一名系统研究者的攀登之路

陈海波
复旦大学

关键词：计算机系统 论文 批判性思维

引言

写好计算机系统领域的研究论文非常不容易，不仅需要有非常好的想法，还要证明这个想法的可行性和应用效果。因此，准备一篇论文的周期通常应在一两年以上。计算机系统领域的学术会议通常每年只接收二十多篇研究论文，以保证学术交流会（single-track session）对每篇论文进行充分的讨论。2011年计算机系统的几大会议——SOSP（OSDI在偶数年召开）、Eurosys、USENIX ATC接收的研究论文总计只有79（28 + 24 + 27）篇。较长的投稿准备周期与较少的论文接收总数使得在计算机系统领域里发表会议论文异常困难。长期以来，这些学术会议的论文被美国、欧洲的一些著名高校、科研机构和公司研究院所占据，我国乃至亚洲地区学者在这些会议上发表论文的数目极少。据统计[1]，截至2010年底，亚洲学者40年来在SOSP上独立发表研究论文的数目仍然为零。

作为一名计算机系统领域的研究者，在计算机系统相关的高水平学术会议上发表研究论文无疑是非常重要的。自2004年起我开始了计算机系统的相关研究，2011年终于与复旦大学并行处理研究所的学生分别在EuroSys 2011、USENIX ATC 2011与SOSP 2011上发表了研究论文或被接收了论文。在论文撰写与投稿的过程中，我们经历了挫折，也积累了一些经验。在此我非常荣幸地将我在计算机系统领域开展研究的经历与感受与大家分享，希望对目前正在从事系统方向研究的研究生有所启发。

研究经历

我接触计算机研究是在2002年的7月，大学二年级结束后的暑假。一次偶然的机会，我接到臧斌宇教授的邀请，加入了复旦大学并行处理研究所的研究团队。当时我参与的是一个与编译相关的项目，主要的工作是为飞利浦Trimedia芯片的超长指令字（very long instruction word，VLIW）指令集GCC（一套由GNU开发的编程语言编译器）移植后端，再进行优化。2003年下半年开始，我们开展了可重配置体系结构的研究，探索如何为媒体与通信应用设计可重配置的处理器结构。

2004年英特尔公司的王文汉博士讲到系统虚拟化将会在十年内流行并产生重大影响。很荣幸，我从读研究生开始就在臧斌宇老师的安排下从事了系统虚拟化的探索工作。当时国内虚拟化研究工作虽然刚刚起步，但是很多需求已经显现出来了。上海电信相关部门的负责人当时提出了面临服务器与服务整合和提高电力供应的问题。当时我们觉得用系统虚拟化应该是一个非常好的解决方案。我们的第一个切入点就是如何去度量企业应用在虚拟化的情况下的性能与服务质量的问题。为此，我们选择了很多的基准测试程序包括TPC-C与TPC-W等来对比分

[1] http://xiao-ma.com/sohof/

析虚拟化层可能存在的问题。

当时我们的想法是用客户操作系统与虚拟机监控器进行配合以减少虚拟环境下性能开销与服务的不确定性。从2004年11月开始设计，到2005年5月我们终于有了一个雏形并准备测试。就在此时，我在查看相关研究动态的时候，突然发现被VEE 2005接受的一篇论文跟我们的想法很像。当时给我的感觉是："撞车了！"真有种万念俱灰的感觉，半年多的工作白费了。我在惋惜、灰心的感觉中度过好几周才慢慢恢复过来。

我们开展的第二项工作是基于系统虚拟化的动态更新系统。某天在浏览电子公告板（BBS）的时候，Windows XP系统突然跳出一段提示："您的补丁已经下载完毕，请重启您的电脑以应用更新。"这段提示不断地跳出来让人觉得很烦。晚上睡觉我思考接下来的研究方向时，迷迷糊糊之际突然想到是否可以用虚拟机监控器来为操作系统实现动态更新，而不需要重启计算机。因为传统的操作系统直接运行于硬件层。操作系统在运行时需要修改自己的状态。如果操作系统自我动态更新的话就涉及到自我状态与被更新状态的相互影响的问题，这就有点像鸡生蛋还是蛋生鸡的问题，需要一个解靴（bootstrap）的办法。因为在虚拟化的环境下，操作系统实际上是运行在虚拟机监控器上的，完全可以用虚拟机监控器来控制客户操作系统的状态以保证更新过程中的状态一致性。基于这个想法，我们设计了一个"双向同步写穿"的协议，维护更新过程中的新旧状态的一致性。这项工作最后被VEE 2006接受，当时在大会演讲的时候引起了较长时间的讨论。会上美国伊利诺伊大学的维克拉姆（Vikram Adve）教授还特意跑到我面前说这是个非常酷的工作。后来VMWare公司也开发出了基于虚拟化的补丁管理系统。在这个工作的基础上，后来我们又开发了第一个支持多线程应用数据结构动态更新的系统POLUS（ICSE 2007）。之前的系统需要在更新的时候一直将操作系统运行在虚拟机监控器上，从而带来了较大的性能开销。我们又设计实现了动态虚拟化系统，从而使操作系统只有在需要的时候才会被虚拟化。为此，我们获得了ICPP 2007的最佳论文。

由于之前进行过一段时间的编译与体系结构的工作，后来我就思考是否可以将现在进行的系统领域研究与它们进行结合。于是2007年初就开始了利用动态信息流加强系统安全的工作。当时利用软件进行动态信息流跟踪的主要问题是性能开销太大，然而基于硬件的动态信息流系统则需要较多目前尚不存在的硬件扩展。由于安腾处理器为了支持猜测执行为每个寄存器增加了一个状态位来跟踪猜测执行过程中的异常情况，于是我们就尝试利用猜测执行硬件支持来实现高效的动态信息流跟踪，并且实现了一个叫SHIFT的系统（ISCA 2008）。在SHIFT工作的基础上，我们又探索这样的特性是否可以解决其他问题。后来，我们设计实现了基于动态信息流的程序控制流混淆系统，通过用户态异常来隐藏程序控制流，从而达到抗逆向分析的效果（MICRO 2009）。

2008年初，美国麻省理工学院的M.弗兰斯·卡肖克（M. Frans Kaashoek）院士对微软亚洲研究院进行为期半年的访问。我非常荣幸地和其他同学一起在弗兰斯的指导下开展了为期半年的研究工作。期间主要探索如何为众核平台设计性能可伸缩的操作系统。在这期间，我与其他同学一起设计实现了Corey操作系统（OSDI 2008）。我负责的是内存密集多核应用的行为分析以及对应的内核抽象支持以提供可伸缩的性能。经过与弗兰斯一起工作，我学习到了非常多的东西，也有非常深刻的体会，后面将具体提到。

2009年1月博士毕业后我留校继续在并行处理研究所带领系统研究组开展相关计算机系统的研究。在工作中，我们针对众核环境中的软件运行栈进行分析与优化，以提高其性能可伸缩性；同时针对云计算环境下的用户数据的安全性与隐私性开展研究。期间与实验室成员一道设计与实现了Tiled MapReduce（通过分块等办法改进MapReduce的编程模型，PACT 2010）；第一个可移植的并行全系统模拟器COREMU（PPoPP 2011）；基于操作系统簇集的众核操作系统可伸缩解决系统Cerberus（Eurosys

2011）；面向JVM平台的高效执行重放平台ORDER（USENIX ATC 2011）与基于嵌套虚拟化的云平台数据保护系统CloudVisor（SOSP 2011）。

感触

九年的研究生涯，我与实验室成员得到了不少的教训，也积累了一些经验，感受颇深。在此结合计算机系统领域的研究把自己的体会介绍给大家：

批判性思维

系统研究中的自由性使系统研究很容易走向“重新发明轮子”或者“发明一个不相干的轮子”的误区。因此，系统研究尤其需要批判性的思维。在与弗兰斯一起工作中，弗兰斯就特意告诫我思考问题需要极度的批判性（super-critical）。我现在还很清楚地记得当时我向他介绍我们发表在ISCA 2008上的论文时候的情形：刚开始介绍论文的意义时，我就被他的一连串问题给难住了。“为什么要采用动态信息流跟踪来做攻击检测？”我举了Buffer Overflow的例子。弗兰斯反驳说，“Buffer Overflow已经有很多办法来解决了，如地址空间随机话与不可执行栈。”我就举了SQL注入的例子。弗兰斯又反驳说，“为什么不能用静态分析的方法来解决？”后来我知道，弗兰斯对这些问题都是非常了解的，他希望通过问答的方式看到我在这个过程中对涉及到的问题是否深入地、批评性地思考过了，而不是简单地接受其他人或论文上的观点。

扎实的基本功

计算机系统偏向于实践，强调的是解决问题的整体能力。因此，比较全面的知识面，扎实的系统编程能力与快速学习能力将对开展系统方向的研究至关重要。而这些能力往往需要较长时间的培养。在这里，我要感谢复旦大学软件学院的以实践为导向的课程体系，为我提供了比较扎实的基本功，使我在本科阶段就积累了比较好的操作系统、体系结构与编译系统等的设计与实现能力。在国外许多著名高校，都是将教学与研究联系得非常紧密的。例如，目前我在教授操作系统课程时采用一个基于显示内核（Exokernel）的JOS作为操作系统的课程实验(源自MIT的课程代号为6.828的操作系统课程)。而2008年我们的Corey操作系统（OSDI 2008）就是以JOS为基础，进行面向众核操作系统的性能可伸缩性的扩展，来设计多种抽象为众核设计操作系统。而麻省理工学院的分布式系统的课程项目则是由当时影响了很多分布式文件系统设计（包括Google文件系统）的Frangipani（SOSP 1997）而来。这样，他们就很容易通过课程实践的项目为学生提供较强的基本功，从而很容易就能将课程上学到的知识应用到研究项目中去。

发散式思维

在研究过程中，如果问题A得到解决，那么是否可以解决问题B？如果问题A通过方法1得到解决，是否还可能通过方法2进行解决呢？各种解决方法各有什么样的优缺点？在研究过程中就需要不断地进行这样发散式的思维。例如，在使用虚拟机更新操作系统的方式提供操作系统的动态更新后，是否可以将类似的想法应用到多线程应用呢？于是我和其他组员一起设计与实现了第一个支持多线程数据结构更新的动态更新系统POLUS（ICSE 2007）。同样，在完成使用动态信息流跟踪的研究后，我就在想是否可以利用它来解决其他问题呢，于是我们就设计实现了基于信息流的控制流混淆技术。同样，在完成Corey的工作后，我们尝试一方面为众核提供更好的开发工具COREMU（PPoPP 2011），另一方面将Corey中的一些功能应用到日用操作系统中去（Cerberus, Eurosys 2011）。

开阔的视野与专注的研究

这看起来更像是一个采用深度优先还是广度优先进行学习与研究的例子。看似一对矛盾体，因此需要去做动态平衡。我个人的体会是，对研究生而言，在一段特定的时间内需要有一个专注的研究点。在选择研究点的时候需要批判性的思考。这样

的一个研究点是否值得去做？而一旦这个研究点确定下来了，就要持续深入地去研究一个相对较长的时间，直到可以很肯定地告诉自己这个研究点的问题已经全部解决了，否则就不轻易放弃。在专注的过程中，还需要以一个开放性的心态去关注其他领域的动态，通过学术会议、报告与小组讨论等方式去获取新的信息。但如果在这个过程中有了新的想法，先别急着去改变自己的方向，而是先将其记录下来，隔段时间拿出来思考一下，然后在当前专注的研究点有了结论后再去尝试新的想法。我在指导学生的过程中也碰到过一些非常聪明的学生。他们的想法非常多，但大部分想法都没有经过深入地带批判性的思考，就会出现这周做系统安全的相关研究，下周又去探索多核操作系统的性能可伸缩性，再下周又去探索分布式系统了。出现这种情况，我通常建议先专注于一个研究点，直到这个研究点有结论了以后再去探索其他的研究点。

认真、逻辑严密的写作

系统领域对写作非常重视，因为大家普遍认为，严谨细致的写作是严谨细致思维的体现。因此，所有系统领域的顶级会议在接受论文后，都会给每篇论文指定一个指导（Shepherd），督促与帮助作者完成论文的最终版本的工作。比如，我们与MIT合作的Corey论文被OSDI 2008接受后，又重新写了一遍论文，系统的设计、实现与实验也重新做了一遍。尽管我们Eurosys 2011论文的6位审稿人都给出了肯定的评价，但我们在准备最终版本的时候仍然修改了五遍。

在这个过程中，我的体会是，中国学者的英文写作可能会存在一定劣势。计算机系统方面的英文写作最重要的是如何理清思路与逻辑，以严谨、清晰的方式将所要表达的意思传递出来。因此对整篇论文、每个章节、每个段落乃至每个句子的逻辑与结构都要进行仔细地推敲这是非常重要的。

耐心

由于计算机系统领域研究的周期相对比较长，因此切忌急功近利。例如，Cerberus、COREMU与CloudVisor系统的周期都接近两年。此外，我们还要沉得住气，尤其是要全面系统地看待他人的工作。系统领域很多研究需要平衡很多因素，强调解决问题的方法应简单与优雅，这样很多非常有影响力与实用价值的论文看起来比较简单。所以很多同学（包括学生时代的我）很容易觉得计算机系统方面的论文很容易就搞定了。我看到过一些同学（包括过去的我）一直盯住一些会议的截止日期，在还有一个月到三个月的时候从零开始，抱一堆相关领域的论文，试图在短时间内搞定一个顶级会议。这种方式到最后基本上都会失败。

由于系统强调实用性，大部分系统领域的研究论文都要有工作原型系统的实现以验证其想法的可行性。所以，经常有可能出现一个bug需要几周的时间来调试。在调试的过程中需要不惧怕艰难的心态，而不是碰到一点困难就在第一时间放弃。弗兰斯在2011年获得ACM-InfoSys奖的访谈时也谈到，系统方向研究最重要的是兴趣与恒心（Persistence）。

面对拒稿

由于计算机系统领域研究人员的极度批判性思维以及总体较少的论文数目，计算机系统领域论文被拒的情况便是家常便饭。如何面对论文被拒，这就需要培养一个良好的心态去面对拒稿，分析其中的原因并进行改进。在学生时期，我们的一篇使用虚拟机监控器来保护不被信任操作系统应用的论文投到了SOSP 2007被拒了。后来我去VMWare公司位于Palo Alto的总部实习的时候，才知道被拒的一个重要原因是因为当时VMWare公司与斯坦福大学也投了一篇具有类似想法的论文，因此程序委员会委员觉得不能接受两篇想法相似的论文，所以两篇都被拒了。后来VMWare公司与斯坦福大学的论文发表在ASPLOS 2008上。这让我感觉非常沮丧，觉得我的工作与别人发生了冲突，已经没有意义了。因此当弗兰斯劝我再在原有的基础上进行改进创新重投OSDI 2008时我都没有信心。现在回想起来觉得当时没必要那么灰心，其实是可以将工作做得更加彻底

的，也许很有可能就有新的发现。同样，我的第一个工作也不应该就此放下，可以通过与同行的工作进行对比，将问题了解得更加清楚，从而获取新的收获。

结语

通过9年的科研工作，我经历了很多的失败，也收获了成功的喜悦。最为重要的是，在这个过程中我学到了很多相关的知识与技能。希望通过对我成长过程的描述，能给同行一些参考，起到抛砖引玉的作用。我要感谢我的导师臧斌宇教授，我的课题组过去与现在的成员，麻省理工学院的M.弗兰斯·卡肖克（M.Frans Kaashoek）院士，明尼苏达大学的游本中（Pen-chung Yew）教授与加州大学圣巴巴拉分校（UCSB）的佛雷德（Fred Chong）教授，以及在我攀登过程中对我提供帮助的人。■

陈海波

CCF会员，2009年CCF优秀博士学位论文奖获得者。复旦大学并行处理研究所讲师。主要研究方向为系统软件与系统结构等。

oldseawave@gmail.com

“浪潮杯”CCF NOI 2011成功举办

CCF NOI 2011竞赛现场

2011年8月6~12日，“浪潮杯”第28届全国青少年信息学奥林匹克竞赛（简称CCF NOI 2011）在吉林大学隆重举行。此次大赛由中国计算机学会主办，吉林大学、吉林省计算机学会、东北师范大学附属中学共同承办，浪潮集团有限公司冠名赞助。

来自全国31个省市和港澳地区的340位选手及117位指导教师参加此次竞赛。340名参赛选手分别于8月8日、10日在吉林大学体育馆参加了两场竞赛。经过10个小时的激烈角逐，共产生CCF NOI金牌选手39名、银牌选手64名、铜牌选手118名；CCF NOI邀请赛金牌选手3名、银牌选手9名、铜牌选手19名。吉林省副省长王化文参加了开幕式。新华社、中央人民广播电台等11家新闻媒体对赛事进行了全程跟踪报道。

来自清华大学、北京大学、复旦大学、上海交通大学等11所“985”高校的招生代表参加了竞赛活动，并与优秀选手现场签署了录取协议。第29届全国青少年信息学奥林匹克竞赛将在江苏常州举行，承办单位是江苏常州高级中学。（立）