使存储速度增加了一倍。

图 3 给出了 C-AMAT 的测量装置。其中命中并发度探测器 (hit concurrency detector, HCD) 统计全部的命中周期和每个命中阶段的同时命中情况，并计算平均命中并发度。HCD 同时通知缺失并发度探测器 (miss concurrency detector, MCD) 当前周期是否有命中发生。MCD 用来探测纯粹缺失周期的数量并统计每个纯粹缺失阶段的并发度，根据 HCD 提供的信息，MCD 可以判别某个周期是否是纯粹缺失

周期，某个访问是否是纯粹缺失访问，从而计算出纯粹缺失并发度 C_M 、纯粹缺失率 pMR 和纯粹缺失代价 $pAMP$ 。这些参数可以在各个存储层次上被测量出来。C-AMAT 也可以在各个存储层次上被测量出来。

值得注意的是，C-AMAT 的重要贡献不在于测量自身的值，因为它的值可以通过测量 APC (Access Per memory active Cycle) 得到^[1]。C-AMAT 的贡献在于它给出了一个同时表征局部性和并发性且与 AMAT 保持形式统一的便于分析的参数公式，给出了系统设计和优化的量化工具。了解 AMAT 的读者都知道，AMAT 的平均缺失代价可以递归到存储层次的下一层。C-AMAT 同样可以在存储层次上进行递归计算。等式 (3) 给出 L1 C-AMAT 递归到 L2 C-AMAT 的示例。依次类推，C-AMAT 可以递归到更深层次。请注意在等式 (3) 中有两个系数 pMR_1 和 η_1 。这两个系数的乘积， $pMR_1 \times \eta_1$ ，体现了只有纯粹缺失（而不是所有的缺失）会造成处理器停顿这一事实。参数 η_1 是一个可以测量且具有物理意义的参数，其中 C_m 表示（普通）缺失的并发度， C_M 表示纯粹缺失的并发度。 pMR_1 和 η_1 均小于 1，起到弱化 C-AMAT₂ 的作用，对存储系统进行跨层次的优化有深刻的指导意义。

$$C-AMAT_1 = \frac{H_1}{C_{H_1}} + pMR_1 \times \eta_1 \times C-AMAT_2 \quad (3)$$

Where

$$C-AMAT_1 = \frac{H_1}{C_{H_1}} + pMR_1 \times \frac{pAMP_1}{C_{M_1}},$$

$$C-AMAT_2 = \frac{H_2}{C_{H_2}} + pMR_2 \times \frac{pAMP_2}{C_{M_2}},$$

$$\eta_1 = \frac{pAMP_1}{AMP_1} \times \frac{C_{m_1}}{C_{M_1}}$$

APC 是 C-AMAT 的测量工具^[1]，同时也保证了 C-AMAT 值的合理性。APC 看起来非常简单，和 IPC (instruction per cycle) 十分相似，但实际上 APC 里的周期和 IPC 里的周期完全不同，定义不同，测量也不同。在 APC 中，周期是存储活动周期 (memory active cycle)，不是通用的 CPU 周期，所以 APC 也

叫 APMAC (存储活动周期平均访问数, access per memory active cycle)。同时 APC 采用重叠 (overlapping) 的访存时间统计方法: 在有两个或多个存储访问同时进行, 周期只增加一次。这两个取值的方法非常重要。用存储活动周期替换 CPU 周期得以把存储与计算进行分割, 使 APC 可应用在多层次存储的每个存储层上。这一替换体现了以数据, 而不是以计算为中心的观点。它使 APC 可直接有力地指导解决存储墙问题。重叠是并行的概念, 是对并行存储进行测量。C-AMAT 提供了一个并行存储的理念和研发并行存储的工具。这个理念和工具给硬件开发和算法设计都打开了一扇新的大门。

C-AMAT 是一个开拓性的工作。它有简洁统一的数学表达, 有严格准确的数学证明, 有便捷可行的测量方法, 它的作用在于表征千变万化的应用运行时高度复杂的存储系统的局部性和并发性的信息, 它的价值表现在利用这些信息的成千上万的设计和算法上, 它的贡献在于表达了并行存储这一理念。C-AMAT 使并行存储对计算机的总体性能的贡献变得可测、可优化, 它把并行存储带入了计算机的总体设计, 它对新一代计算机和计算机算法的设计将产生巨大的影响。■

致谢:

C-AMAT 是王达伟博士与作者的共同工作。C-AMAT 的应用是刘宇航博士与作者的共同工作。刘宇航博士对此文的形成给予了帮助。

孙贤和教授



孙贤和博士是伊利诺理工大学(Illinois Institute of Technology)计算机科学系的大学杰出教授(University Distinguished Professor)和系主任。国际电气电子工程师学会院士(IEEE Fellow), 美国阿贡国家实验室(Argonne National Laboratory)客座教授和伊利诺理工大学可扩展计算软件实验室主任。他是八个国际期刊的编委, 包括并行与分布式计算领域的顶级期刊*IEEE Transactions on Parallel and Distributed Systems*和*Journal of Parallel and Distributed Computing*, 是重要学术会议的执行主席、副主席、程序委员会委员。他是中国科学院海外评审委员, 中国科学技术协会海智专家, 八大特邀代表。现在的主要研究兴趣包括并行和分布式计算, 内存和I/O系统, 大数据系统软件, 计算系统性能评估和优化。从2009年到2013年, 他的论文被引用次数超过2694次。因为C-AMAT的贡献和C-AMAT对工商业未来的潜在影响, 孙教授被著名的美国工商业杂志*Crain*评选为2014年3月的风云人物。

参考文献

- [1] X. H. Sun, D. Wang, Concurrent Average Memory Access Time, *IEEE Computer*, vol.47, no.5 pp.74~80, May 2014.
- [2] X. H. Sun and L. M. Ni, Another view on parallel speedup, *Proceedings of IEEE Supercomputing Conference*, Nov. 1990, 324~333.
- [3] W. A. Wulf and S. A. McKee, Hitting the memory wall: implications of the obvious, *ACM SIGARCH computer architecture news*, vol. 23, 20~24, 1995.
- [4] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A quantitative approach* (5th edition): Elsevier, 2012.
- [5] Y. Chou, B. Fahs, and S. Abraham, Microarchitecture Optimizations for Memory-Level Parallelism, *Proceedings of the 31st International Symposium on Computer Architecture*, June 2004.
- [6] D. Wang, X. H. Sun. APC: A Novel Memory Metric and Measurement Methodology for Modern Memory System, *IEEE Transactions on Computers*, June 2014, IEEE TC digital print (DOI Bookmark: <http://doi.ieeeecomputersociety.org/10.1109/TC.2013.38>).
- [7] Y. H. Liu, X. H. Sun, Reevaluating Memory Stall Time via Concurrent AMAT. Illinois Institute of Technology Technical Report (IIT/CS-SCS-2013-12), 2013.