DOI:10.1145/2856453

如需阅读配套的论文，
请访问 doi.acm.org/10.1145/2856449

技术视角

在负载迁移的计算中捕捉谎言（与错误）

Michael Mitzenmacher，Justin Thaler

考虑有位客户 想运行一个计算机程序处理某个数据集，但他又缺少完成该处理任务的能力。因此，客户或者验证者需访问一个强大的但不受信任的证明者。证明者不仅必须运行程序，返回输出，还必须提供正式保证，保证输出是正确的。该框架体现了种类繁多的真实世界场景的本质。验证者和证明者可以分别作为客户端和商业的云计算服务的模型，或作为 CPU 和快速但可能有错的协处理器的模型，或为外围设备和大型机的模型。

如果忽略证明者，只在本地运行程序，那么验证者如何才能获取正确性的保证呢？能否实现这点并不显然，至少如果不对“作弊”的证明者的行为做出强假设的话。实际上，研究人员已经提出了依赖上述假设的多种解决方案，其中包括复制、审计以及使用可信的硬件。相比之下，在名为可验证计算（VC）的研究领域中，目标更为宏伟：它所寻求的解决方案并不需要对作弊的证明者的行为做出假设。

在二十世纪八十年代末和二十世纪九十年代初，计算机理论科学家发现了效率惊人的 VC 协议。虽然这些协议各自特点不同【它们的名称为交互证明、概率可验证证明（probabilistically checkable proofs）及论证系统】，但是它们都提供了下列保证：如果证明者返回的输出不正确，那么无论证明者付出多大努力向验证者说明它没有说谎，验证者捕捉到证明者说谎的概率都会很高。不仅如此，这些协议保证，验证者基本上只要读取输入，证明者基本上只要执行程序。

这些发现给计算复杂度理论带来了变革。研究人员现在仍在探索它们带来的诸多影响。但是，尽管这些渐近令人瞩目，研究人员仍认为所有的这些协议都远离实际，他们的理由也相当充分。直接实现这些协议会产生喜剧性的、具体的高开销——即使是相当小的计算，证明者也需要数百万年才能证明它们的正确性；同时，只能在真正庞大的输入面前，验证者才能相对本地执行节省时间。

不过，在最近几年这一观点受到了挑战，有若干研究团队发展了开销骤降的 VC 协议。这些团队追寻了不同的道路，使用了不同的理论方法。由此得出的实现综合了算法方面的改进与系统方面的成果，以便把开销降到接近真正实用的水平。

后文描述的 Pinocchio 系统改进了 Gennaro 等人提出的，重要的理论进步成果。[1] 综合考虑，这两项成果代表了速度、通用性和功能方面的巨大改进。Pinocchio 提供了非交互的论证系统，它支持使用 C 子集编写的程序，可按照可验证的方式自动执行该程序。Pinocchio 的验证者会执行一次性的预计算，以基于拟执行的计算构造一个公钥；如果相同的计算机程序使用多个输入运行，则可对这些输入使用相同的公钥。Pinocchio 的证明者所产生的证明相当短（288 字节），很快就能验证。后文的作者们使用了若干测试程序来展示系统的能力，这些程序涵盖了从矩阵乘法到格子气仿真（lattice gas simulation）的范围。

在更广的背景下研究 Pinocchio 是值得的。已实现的各种 VC 协议提供了表达性、特性和效率之间的各种取舍——针对这些取舍，Blumberg 和 Walfish 提供了详细的对比。[2] 一般来说，具体应用时，交互证明的通用性最差，但开销最低。论证系统，特别是类似 Pinocchio 的非交互式论证系统，开销更高：Pinocchio 的若干开销仍然非常高，特别是证明者的运行时间。另外，用于验证者的一次性预计算可能比本地执行的开销高几个数量级，这意味着需要为该验证者准备很多输入，以节省工作量。但是，伴随这些开销的是通用性和特性方面的重要改进。具体来说，Pinocchio 支持的关键特性是零知识，关于这点最好使用举例说明：假设某个计算机程序接收两个输入，一个来自验证者，另一个来自证明者，且证明者的输入是敏感输入。人们可能希望该证明者运行程序，向验证者提供回答，同时不揭示与证明者的输入有关的任何额外信息。标准的例子是，某人想与他人比较工资，但是不能泄露其他人的实际工资。Pinocchio 是第一个实现了提供这种零知识证明的系统。

随着研究人员持续地减少 VC 协议的开销，或许可以真正地把它们实际用于各种应用，从而为处理真实系统中的信任和正确性问题提供一种新的途径。但是，关键点是，如果多方都拥有敏感的输入，那么除了零知识证明之外，可能别无选择（相比之下，当输入不敏感时，本地执行总是一种相对于外包计算的另一选择，虽然它并不吸引人）。因此，即使效率的提升逐渐变小，在某些重要的场景中，类似 Pinocchio 的系统仍然是唯一的选择。在这些场景中，后文作者们实现的效率提升至关重要。而且，在类似 Pinocchio 的系统中，我们观察到的各种开销或许已经可以接受。

参考资料
1. Gennaro, R. et al. Quadratic span programs and succinct NIZKs without PCPs.EUROCRYPT, 2013, 626–645.
2. Walfish, M. and Blumberg, A.J. Verifying computations without reexecuting them:From theoretical possibility to near-practicality.*Commun.ACM 58*, 2 (Feb. 2015), 74–84.

Michael Mitzenmacher 是马萨诸塞州剑桥市哈佛大学计算机科学教授。Justin Thaler i 是纽约州纽约市雅虎实验室的研究科学家。

译文责任编辑：陈海波