

极大化科学研究的影响力 *

作者: 马克·希尔 (Mark D. Hill)

关键词: 埃克特 - 莫契利奖 洞察与初步假设 测试和优化假设

译者: 刘宇航

译者前记

2019年计算机体系结构领域最高奖埃克特 - 莫契利奖 (Eckert-Mauchly Award) 颁发给美国威斯康星大学麦迪逊分校教授马克·希尔 (Mark D. Hill), 以表彰他对存储系统和并行计算机的结构设计和性能评估作出的卓越贡献。埃克特 - 莫契利奖以 1946 年诞生的世界上第一台电子计算机 ENIAC 的两位发明人约翰·普雷斯珀·埃克特 (John Presper Eckert) 和约翰·威廉·莫契利 (John William Mauchly) 的名字命名, 始颁于 1979 年, 至今已有 40 年的历史, 由 ACM 和 IEEE-CS 共同主办, 表彰获奖人对计算机和数字系统体系结构的贡献。历史获奖人中至今有六位最终获得了 ACM 图灵奖, 包括获得 2017 图灵奖的大卫·帕特森 (David Patterson), 此外还有一批接近图灵奖的计算机科学家, 比如吉恩·阿姆达尔 (Gene Amdahl)、西摩·克雷 (Seymour Cray)、戈登·贝尔 (Gordon Bell) 等。

引言

别人的许多研究论文都展示了他们的精彩成果, 中国有句古话, “与其临渊羡鱼, 不如退而结

网”¹, 这篇短文旨在帮助你更好地开展自己的研究, 创造你自己的优秀研究成果, 特别是在计算机体系结构和系统这个领域开展研究。这些经验是我在职业生涯中学到的, 我荣幸地获得了埃克特 - 莫契利奖, 所以我职业生涯中的经验可能对你具有借鉴意义。我按照科学的方法组织这篇短文, 分以下四个步骤依次介绍: 选择一个好的问题; 提出见解和假设; 测试和改进假设; 根据需要重复上述步骤。

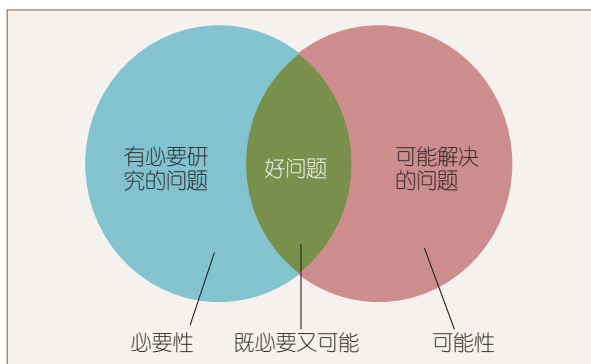


图1 好问题的标准

步骤一: 选择一个好的问题

开展具有创造性、有重要影响力的研究的第一步, 是从无限多可能问题的集合中挑选出一个好问题。如图1所示, 好的研究问题处于两个准则的交

* 这是马克·希尔 (Mark Hill) 近期撰写的一篇博客 (<https://www.sigarch.org/increasing-your-research-impact/>), 译者进行了适度修改和补充。

¹ 出自《淮南子·说林训》, 此句为译者添加。

集的位置，也就是通常需要同时满足两个标准：一是必要性，即“如果你能做出来，人们会在意”；二是可能性，即“你能做出来，哪怕是部分地做出来”。开创性的研究不会来自你玩《危险边缘》(Jeopardy!)²的诱惑。

以我在读研究生时所目睹的廉价冗余磁盘阵列(RAID)的研发过程为例，来考虑如何选题吧。你可能对 RAID 5³ 对旋转块擦除代码的新颖工作方式印象深刻，但这项研究真正的智慧之处是认识或意识到问题及其机遇。个人计算机硬盘因为销量巨大，比传统的洗衣机那样大的老式磁盘更有效率地存储大量数据。但是，个人计算机硬盘不能用来储存数量极其庞大的数据，因为个人计算机磁盘在一定程度上不可靠，把它们当成一个阵列使用时更不可靠⁴。因此，研究者寻求一种方案让磁盘阵列更可靠，于是 RAID 就诞生了。

正如研发 RAID 的过程中所发生的那样，我建议寻找将改变(change)最佳方案并具有巨大影响力的问题。在计算机体系结构和系统中，这些机遇的发生可能源自新软件（如关于增强现实的软件）、新技术（如个人计算机磁盘或非易失性存储器）或其他(细分)领域的进步(如使用机器学习来优化硬件)。

我还建议年轻人要积极参加学术会议或在学术

休假(sabbaticals)⁵期间与业界紧密联系，从而甄选出好的研究问题。正如本文稍后将会讨论到的，通过学术会议的方式，我们酝酿诞生了 GEMS；通过学术休假的方式，我们酝酿诞生了 Gables。

步骤二：洞察与初步假设

在选择了好问题的基础上，好的研究总是从初始的洞察和初步的假设下寻找解决方案开始。虽然模拟器在计算机体系结构领域论文的实验方法中居主导地位，但我发现通过分类、关联和模型这三种简单的方法，就可以很好地完成洞察与假设这个初始步骤。

分类(Taxonomies)将问题或现有解决方案组织成一些可以展示新机遇的“格子”。自然科学上最著名的分类是门捷列夫⁶的元素周期表，其目的是为了发现缺失的元素。计算机体系结构中众所周知的一个分类是我们提出的高速缓存缺失 3C 模型⁷，当时是为了从洪水般大量的高速缓存模拟数据中发现新的富有洞察力的结论。作为第三个关于分类的例子，基于日志的事务内存(LogTM)，对学术界的影响目前已经超过了对工业界产品的影响。我们寻求一个可以提交任何事务的系统，但只是适度改变

² 《危险边缘》是美国哥伦比亚广播公司益智问答游戏节目，已有数十年历史。该节目的比赛以一种独特的问答形式进行，问题设置的涵盖面非常广泛，涉及到历史、文学、艺术、流行文化、科技、体育、地理、文字游戏等各个领域。与一般问答节目相反，《危险边缘》以答案形式提问、以问题形式作答。参赛者需具备丰富的知识，还会解析隐晦含义、反讽与谜语等，而电脑并不擅长进行这类复杂思考。——译者注

³ RAID 5 是一种存储性能、数据安全和存储成本兼顾的存储解决方案。——译者注

⁴ 在没有容错机制时，一个由多个磁盘组成的阵列，只要有一个磁盘发生故障，这个阵列就发生故障。——译者注

⁵ 学术休假在 1880 年由哈佛大学首创，是美国大学教师发展的一种重要制度形式，它源于 19 世纪末美国研究型大学的崛起及其对教师国际化的需求，后被证实能提升教师教学水平、促进科研创新能力、提高教师队伍士气、缓解教师职业倦怠等方面有明显功效。近年来，我国也在探索实践类似的制度。——译者注

⁶ 门捷列夫(1834—1907)，俄国科学家，发现化学元素的周期性，制作出世界上第一张元素周期表，并据此预见了一些尚未发现的元素。——译者注

⁷ 关于高速缓存缺失的 3C（将缺失分为 Compulsory、Capacity 和 Conflict 三种）模型是作者于 1989 年即 30 年前提出，M.D. Hill, and A. J. Smith. “Evaluating Associativity in CPU Caches.” *IEEE Transactions on Computers* 38.12(1989):1612-1630. 该模型是计算机体系结构的经典之作，被收录于 David Pattern 的《计算机体系结构量化研究方法》教科书，如表 1 所示，后被扩展为 4C 模型（增加了维护多个缓存之间的一致性导致的缺失，即 Coherence 缺失）。有趣的是，四种缺失的英文首字母都是 C。——译者注

了传统硬件。我们首先做了一个二乘二分类，其中新旧值存储在其中一个维度中，在另一个维度中可以检测到事务冲突。我们认识或意识到，每个“格子”对应一些在商业上成功的数据库管理系统，所以为了与尚无实例与之高效地⁸对应的那个格子对应，LogTM 得以研发并实现⁹。

关联 (Connections) 是研究的关键。史蒂夫·乔布斯 (Steve Jobs) 曾说：“创新就是将多个事物关联起来。”我的经验是可以通过与同事和学生合作来探寻关联的机会——我所有想法都得益于曾合作过的 160 位共同作者（如图 3 所示）。不要过分担心你的贡献被人瓜分了；相反，慷慨地分享想法，你的贡献经常会成倍增加并带来更多的合作，

表 1 事务内存的分类¹⁰

		Version Management	
		Lazy	Eager
Conflict	Lazy	OCC DBMSs Stanford TCC	none
	Eager	MIT LTM Intel/Brown VTM (on cache conflicts)	CCC DBMSs MIT UTM LogTM

⁸ “高效地”三字为译者添加。如表 1 所示，我们发现作者提出的 LogTM 不是位于表格的右上角（尚无实例与之对应），而是位于右下角（已有实例与之对应）。相对 UTM 等已有系统，LogTM 的复杂度较低，更高效了。也就是说，作者不是填补右上角那个完全的空白，而是从提升效率的意义上填补右下角的空白。在历史上，Flynn 分类法中也出现类似的情况（Michael J. Flynn. “Very high-speed computing systems.” *Proceedings of the IEEE*, 54.12(1967):1901-1909.）。Flynn 根据指令流和数据流的数量将计算机分为四类，依次为单指令单数据 (SISD)、单指令多数据 (SIMD)、多指令单数据 (MISD)、多指令多数据 (MIMD)。其中 MISD 尚无实例与之对应，就像表 1 右上角那样尚无实例与之对应。很多体系结构设计一般是从提升效率的角度填补 MIMD 那个格子的空白。——译者注

⁹ 这里作者举了三个关于分类的例子，其实还有很多成功的例子，比如现在研究比较热的忆阻器。1971 年时任伯克利大学教授蔡绍棠 (Leon Chua) 在研究电荷、电流、电压和磁通量之间的关系时，推断在电阻、电容和电感器之外，应该还有一种组件，代表着电荷与磁通量之间的关系，提出了存在第四无源电子元件的理论，并给出了相关的公理公式推导 (Chua Leon. “Memristor-The Missing Circuit Element.” *IEEE Transactions on Circuit Theory* 18.5(1971):507-519.)。直到 2007 年，在 HP 实验室证明了忆阻器的存在，此后国际上相关研究所和大学等加入对此元件的研究。另外，脚注 8 提到的 Flynn 分类法也是计算机体系结构学科关于分类的一个经典例子。——译者注

¹⁰ 此表原载于 Kevin E. Moore, Jayaram Bobba, Michelle J. Moravan, Mark D. Hill, and David A. Wood. “LogTM: Log-based Transactional Memory.” *Proceedings of the Twelfth IEEE Symposium on High-Performance Computer Architecture*, IEEE Computer Society, Austin, TX, Feb, 2006:1-12. ——译者注

¹¹ 此图为译者添加。

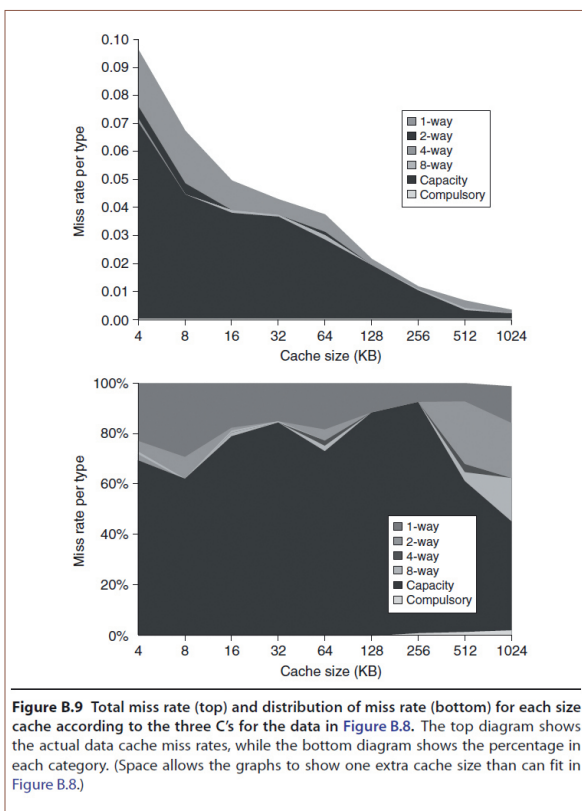


图 2 《计算机体系结构量化研究方法》教科书对高速缓存 3C 模型的介绍¹¹

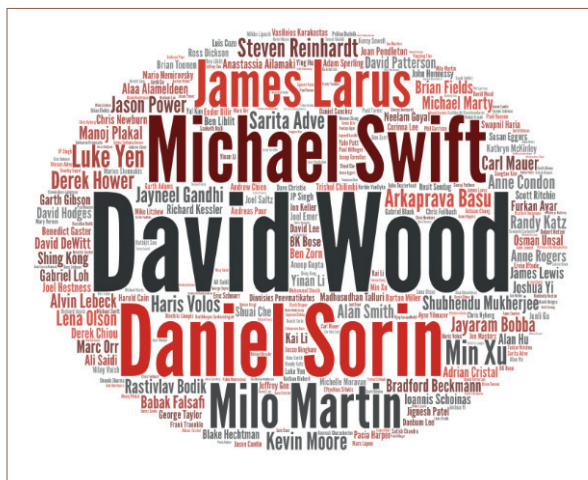


图3 希尔教授的160余位合作者

从而产生更有价值的研究。通过参加讨论和交谈也可以实现关联。在一次与巴特·米勒 (Bart Miller) 教授¹²关于软件数据竞争检测的讨论和交谈中, 我们忽然认识到数据竞争与弱/宽松的内存模型之间存在某种关联。经过一番艰苦卓绝的深入思考, 我们让这种关联更清晰明了, 建立了“顺序一致性内存模型”, 使得程序免于数据竞争。顺序一致性内存模型影响了随后中央处理器 (CPU) 和图形处理器 (GPU) 硬件的设计, 同时是 C++ 和 Java 语言的内存模型的基础。

模型 (Models) 对初始思考来说也很关键。好的模型通过抛弃次要的东西来关注本质性的东西。阿姆达尔定律 (Amdahl's law) 是计算机体系结构中最广为人知的模型。我向大家倡导瓶颈分析和 Little 定律¹³、M/M/1 队列。在一次学术休假中，我与谷歌进行了交谈，我们开发了针对移动系统芯片的 Gables 模型。Gables 来自这样的想法：片上系统带有 CPU、GPU 和数十个加速器，其中每个加速器可以运行 10 到 20

个用例（如视频录制或回放）。乔治·博克斯 (George E.P.Box) 说：“所有模型都是错的，但有一些是有用的。”虽然需要时间来证明 Gables 是否有用，但它已经展露出加速器级的并行性，我们预计它以后将被运用到许多计算机系统中。需要指出的是，在不同的层次上，对应的最合适的模型往往不同。

在我的经验中，最好的研究通常是简单的。我想引用爱因斯坦说过的一句话：“事情应力求简单，简单到不能再简单。”简单化通过避开不太重要的东西，直击本质能促进更大胆的工作。此外，即使是简单的计算机体系结构和系统的想法在实际部署产品中也会变得更为复杂。

步骤三：测试和优化假设：

按照科学的方法，以**结构化思维**来表达：

重复直到完成 {

制定一个或多个新假设（通过思考）；

设计一个或多个实验来检验假设 (通过思考);

预测实验结果（通过思考）；

运行实验并观察实际结果与预测的不同之处；

}

强推理 (Strong Inference) 科研方法首先会提出多个假设 (来促进并发地考虑问题, 同时缓解考虑问题过度依赖于“你”的假设), 然后设计实验去证实或证伪。也就是说, 测试这个过程不仅仅是收集数据。由于今天计算机体系结构领域的模拟实验通常运行得很快, 科研人员反而容易陷入先运行实验后建立假设的陷阱, 这往往会导致漫无目的、无法证实或证伪假设的实验。格雷戈尔·孟德尔¹⁴在创立遗传学时因为种植豌豆实验需要时间, 所以他能

¹²巴特·米勒为威斯康星大学麦迪逊分校计算机科学系教授，ACM Fellow，研究领域包括软件测试和二进制代码分析等。——译者注

¹³ 由麻省理工大学斯隆商学院教授约翰·利特尔 (John Little) 在 1961 年证明：在一个稳定的系统 (L) 中，长期的平均顾客数量，等于长期的平均到达速率 (λ)，乘以顾客在这个系统中的平均等待时间 (W)，即 $L = \lambda \times W$ 。原文见 John D. C. Little, (1961) “A Proof of the Queuing Formula: $L = \lambda W$,” *Operations Research*, 9, (3) 383-387。——译者注

¹⁴格罗戈尔·孟德尔 (Gregor Johann Mendel, 1822—1884), 奥地利帝国生物学家, 遗传学奠基人。他通过豌豆实验, 发现了遗传学三大基本规律中的两个, 分别为分离规律和自由组合规律。

进行深入的思考。虽然分类、联系和模型有助于初始思考，但模拟 (simulation) 或有时构建硬件有助于证伪或完善包含许多已知和未知因素的假设的实验。然而，应该仔细考虑计划使用或构建的模拟器。2000 年左右，大多数模拟器只是用户级的，而我们工业上的同事要求在包括输入输出 (I/O) 的操作系统上运行商用工作负载。出于这个原因，我们着手建立 GEMS 模拟器。为了简化工作并加速开发，我们“富有创造性地懒惰偷闲”，实现了 GEMS 的性能部分，而使用初始版本的商业 SimICS 模拟器来启动操作系统和支持外部硬件。这种共生是一个巨大的成功，但限制了我们与他人分享这个模型。幸运的是，(前) 密歇根的伙伴已经在 m5 模拟器中实现了全系统功能，并建议我们与他们合并以形成 gem5。作为许多任务的适当级别的模拟器，gem5 现在广泛用于学术界和工业界。

步骤四：螺旋迭代式研究

正如爱迪生所说，“我并不是失败了 1000 次，而是用了 1000 个步骤才发明了灯泡”，好的研究很少能通过捷径找到解决方案——论文中所谓的“捷径”往往是经历艰辛探索后才找到。相反，就像我们为 LogTM 变体所做的那样，一个人在徘徊，可能会遇到一些死胡同，因此必须在持久性和灵活性之间做好平衡。简化事情需要时间，就像我们研发顺序一致性内存模型所发生的那样。要接受有些工作无法发表的现实，例如我关于 SIMD 高速缓存模拟的工作和概率性写回的工作。因此，最后我建议：你需要用科学方法快速而坚定地开展研究；你需要知道并不是只有你自己会遇到挫折，所有的研究都会经历挫折。

译者后记

在翻译马克·希尔教授的这篇经验之谈的短文之后，译者想说三句话。第一，多有共鸣：可能是小同行的原因，感慨颇多，有很多共鸣，文中提到的元素周期表、阿姆达尔定律、Little 定律等，译者有了一些研究结果^{15, 16, 17}。第二，学会表达：这篇以博客呈现的短文貌似朴实无华，但逻辑清晰，内容要点切中要害，年轻的研究者需要学习这种风格，不要把简单的事情复杂化，而要把复杂的事情简单化，培养和提高包括中文和英文、口头和书面在内的表达能力。第三，学会研究：注意分析，善于提出多个假设，会推理，善于设计实验，能调程序的 bug，能综合运用四种研究范式对应的研究方法，能吸取合作者的合理意见，有主动性，经受挫折但愈挫愈奋，直至成功。 ■

作者：



马克·希尔 (Mark D. Hill)

2019 年埃克特 - 莫契利奖获得者，IEEE/ACM Fellow。威斯康星大学麦迪逊分校计算机科学系教授。主要研究方向为并行计算机系统设计、存储系统设计和计算机模拟。

译者：



刘宇航

CCF 专业会员，CCCF 特邀译者、特邀专栏作家。中国科学院计算技术研究所副研究员。主要研究方向为计算机体系结构、高性能计算、大数据、智能并发系统。
liuyuhang@ict.ac.cn

¹⁵ Yuhang Liu, et al. Gene-Patterns: Should Architecture be Customized for Each Application? <https://arxiv.org/abs/1909.09765> 此文研究了访存模式的“元素周期表”。

¹⁶ Yuhang Liu, Xian-He Sun. C²-bound: A Capacity and Concurrency driven Analytical Model for Manycore Design. *Proc. of International Conference for High Performance Computing, Networking, Storage and Analysis 2015 (SC'15)*, Texas, Austin: 1-11, 此文研究了 Amdahl's law 及其扩展形式在存储受限对多核芯片设计的影响。

¹⁷ Yuhang Liu, Xian-He Sun. CaL: Extending Data Locality to Consider Concurrency for Performance Optimization. *IEEE Transactions on Big Data*. 2018.4.30 5(2): 273-288. 此文研究了 Little's law 在存储系统中的应用。