# 对超级计算机性能分析工具的认识与研究

刘 旭
美国威廉玛丽学院

**关键词：**超级计算机 性能分析工具

**编者按：**中国超级计算机研制水平已处于世界领先，超级计算机应用也在去年实现突破，首次获得了戈登贝尔奖(Gordon Bell Prize)。在中国超级计算机研究开展得如火如荼之际，本刊特别邀请美国威廉玛丽学院的刘旭教授分享他对超级计算机系统软件的认识与研究体会。刘旭教授长期从事超级计算机的系统软件研究，取得了多项出色的成果，在SC、ASPLOS、PPoPP等顶级会议上发表了一系列论文，并获得了SC'15唯一的会议最佳论文、ASPLOS'17会议最佳论文提名。

超级计算一直是各大国之间角逐的领域。中国在硬件和软件两个领域都取得了举世瞩目的成就。在硬件领域，中国制造的超级计算机已经连续多年问鼎全球最快的计算机。在软件领域，中国科学家开发的软件获得了2016年超级计算软件的最高奖——戈登贝尔奖(Gordon Bell Prize)。与此同时，为了与中国的超级计算机竞争，美国也在争分夺秒地进行着CORAL项目。此项目是由美国能源部下属的多个国家实验室和工业界紧密合作，计划在2017年研发和部署的最新超级计算机。

超级计算机的硬件和应用软件分别是整个超级计算机系统的最底层和最上层，在它们之间还存在着我们称之为软件栈(software stack)的系统软件，包括编译器、并行运行环境、作业调度系统、性能分析和调试工具等等。系统软件能紧密结合硬件和应用软件，进而提高硬件使用效率和软件运行速度。长期以来，有效的系统软件是科研工作者研究的热点。我的研究领域也是超级计算机的系统软件，具体讲就是程序性能分析工具。在本文中，我将介绍自己对本领域的一些认识，分享自己的研究体会。

## 对研究领域的认识

### 研究领域——什么是程序性能分析工具

程序性能分析工具(profiler)是通过静态(static)或者动态(dynamic)程序分析来指出程序运行时的性能瓶颈，并给用户（程序开发者或者编译器）提供反馈来进行优化。通常，编写一个高效的应用软件是非常困难的，特别是对于动辄有几百万行代码的复杂的超级计算并行应用程序。很多并行程序会遇到各种性能问题，例如，软件问题（低效的算法，不合适的数据结构），编译器问题（编译器低效的生成代码），运行时问题（并行的开销，负载不均衡，线程进程同步和通信的开销，内存缓存和带宽的低效使用）等等。通常情况下，寻找这些性能瓶颈需要大量的人力。为了提高效率，研究人员一般依赖性能分析工具来对程序进行自动化的分析。

现有的部署在超级计算机上的主流性能分析工具（英特尔的商业工具 VTune，俄勒冈大学 (University of Oregon) 的 TAU，莱斯大学 (Rice University) 的 HPCToolkit，欧洲的 Scalasca 等等）主要依赖动态程序分析，同时也会用一些静态信息作为辅助。动态分析的工作流程首先是运行一遍待分析的程序，同时运行分析工具动态的监测程序来收集相关的性能指标。一旦程序运行结束，性能工具把所有的性能数据写入文件或者数据库，以待后续分析。一般情况下，性能分析工具会有一个线下的分析模块来处理所有的性能数据，并将其关联到程序的源代码中。用户进而根据分析结果来优化程序。如果仍然得不到预期的性能，用户可以重复之前的流程来逐步优化各个性能瓶颈。

## 论文评审——什么是好的性能分析工具

长期以来，关于程序性能分析工具的论文主要发表在 SC、CGO、PPoPP、PLDI 和 IPDPS 等会议上，并且多次被会议委员会评为最佳论文。近些年来，一些顶级的系统会议也乐于接收有关性能分析工具的文章。例如，SOSP'15 收录了马萨诸塞大学 (University of Massachusetts) 的 COZ 性能分析工具。COZ 可以用来分析并行程序的关键路径 (critical path)，进而指出和性能紧密相关的关键代码来优化。这篇有关 COZ 的论文被 SOSP'15 评为最佳论文。另外，ASPLOS'17 收录了我们的 RedSpy 性能分析工具的论文。RedSpy 是用一个细粒度的分析工具来分析程序冗余的内存写入 (memory write) 指令。这篇有关 RedSpy 的论文被提名为 ASPLOS'17 的最佳论文。

根据我投稿和审稿的经验，评价性能分析工具比较直接明了。一篇高档次的论文一般都具有以下几个特点：第一，**研究贡献**。这些论文都能发现现有的工具不能发现的性能问题，或者付出较小的代价来发现已有的性能问题。这一般需要性能工具采用一些创新的检测或者分析方法。第二，**系统实现**。这些论文都详细阐述了性能工具实现中遇到的各种问题和挑战，比如如何降低性能分析的开销，如果处理多线程和多进程的程序，如何更直观地给用户提供分析结果。第三，**系统评测**。对工具详细的性能评测包括工具运行时的时间和内存开销，以及是否能发现已有的相关工具所不能发现的性能问题。同时系统评测还包括有说服力的案例分析 (case study) 来展示工具的创新性和可用性。下面我展开谈一谈案例分析。

什么样的案例分析是有说服力的呢？一方面是要分析业界通用的程序集，例如 SPEC、PARSEC、NPB，以及一些美国能源部国家实验室发布的程序集，如 Sequoia、CORAL、APEX。这些程序集大多已经发布了十年以上，并经过了多次优化。同时，当前很多研究人员也致力于优化这些程序，并且发表优化的结果。可以说，这些程序集已经成为评估硬件、编译器以及性能工具的标准程序集。如果一个性能分析工具仍然能在这些程序集里找到之前没有发现的优化机会，那么这个工具就是有创新的。另一方面，如果一个性能分析工具能找到现有重要的应用程序上的性能问题，把它们作为案例分析也是非常有说服力的。比如，我们之前在美国能源部下太平洋西北国家实验室开发的计算化学软件 NWChem 里发现了性能问题，并给出了优化方案，使得有 600 万行代码的 NWChem 性能提升了 1.5 倍。用既真实又重要的应用程序作为案例分析，既能证明工具的鲁棒性，又能引起业界审稿人的兴趣。

## 发展前景——对性能分析工具的展望

性能分析工具在超级计算机的软件栈中占据重要的位置。近些年，为推动性能分析工具的发展，业界每年都会在加利福尼亚州的太浩湖 (Lake Tahoe) 举行为期四天的研讨会，讨论未来性能工具的发展趋势。与会的专家有来自工业界的谷歌、克雷、英特尔和 IBM 等，有学术界的威斯康星大学、莱斯大学、德克萨斯州大学奥斯汀分校和俄勒冈大学等，有国家实验室和超算中心，如劳伦斯·利弗莫尔国家实验室、于利希 (Julich) 超级计算机中心和巴塞罗那超级计算机中心等。近些年来，北京航空航天大学钱德沛教授团队也来参加这个研讨会，

和国外同行进行研讨。根据自己的研究经历和多次参加工具研讨会得到的一些经验，我认为性能分析领域未来会朝着以下几个方向发展：

**1. 可扩展到百亿亿次 (exascale) 超级计算机的性能分析工具。**设计可扩展在百亿亿次超级计算机上运行的应用程序非常有挑战性。这就需要性能分析工具指出并消除影响可扩展性的性能瓶颈。但是，性能工具运行在大规模并行计算机上面临着如何在收集到的大量数据中找到性能问题并展示给用户的挑战。同时，性能分析的开销要足够小进而不能成为运行时的可扩展性瓶颈，否则工具收集到的数据很难暴露出低效的软硬件交互，并且用户也不会很容易接受使用这个工具。

**2. 语义级别 (semantic level) 的性能分析工具。**现有的性能分析工具大多数能找到程序的热点并且关联到程序的循环体或者函数。热点分析固然不可或缺，但是提供的信息相对较少。比如，性能工具指出一个函数耗费了大量的运行时间并造成了大量的缓存缺失，但是分析出真正的原因并给出针对性的优化策略还需要人工进一步分析。所以，将来的性能分析工具会向语义分析的方向发展，并提供高层次的算法，数据结构信息只作为热点分析的补充。这样的性能工具将会给出更准确的判断来确定性能瓶颈的位置，并且全自动地给出性能瓶颈的原因，作为程序优化的参考。

# 研究实践中的体会

我在中国科学院计算技术研究所读了三年的硕士，在美国莱斯大学读了五年的博士，目前在美国威廉玛丽学院 (College of William & Mary) 从事教学研究已有三年。如何取得高质量的研究成果，我有以下三点深刻体会。

## 1. 重视研究的积累

现有的主流性能分析工具都有长时间的开发积累。例如，英特尔的 VTune，俄勒冈大学的 TAU，以及莱斯大学的 HPCToolkit 都有十年以上的开发经历。这样长时间的开发经历让这些性能分析工具愈加完善，同时一些新的想法基于已有的开发框架也更容易实现，更有说服力。我现在的研究主要集中在两个性能分析工具上，一个是 HPCToolkit，另一个是 CCTLib。基于这两个工具，我们都发表了一系列论文，并且获得主要会议的认可。

基于 HPCToolkit 上的研究是我在博士期间所做研究的一个延续。我在 HPCToolkit 中加入了以数据结构为中心的性能分析，进而能更轻量级地分析和内存相关的性能瓶颈。在这项工作之前，传统的性能工具需要通过插桩 (instrumentation) 和模拟 (simulation) 来得到内存相关的性能瓶颈。这些传统工具一般要产生十几倍到几百倍的运行时间开销 (overhead)，非常不利于在检测大型超级计算程序上应用。我的工作是通过使用硬件计数器 (performance counter) 把这个时间开销降低到 10% 以下。我们在 SC'13、PPoPP'14、PACT'14、SC'15 和 HPDC'16 发表了一系列论文，阐述了这个轻量级的性能分析工具对不同内存性能问题的分析。其中我们的论文在 SC'15 上斩获了唯一的会议最佳论文，会议委员会一致认为我们的性能工具为解决超级计算机系统中内存墙 (memory wall) 问题提供了强有力的指导。

CCTLib 是我们为性能分析工具开发者所研发的一套细粒度性能分析工具框架。基于 CCTLib，我们开发了一系列的细粒度的工具来寻找程序中的冗余计算以及冗余的内存操作。对于这些冗余的操作，传统的性能分析工具很难找到它们。我们在 CGO'14、PACT'15 和 ISMM'16 发表了一系列的论文来探讨 CCTLib 的应用。我们最新的一篇论文发表在 ASPLOS'17，并被提名为会议最佳论文。专家委员会认为，我们解决了一个在性能分析上的基础性问题 (fundamental problem)。

## 2. 开发有实际用途的工具

若要让用户认可一个性能分析工具并真正应用在自己的研究或者开发环境中，这个性能分析工具必须要有自己独有的特点。现有的主流工具都具备这个特点。比如，VTune 能更好地利用英特尔处理

器上的性能检测单元 (PMU)；TAU 支持不同级别的程序插桩，可以快速地移植到各个不同的系统平台；HPCToolkit 支持准确的轻量级的调用堆栈展开 (call stack unwinding) 以及性能瓶颈原因分析 (root cause analysis)；CCTLib 可以支持细粒度的程序分析，进而可以指出冗余的和不必要的操作。正因为这些不同的特性，这些工具根据不同的需要才有不同的应用场景。

作为能够实用的工具，除了具备新颖的分析方法，直观地展示分析结果也非常重要。这是因为大多数用户是应用程序开发者而不是计算机系统的专家，让这些用户感觉到工具“好用”是非常关键的。通常，用户习惯使用一个工具后，他们很少再去更换使用别的工具。相反，如果一个工具给用户的第一印象不好，那么以后也很难说服用户改变他们的看法。基于这个原因，大多数工具研发人员都会设计直观的图形界面 (graphic user interface) 来直接显示程序的性能问题。

## 3. 对研究结果的宣传

众所周知，发表高档次的论文并在会议上演讲是推销自己研究成果行之有效的方法。除此之外，还有四个方法能够有效地扩大自己研究的影响力以得到更广泛的关注：系统的开源，和用户的沟通，在会议上组织工具的教程，以及积极参加标准制定委员会。我通过我的博士导师约翰·梅勒 - 克拉米 (John Mellor-Crummey) 教授的经历来介绍这几种推销方法。

约翰·梅勒 - 克拉米教授是 HPCToolkit 开源工具的创始人和维护者。我们提交代码都需要通过他的代码审核来保证 HPCToolkit 的代码质量。同时，他又是一个非常优秀的程序员，长期工作在代码编程的第一线。开源的 HPCToolkit 也得到了更多用户的使用和认可。

约翰·梅勒 - 克拉米教授每年都会到美国能源部下属的各个国家实验室，和那里的超级计算应用开发者一起研讨。他会详细地询问应用开发人员对性能工具的需求以及他们对自己开发的应用程序上的性能疑问。之后，他会有针对性地开发 HPCToolkit，并推销给这些用户去使用。经过多年的努力，HPCToolkit 已经部署在绝大多数国家实验室的超级计算机上，并得到程序开发人员的广泛应用。

约翰·梅勒 - 克拉米教授会在一些主流会议上组织 HPCToolkit 的教程，从与会者中吸引更多的用户。受到导师的影响和启发，我也在 2017 年的 CGO 上组织了 CCTLib 的教程。我们花费了近一个月的时间准备演讲稿、性能工具的安装使用教程和直观的教程用例。组织这个教程能更快地扩大 CCTLib 的影响力。我们由此吸引了美国和欧洲的一些工业界、学术界的与会开发人员参加教程，并且让他们使用和参与开发了 CCTLib。

约翰·梅勒 - 克拉米与我自己都加入了 OpenMP 旗下的关于工具接口的委员会。OpenMP 是超级计算中应用最为广泛的标准的共享内存并行编程模型之一。我们致力于设计一套轻量级的用户编程接口来让性能工具更好地分析 OpenMP 程序。经过近五年的努力（2012 年初到 2016 年底），我们成功地在 OpenMP 标准中加入了我们设计的接口——OMPT。HPCToolkit 成为第一个支持 OMPT 的性能工具，影响了整个领域的发展。

随着超级计算硬件和应用软件的迅猛发展，未来对性能工具这样的系统软件工具会有更多更高的要求。我希望能有更多的研究人员能加入这个领域，致力于提高计算机的软件效率和硬件使用率。 ■

**刘 旭**
美国威廉玛丽学院助理教授。获得SC'15最佳论文、ASPLOS'17最佳论文提名
主要研究方向为并行优化，性能分析工具。xl10@cs.wm.edu

2017 CCF CCSP大学生计算机系统与程序设计竞赛将于10月在福州举办