★CACM 中国版★

计算机协会通讯

CACM.ACM.ORG

2014 年 10 月第 57 卷第 10 期

利用空间同义词在地图上阅读新闻

大学的颠覆与革新

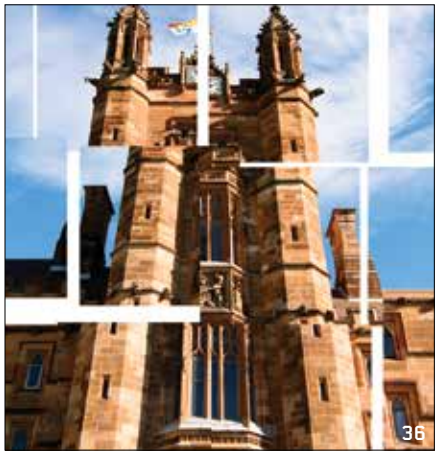
证书透明度

软件定义网络

解决双复合搜索问题

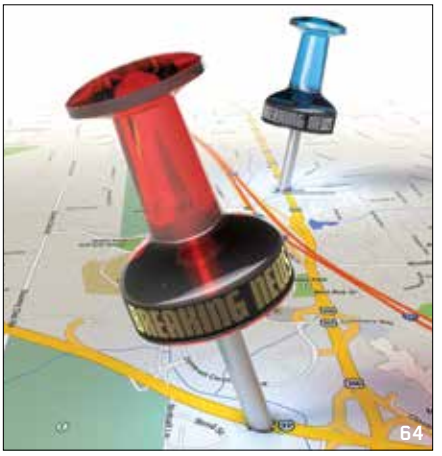
Association for Computing Machinery

观点

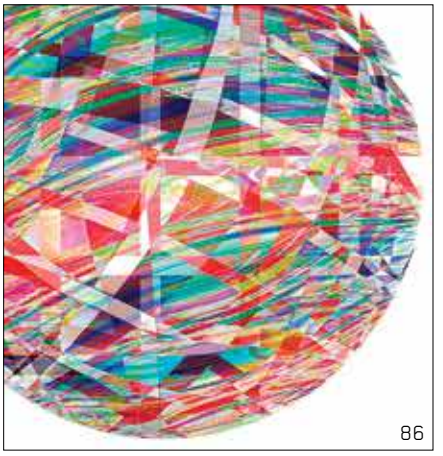


实践

投稿文章



评论文章



研究亮点

关于封面：

本月的封面故事（第 64 页）介绍了一种用地图查询界面来呈现网上新闻的系统。该系统名为 NewsStand（报亭），展现了从新闻报道中提取地理位置详情可如何为读者增加全新的资讯维度，以便他们可更好地鉴赏和理解内容。封面插图照片由 Coherent Images 提供。

中间图片由COHERENT IMAGES提供；右边图片由AVN PHOTO LAB提供

Association for Computing Machinery
Advancing Computing as a Science & Profession

DOI:10.1145/2661055

Henry Lucas

观点
大学的颠覆与革新

为应对由技术驱动的冲击，
高等教育机构必须变革其业务模型。

柯达（KODAK）、鲍德斯（BORDERS）和百视达（BLOCKBUSTER）公司最近接连破产，大学是否会步其后尘，被人们按相同的方式颠覆？与普通的大学相比，混合课程、在线学习和MOOC（大规模在线开放课程）正以光速前进。本期的观点着重分析了技术带来的机会和威胁，并建议高等教育机构积极应对这些挑战，变革其业务模型。本文列举了一些高等教育方面的变革，其中包括：

- 成立一所盈利性顶级研究性大学的密涅瓦计划（Minerva Project）。学生在世界各地不同的校园内共同生活，而顶尖的教授则把在线课程变成了一系列的学生研讨会（2013年4月21日《*纽约时报*》）。
- 新一代卢旺达(Generation Rwanda)正在启动一所完全由MOOC（大规模在线开放课程）组成的大学，收取的学费低于每年$1,500美元。根据该计划，将在卢旺达成立一所400人的大学，由MOOC提供内容，教员负责讨论。新罕布什尔南方大学（South-

ern New Hampshire University）会对学生进行考试，并颁发大专证书。现在，卢旺达的人口中有 1% 拥有大专以上学历（2013 年 3 月 15 日《*技术评论（Technology Review）*》）。

▸佐治亚理工学院（Georgia Tech）宣布，其将与 Udacity 和美国电话电报公司（AT&T）携手通过 MOOC 颁发专业的在线计算机科学理科硕士学位。该专业三年课程的学费预计为 $6,600 美元，并设有 4,500 个考试中心，用于有人监考的考试。据最近的报道，该学院所收到的该专业的申请人数已经达到了预计人数的两倍（2013 年 10 月 29 日《*华尔街日报*》）。

虽然这些方案可能会成功，也可能面临失败，但技术正在让高等教育中的很多新理念成为现实。本期观点阐述了各种创新的前景（和威胁），比如本人研究了颠覆性技术并亲自教学后所发现的各种前景（和威胁）。[2,5]

拓扑分布

表 1 总结了通过技术实现的多种新教学实践。在线教育包括全部采用在线课程、混合式学习和 MOOC；这些手段可以互相融合，互相配合，从而为学生创造大量的学习经历。[a]

机遇与威胁

表 2 描绘了用于学习的颠覆性技术所带来的机遇与威胁。该表区分了创造和使用 MOOC 和其他在线材料的内容生产者 / 消费者，以及使用他人创造的内容的内容消费者。

MOOC 及其他类型的在线教育的发展拓宽了大学的范围，使之远远超出了校内学生和校友的圈子；学校可以覆盖全球。在我的第一门 MOOC 课程中，学生来自将近 100 个国家。随着范围的扩大，竞争也日益加剧；MOOC 和在线课程把学校的品牌推向了全世界。不过，这意味着大学将会彼此竞争，吸引学生选修各种类型的专业和课程。而且，在单独的课程和单独的课程模块层面上，我们也会看到竞争。

通过技术增强的讲授把更多的学习责任放在了学生身上，相对减轻了教员的责任。

新一代卢旺达和密涅瓦计划展示了大学可能面临的最大威胁：消费内容的新企业，但其成本结构却只相当于现有大学成本的一小部分。内容生产商开发和分发内容；它必须顾虑开发课程所需的高昂投资。而以购买内容为主的大学则受益于这些工作。然而，必须达到一种平衡，让内容生产商和消费者在财务上保持活力。当类似 Coursera 的组织向学校授予 MOOC 许可，收取版税，并与创造 MOOC 的学校分享版税后，将会出现了一种新的融资模型。[4] 例如，圣荷西州立大学正在使用 MOOC 进行补救教学（2013 年 4 月 29 日《*纽约时报*》）。一些大学正把 MOOC 作为一种免费的，有学分的基础课程，用以鼓励学生就读大学，获取大学学位（见 http://mooc2degree.com）。

实际的校园也会发生变化。一些学生会选择在校园里呆上不到一个学年。他们在学期的某个阶段中会一边在家学习在线课程，一边做兼职。以后，大学需要的宿舍会更像酒店，而不是公寓。大多数教员在讲授混合式课程时，会减少实际的课时数，这减少了建设新教室的需要。同时，大学需要各种新的空间用于准备课程，讲授课程，开展学生互动。

类似 MOOC U 的项目很有可能会成为传统盈利性大学的终点。[3] 由 MOOC 和 / 或国家顶级教员的在线课程颁发的证书以及最终的学位很快就会优于现有盈利性大学的学位，而且花费肯定会更少。

表1讲授以及通过技术增强的教学的类型

讲授的类型	通过技术增强的教学的示例
盈利性大学的讲授	通常提供由非博士学位的教员讲授的非同步在线课程，基本无或无直接师生互动。
为了触及无法到达实际校园的学生。例如，在线的MBA课程。	1.同步的在线课程，拥有博士学位的教员和学生进行在线互动，外加非同步的家庭作业；可能还会与短期的住校课程结合。2.类似佐治亚理工学院的项目以MOOC和有人监考的实际考试为特征。
为了提高师生实际共处的课程的质量第一部分：	混合课程配有较传统课程时间更短的实际课堂讨论，并利用了视频授课和多媒体家庭作业；通常需要拥有博士学位的教员。
为了提高师生实际共处的课程的质量第二部分：	在由拥有博士学位的，具有创造力，能力强的教员讲授的传统课堂中纳入MOOC。
拥有对服务不足的人口和国家进行讲授	带有非即时视频和互动课时的免费MOOC，通过谷歌环聊（Hangouts）提供给一小部分参与者和教员。
大学的新模型：对传统学院的威胁。	1.密涅瓦计划，配有顶级教员提供的流媒体视频和本地讨论组。2.MOOC学位计划3.MOOC大学

a 混合式通常指课程没有讲授部分，课堂时间用于讨论或解决问题。该方式与“翻转教室”相关联。“翻转教室”是可汗学院成功挑战传统教学方法的项目。

对于学生而言，技术带来了难以置信的灵活性以及更多的选择，但代价是：通过技术增强的讲授把更多的学习责任放在了学生身上，相对减轻了教员的责任。马里兰州的史密斯商学院开设的在线 MBA 课程几乎能让学生在任何地点攻读学位。以后，将可以设置定制化的学位课程，这样学生便不再局限于完全只由教员确定的学习课程。

我们能否确信，通过技术增强的学习比传统学习更好？ Coursera 的奠基人之一 Daphne Koller 认为，技术能提高教育的质量，但不能取代教师。“在线的经历允许学生开展自我管理，在课程进入下一个主题前熟练掌握当前主题，并在掌握知识前真正进行实践（2013 年 5 月 15 日《*华尔街日报*》）”。从这一点来说，在大学层面，对于同步和非同步学习或 MOOC，很少见到严密的、可信的研究。[b] 圣何塞州立大学发现，在混合式电路课程中，举行讨论会并在课外时间使用麻省理工学院（MIT）的在线电路课程后，91% 的学生通过了该门课程，而传统课程中只有 59% 的学生通过（2013 年 4 月 30 日《*纽约时报*》）。我的经验说明，与讲座相反，注重讨论的课时虽然更短，但能创造更有活力的讨论。但是，上述两种观察结果都不是从受控的实验中得出的，所以问题尚未解决。

使用 MOOC 和其他材料后，教员可以为现有的课程增加价值。一位经济学教授可以在 MOOC 部分把两位获得诺贝尔奖的经济学家带入课堂。最大的担忧是，顶尖大学的少数明星教员讲授视频课程，而其他教员则沦为讨论的主持人或教学助理。在创建课程时，教员需要确定教学内容，筛选可获取的在线材料和阅读材料，设计课堂时间，准备作业以及把相关研究引入讨论。教员的最佳策略是创造性地使用新老方法增加其课程的价值，同时学习如何成为学生的导师和辅导老师。在最近的访谈中，Koller 谈到：“我们希望并相信教师的作用会发生变化。教师会有更多的时间用于教学，而不是把时间放在内容开发和准备以及反复批阅无穷无尽的相同作业上。学生来到课堂时，会真正享受与其他学生和讲师之间意味深长、引人入胜的讨论。”（2013 年 5 月 15 日《*华尔街日报*》）。

表2技术使能的学习所带来的机遇与威胁

实体	机遇
大学	
内容消费者	在全球范围拓展品牌，通过内容获得收入，为提高教育的全球水平做出共享，重新布置校园。
内容生产者/消费者	
成熟者	用顶级教员准备的新材料填补课程，重新布置校园
起步者	成立无昂贵教职员或物理工厂的新学校
学生	更高的灵活性；有机会减少在校园花费的时间，可获得各种类型的科目，学习效果更好，提升全世界的教育机会
教员	有机会使用新内容为课程增加价值
实体	**威胁**
大学	
内容消费者	投资成本更高，超支，面临开发内容的同行以及成熟的大学以及成本结构更低的新增大学的竞争；竞争的层面包括单独的课程和单独的课程模块。
内容生产者/消费者	
成熟者	受到来自起步者以及教育模型各异但成本结构更低的学校的竞争。
起步者	树立品牌和质优的声誉
学生	承担更多的学习责任，同时也对学习效果是否更好以及非传统学位和证书是否能得到雇主和他人认可承担责任。
教员	成为明星内容生产者的教学助理，对拥有博士学位的教员的需求减少。

建议的策略

美国和其他国家的大学可以通过使用技术变革讲授和学习过程。但是，这种技术本质上是对现有高等教育模型的颠覆。[6] 如果大学不承认颠覆性的技术会影响它们，那它们最终会沦落到柯达、百视达、鲍德斯或报业的境地，这不啻为一出悲剧。大学应该抱有大胆的思想，采取果断的行动来应对这些挑战，而不是进行渐进式的变革。为此，本文提出了下列建议：

- 任命主管混合式课程、在线课程和 MOOC 的副教务长和副院长。把成百上千门课程转换成混合式课程需要强有力的领导人和拥护者。
- 积极进军不同课程的发展中市场。通过尽早进入市场，树立学校品牌，有望取得较大的先发优势。
- 通过激励教员转变课程，加快混合式和在线课程的开发工作。此项工作范围很大，需要大量的资源和资金。
- 提供在线的学位计划，着重于拥有师生互动的高质量和高水平课程。

b 美国教育部网站上列出了几个在线教育的元研究（meta studies）。虽然研究的数量庞大，但是其中很多研究控制较弱，不够严密。在 Ithaka项目中，研究人员对卡耐基梅隆大学（CMU）开发的互动统计学课程的使用情况进行了随机研究，他们发现，在传统课程的学生和CMU课程的学生之间，学习效果不存在重要的差别。[1]

- 迅速把兼读制课程转变为混合式课程，减少学生在课堂中花费的时间。
- 利用大学的品牌创建和营销MOOC，并利用课程的版税取得商业顺差。
- 基于MOOC开发各种课程和学位课程，向支付学费并通过有监考的考试的学生授予学分。
- 培养支撑员工并建设相关基础设置，以协助教员进行转变。
- 雇佣能够把研究带入课程并持续更新课程的明星教员。新的明星将是那些通过互动的课时和论坛与学生沟通并挑战学生的人。
- 根据教学和研究潜力雇佣员工。
- 评审晋升和终身教职政策，增大教学的比重。同时，观察教员是否能把研究带入课程。
- 允许学生创建定制化的学位课程，支持他们把本校的课程与其他学校的在线课程和MOOC进行混合和匹配。
- 为学生提供灵活性，允许他们在本科各学年中在校学习和在家学习。
- 规划相应的物理设施以满足通过技术增强的学习的要求。大型的讲座已经死亡；混合式课程的规模在30-50名学生之间；学生在课堂上花费的时间更少，释放了教室容量。
- 修建可短期居住（几个月到一个学期）的宿舍。
- 考虑与提供MOOC的组织或为开发在线课程提供支持的公司达成伙伴关系。
- 找到替代学分制的机制；学生的学位要求以及教员的工作量现在均与学分制紧密相连。不过，在混合式课程中，什么是学分呢？

结论

对于大学而言，技术带来变革，也引发颠覆。密涅瓦计划和新一代卢旺达举出了个人如何创建一所通过技术实现的大学的范例。佐治亚理工学院的计算机科学硕士学位说明了如何利用MOOC为攻读学位的学生大幅降低学费。谁受到技术使能的学习的威胁？每所大学都有所担心，因为尚不清楚不同类型大学会受到何种冲击。总的来说，这个时代孕育了提高教育质量的大好机会，它让每个学生需要为自己的学习承担更多的责任，同时大大拓展了每所教育机构的学生范围。技术可让我们利用校园中的人力资本和知识，并用它来提高全世界的教育水平。大学会把他们的资源和精力贡献出来，应对这一挑战吗？ C

每所大学都有所担心，因为尚不清楚不同类型大学会受到何种冲击。

参考资料

1. Bowen, W., Chingos, M., Lack, K., and Nygren, T. Interactive learning online at public universities:Evidence from randomized trials.*Ithaka S+R*, 2012.
2. Christensen, C., Horn, M., and Johnson, C. *Disrupting Class*.McGraw-Hill, New York, 2011.
3. Cusumano, M. Are the costs of 'free' too high in online education?*Commun.ACM 56*, 4 (Apr. 2013), 26–29.
4. Dellarocas, C. and Van Alstyne, M. Economic and business dimensions money models for MOOCs. *Commun.ACM 56*, 8 (Aug. 2013), 15–28.
5. Lucas, H.C., Jr.*The Search for Survival:Lessons from Disruptive Technologies*. Praeger, Santa Barbara, CA, 2012.
6. Vardi, M. Will MOOCs destroy academia?*Commun. ACM 55*, 11 (Nov. 2012), 5.

Henry Lucas（hlucas@rhsmith.umd.edu）是马里兰州帕克分校史密斯商学院的信息系统教授。

在此感谢纽约大学斯特恩商学院前院长和巴布森学院前校长William Dill先生，他对本期观点的初稿提出了自己的真知灼见。

译文责任编辑：谢涛

活动日历

10月19日-21日
关于系统、程序设计和应用的会议：改善人类的软件（Software for Humanity）
奥勒冈州波特兰
承办单位：SIGPLAN
联系人：Andrew P. Black
电子邮箱：black@cs.pdx.edu

10月19日-22日
游戏中的人机互动年会
加拿大多伦多
承办单位：SIGCHI
联系人：Lennart Nacke
电子邮箱：acagamic@googlemail.com

10月19日-22日
第32界IEEE国际计算机设计会议
韩国首尔
联系人：Naehyuck Chang
电子邮箱：naehyuck@elpl.snu.ac.kr

10月20日-21日
网络与通信系统架构研讨会
加利福尼亚州洛杉矶
承办单位：SIGARCH、SIGCOMM
联系人：Viktor Prasanna
电子邮箱：prasanna@usc.edu

10月20日-22日
第16界国际计算机和可访问性ACM SIGACCESS会议
纽约州罗切斯特
承办单位：SIGACCESS
联系人：Sri Kurnianwan
电子邮箱：srikur@soe.ucsc.edu

10月20日-24日
关于系统、程序设计和应用的会议：改善人类的软件（Software for Humanity）
奥勒冈州波特兰
承办单位：SIGPLAN
联系人：Andrew P. Black
电子邮箱：black@cs.pdx.edu

10月27日-28日
第13界ACM网络热点议题研讨会
加利福尼亚州洛杉矶
承办单位：SIGCOMM
联系人：John Heidemann
电子邮箱：johnh@isi.edu

DOI:10.1145/2659897

文章编写主导者 acmqueue
queue.acm.org

公开的、可验证的、只能追加的日志

作者：BEN LAURIE

证书透明度

2011 年 8 月 28 日，攻击者利用为 google.com 误发的 HTTPS 通配符证书向伊朗的多位用户发动了中间人攻击。该证书由一家名为 DigiNotar 的荷兰认证机构（CA）颁发，该机构为 VASCO 数据安全公司（VASCO Data Security International）的子公司。随后的分析表明，DigiNotar 至少在 7 月 19 日，即一个多月前就已经知晓了系统的漏洞。同时，分析还表明，该机构至少颁发了 531 张虚假的证书。最终的确切数字可能永远无法知晓，因为 DigiNotar 未对所有误发的证书进行记录。2011 年 9 月 20 日，DigiNotar 宣告破产。

这一漏洞造成的破坏并不仅限于伊朗。在最开始发现误发的两周后，DigiNotar 的根证书最终被吊销了，但其中包括一张荷兰政府用于提供互联网服务的证书。这次吊销使得荷兰人无法购买和销售汽车、进行电子清关、在国际市场上购买电力以及开展很多其他的活动。当然，从 DigiNotar 获取证书的每台网络服务器还必须竞相申请新证书。

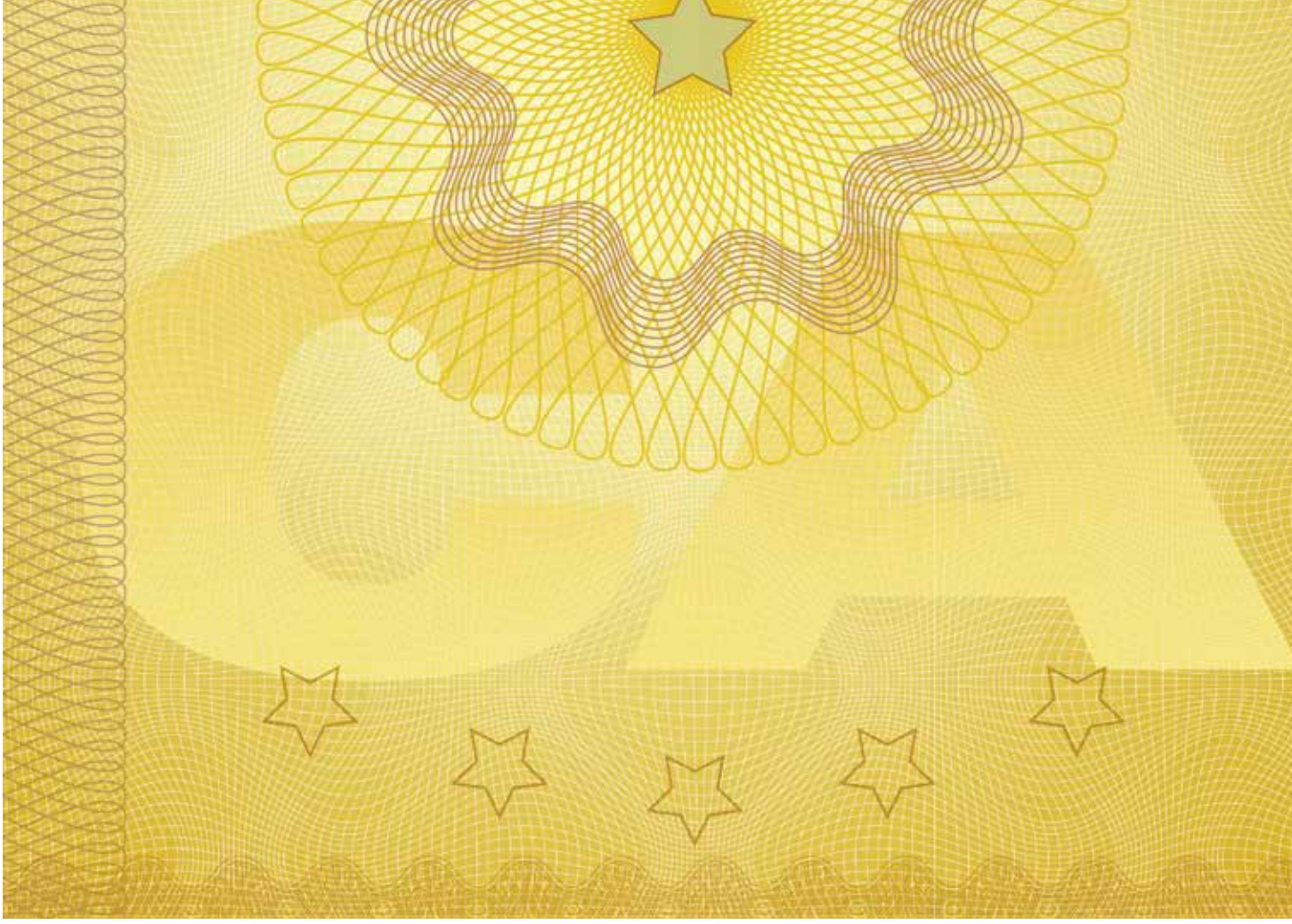
这不是第一家被攻破的认证机构（CA），也不会是最后一家。科摩多集团（颁发了谷歌（Google）、雅虎（Yahoo!）、Skype 和其他公司的虚假证书）、[1] TürkTrust（未经授权的 google.com 证书）[2] 以及 ANSSI（通过中间人颁发的证书，据称用于局域网监控）均报告过漏洞或内部的误发。此类事件肯定会越来越多。

因这些漏洞责怪认证机构的安全性差很容易，但事实是软件工程和系统设计的最新进展尚不足以让任何人在攻击下保持绝对安全。虽然认证机构肯定脱不了干系（特别是有些认证机构明知事情已经发生，却捂着不公布，希望它们能够自己摆脱！），但无人知道如何构建一个完全万无一失的系统。那么，我们怎么才能做出改进呢？

认证机构的替代方案

或许我们可以退后一步，考虑我们真正想解决的问题。这可能更有指导意义：我们最终希望确保网络用户正在交谈的对象实际上与他们认为的对象相同，且其他人无法截获对话。这真是一个无法完成的任务——计算机怎么能知道用户在想什么——但是，让我们暂时把这个问题简化为一个稍微不同的问题：如何确保网络用户的谈话对象与正使用的 URL 所对应的域名所有人一致。这个策略的确相当弱，但是它至少在某些场景下奏效。很多人知道大型网站（如 Google、Pay-Pal 和 eBay）的正确 URL。同样，如果用户从诚实的源头开始，一直打开正确的链接（实际上，大多数链接都是这样的），他们便能得到保护。

计算机领域依赖多年的解决方案是在系统中引入可信第三方（认证机构），由他们担保域名和私钥之间的绑定关系。问题是，我们已经支持了上百家理应可信的受信任方，其中任何一家都能够为任何域



名提供担保。其中不时有一家会犯错，有时候影响会很大。

有哪些替代方案呢？它们又会面临怎样的约束呢？首先，我们来谈下约束。

约束约束必定会因系统的本质不同而不同。在此，本文的重点是从谷歌浏览器（Google Chrome）视角出发，对万维网进行探讨。在为 Chrome 设计安全系统时，需要考虑下列约束：

- *迁移路径*必须有某种可行的方式从此处到达彼处。例如，需要在某个特定日期改变整个世界的方案肯定不会成功。
- *通用*没人能搞特殊。每个人都必须能够参与。换言之，解决方案必须能够伸缩。
- *（几乎）不增加延迟。* Chrome 非常重视页面加载时间，所以不能明显让浏览器变慢。由此可得出不应存在同步的带外通信：页面加载前所需的一切数据必须通过与页面本身相同的信道送达。经验表明，如果我们必须引用另一个源，则加载会失败，且会相当慢。（有关该经验的讨论，参见 https://www.imperialviolet.org/2014/04/19/revchecking.html。）
- *别把情况弄得更糟。*例如，通过引入另一家第三方来“解决”一群受信第三方的问题似乎没有进步。
- *不要把决策丢给终端用户。*我们已经知道，用户无法理解证书警告。期望他们能够理解更为复杂的方案不大可能成为我们的答案。

考虑了认证机构和受信第三方系统在确保互联网安全方面的局限后，人们发现了几种替代方案：在谷歌，我们只找到了一种（破坏者预警）——证书透明度——它可以克服所有的约束，为安全问题提供一个合理的解决方案。在探讨这种情况的理由之前，我们来看看其他的一些替代方案。

绑定。替代方案之一是，网站向外通告正确的证书（或认证机构）列表，如果证书不在该证书列表上，则浏览器拒绝此证书。现在，某些站点的绑定关系内置在 Chrome 里，但是也有方案要求允许任何人宣告绑定关系（例如，https://datatracker.ietf.org/doc/draft-ietf-websec-key-pinning/ 和 http://datatracker.ietf.org/doc/draftperrin-tls-tack/）。

出于某些微妙的原因，绑定失败了。如果出问题了怎么办？很明显，不能简单地用新的绑定关系替换旧绑定关系，否则整个绑定系统都会失败。所以，经过某段预定的时间后，绑定关系会失效。如果你在绑定关系失效前丢失了秘钥，那么你的站点在绑定关系失效前将无法访问。如果有效期短，则很难防护入侵者的攻击。但是，如果绑定关系的有效期长，则意味着出现灾难时故障时间会相当长。

现在，如果出现上述事件，采取的方法是联系 Chrome，要求变更您的绑定关系，但是这不是一个可伸缩的、包容性强的解决方案。不仅如此，只要有一些绑定关系内

所有的证书图片由ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES提供

置在 Chrome 时，也就相当于引入了新的受信第三方（比如谷歌）。

公证机构。另一种流行的替代方案是使用公证机构。最有名的公证机构为 Perspectives 项目（http://perspectives-project.org/）和 Convergence（http://convergence.io/）。谷歌也曾短期运营过一家名为 SSL Certificate Catalog[4] 的公证机构，但在证书透明度的工作启动后，谷歌决定不再支持该项目。该理念采用定期扫描互联网的做法（如果从多个视角则最为理想），把所有的证书插入公证日志，以便回答日后的下列询问："您之前是否见过这张证书？"如果答案是"没见过"或"不是经常看到"，则可能需要带着某种怀疑的眼光观察该证书。

不过，这一理念存在不少问题。最大的问题是，"没有见过"的回答可能仅仅说明站点刚换过证书。在站点每次续约证书时，是否都应无法访问？这听起来相当怪。其次，使用公证机构的方法需要带外检查，破坏了一条可部署性规则。第三，胆大的攻击者可以广泛部署虚假证书，让公证机构认为证书是好的。最后，它引入了新的受信第三方。

DNSSEC.另一种替代方案则基于对域名系统安全扩展（DNSSEC）的信任。有两种机制可实现这一目标：基于 DNS 的命名实体验证（DANE; https://tools.ietf.org/html/rfc6698）以及认证机构的授权（CAA；https://tools.ietf.org/html/rfc6844）。严格来说，使用 CAA 后，DNSSEC 不是必须的，但仍被强烈建议使用。通过为主机名保存合适的 DNS 记录，DANE 和 CAA 两者均用明显的方式把特定的证书或认证机构与主机关联起来。它们的区别在于记录的使用方式：DANE 记录供客户端在连接服务器时检查；CAA 记录供认证机构在颁发证书时检查。如果新证书的认证机构在 CAA 记录中不存在，那么它应该拒绝颁发。

在两种方案中，均存在 DNSSEC 固有的问题，但是 CAA 还存在更深层次的问题，即没有真正解决下列问题——被侵入的认证机构或怀有恶意的认证机构误发证书。很明显，它们不会费事去查询 CAA 记录。

考虑了认证机构和受信第三方系统在确保互联网安全方面的局限后，人们发现了几种替代方案：在谷歌，我们找到了一种可行方案（破坏者预警）——证书透明度。

然而，更重要的是，与现有的公钥基础设施（PKI）类似，DNSSEC 在场景中引入了多个受信的第三方，它们或许能，或许不能完成自己的职责。在这种情况下，受信的第三方为 DNS 注册表和注册机构。不幸的是，它们的安全记录比认证机构的要差得多。

一些人认为，DNSSEC 本身可以防范这些受信第三方的破坏，因为 DNS 的公共性质让任何干预立刻显形。该理论存在的问题是，DNS 是一个分布式系统——我的视图不是你的视图，而且也无法保证我们的两个视图完全一致。因此，攻击者可以相对容易地呈现一个分裂的世界，向受害者展示一组 DNS 记录，而向试图检查 DNS 完整性的人展示另一组记录。

另一个问题仅仅是因为 DNSSEC 尚未广泛部署，所以这会让我们把一种改进放在我们已经等了 10 多年的另一种改进上（事实上，8 年以前，我曾经一度忙于解决 DNSSEC 存在的"最后"一个问题；见 RFC 5155 http://tools.ietf.org/html/rfc5155）。

最后，实验说明，或是由于路由器接管了 DNS 且不支持 DNSSEC，或是由于强制网络门户以及类似机制，或是由于在端口 53 上阻塞了 TCP（DNS 通常通过 UDP 传输，但是较长的记录则需要使用 TCP），或是由于其他原因，至少 4% 的客户端完全无法获取 DNSSEC 记录。

不过，请注意，在 SMTP 的场景中，DANE 能发挥作用。SMTP 服务器已经完全受 DNS 的控制，且现在只使用机会型传输层安全（TLS）。DANE 肯定是一种进步。

基于比特币的解决方案。我曾经写了多篇文章论述比特币的缺陷（例如，它是史上最不环保的发明。如果它真的是分散的（实际它做不到），那么足够强大的对手便可以摧毁全部的历史。）然而，人们仍

然毫无理智地继续膜拜比特币的神坛，而且这种崇拜扩展到了 DNS 和秘钥领域——例如，DNSChain (https://github.com/okTurtles/dnschain)。

除了价格极其昂贵外（按浪费的能源和永久性来衡量），这种解决方案还引入了新的受信第三方（在区块链中建立“共识”的那些机构），且缺乏验证机制。

证书透明度

鉴于上述所有替代方案均存在缺点，我们转而追求一种名为证书透明度的方法。证书透明度背后的核心理念是公开、可验证且只能追加的日志。创建记录所有已颁发证书的日志后（由于这些证书可从密码学上进行验证，所以并不需要这些证书都是可信的（而且事实证明这是可行的，后面我们会详细解释）），便可支持客户端检查证书是否在日志中存在，而服务器则可以监测日志，防范误发的证书。如果客户端拒绝连接未在日志中记录的站点，则上述方法便可成为一个完整的解决方案。而且，误发证书不可能不被发现。

这种机制可让我们满足之前列出的所有约束。存在迁移路径：证书可以继续被颁发和吊销，就如同之前一直的做法一样。而且，随着时间的推移，越来越多的客户端会检查日志的是否包含证书，而越来越多的服务器则会监测日志。在所有客户端检查前，只要某些客户端检查了日志，就能为未检查的客户端提供一种群体免疫机制。

人人都能参与。把证书添加进日志不难。而且，由于日志本身并不判断证书的正确性，吊销虚假证书的过程并无变化，仍由认证机构完成。

没有增加延迟，因为日志入选的证明相当简凑，且包含在 TLS 握手中。

也没有引入受信第三方。虽然日志确实是第三方，但它不是受信第三方；任何人均可以检查其操作是否正确，而且如果它的操作有误，也能予以证明。

最后，证书透明度未把决策推向用户。证书或是已记入日志，或是未记入日志。如果已记入日志，则对应的服务器操作员（或其他相关方）能够看到证书；如果证书无效，则它们会采取妥善的行动。如果证书未记入日志，则浏览器可以简单地拒绝建立连接。

工作原理如何才能构建一个可验证且只能追加的日志？虽然存在一些设计上的权衡，也需要解决一些有趣的问题，但从本质上来说，这相对简单。一个明显的方法是，日志的客户端可以简单地通过定期下载整个日志来验证其是否满足只能追加的属性。客户端还可以比较自己的日志副本与其他多个客户端的副本，以验证它们观察的所有日志是否相同。然后，人人都会知道，该日志确实是公开的，且只能追加。

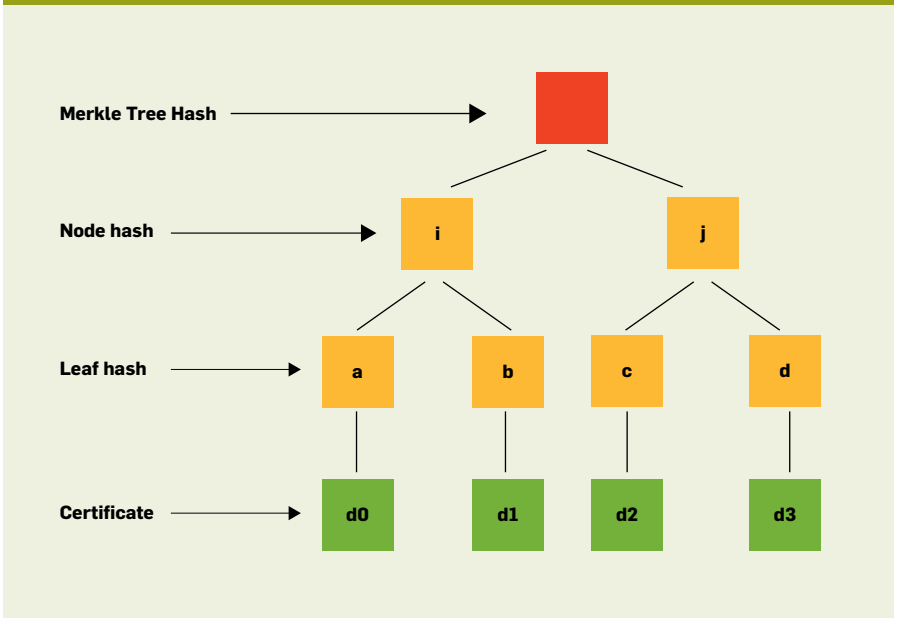
不过，这非常浪费带宽和存储空间。一个更好的方法是使用附图中的 Merkle 树。Merkle 树的叶子代表日志中的条目。树的每条分支为分支以下节点的密码散列（这符合一种略显奇怪的习惯，即树向下生长；根在顶部，叶子在底部）。很明显，基于密码散列的性质，每个节点为其下方所有节点的汇总：在不改变散列值的情况下，无法改变任何东西。据此，树的根为所有叶子节点的汇总。这意味着，现在客户端只要简单地比较散列值便可高效地验证它们看到的树是否相同。

然而，我们希望日志的作用更大。我们希望验证声称在日志中存在的事物确实在日志中存在，且日志具有只能追加的属性（或者，换句话说，今天的日志包含了昨天的全部日志）。幸运的是，Merkle 树可以高效地实现这一功能。为了证明日志中包含特定的叶子，我们只需要该叶子的散列值以及组成树的相邻散列值的列表。根可根据该列表重新计算得出。而且，由于散列值是加密的，这也意味着叶子必须在树中存在（否则，在结合生成的散列值列表与叶子的散列值后，将无法生成根的散列值）。与此类似，由于添加进树中的新条目也会有散列值，可通过它们的散列值列表把昨天的树根链接到今天的树根上。

完整的系统还差一环。如果日志出现问题，则我们需要证明它存在问题。幸运地是，这次还是相当容易。日志只需要在事物上签名。原则上，日志实际只需要在一件事物上签名：树的根节点。这同时也是对树中所有事物的签名。

这引发了第一个设计权衡点。日志应该具有高可用性和一致性。为了让 TLS 客户端确保证书在日志中确实存在，每张证书必须包括某种入选证明。（从技术角度来说，该证明可在与服务器对话的任何位

Merkle 散列树。



置出现。不过，由于发布新的服务器软件需要数年时间，现在能够普遍使用的唯一选择是把该证明直接放在证书中。）

我们最初的设想是包含 Merkle 证明——也就是说，已签名的树根以及证书条目至已签名树根之间的散列值。然而，这会直接面对可用性 / 一致性的权衡：为了保证日志的高可用性，你需要在不同的地理位置上运行多个实例。这意味着，实例间同步过程可能需要一段时间，而且为了实现只能追加的属性，它们之间必须同步。因此，在认证机构颁发证书前，它必须向日志提交证书，并等待日志完成所有实例的同步（以便在日志中为新证书分配位置）及返回日志新版本的 Merkle 证明。虽然大多数时候很快，但如果出现网络瘫痪和服务器宕机，它的时间会相当长——几小时，甚至数天。

认证机构发现，在颁发证书的管道中出现这种延迟是不可接受的。事实上，在权衡的范围内有些场景甚至更慢——例如，日志会同时返回入选证明和另一种证明，即日志监测器不但已经发现了新证书，而且如果误发了证书，还有时间采取行动。

幸运地是，权衡时也有一些更快的因素。在最终实现的版本中，有一台日志服务器，它会返回新证书的已签名时间戳，称之为已签名证书时间戳（SCT）。这种已签名时间戳承诺会在未来把证书加入日志。每个日志带有最大归并延迟（MMD）参数，其说明了在颁发已签名时间戳后，允许它在加入日志前等待多久。

表面来看，这似乎允许日志立即响应，因为日志只需要生成签名，而在现代的硬件中，签名的过程不用一毫秒。但是，这不完全正确。既然 SCT 是一种加入证书的承诺，那么在颁发 SCT 后，日志便绝对无法承受证书的丢失。因此，在存储收到的证书时，日志必须具备一定的冗余；证书或许应该储存在多个数据中心内。这确实可以降低对一致性的要求。日志可以把接收的证书存储在其冗余实例的某个子集中，尔后在确定新证书的顺序以及生成日志的新版本前，解决任何不一致之处。这种类型的冗余较一致的冗余要快很多。

第二个权衡点是 MMD。很明显，监测日志的机制希望 MMD 尽可能短，因为倘若日志参与了故意误发证书，那么很可能会尽可能地推迟把证书记入日志。注意，如果日志没有故意串通或丢失对其秘钥的控制，那么认证机构（或认证机构的攻击者）无法影响日志集成新证书所需的时间。另一方面，日志的运营方则需要足够长的 MMD，以便让日志达到一致的状态，甚至在出现软件错误时或许也能达到这一目标。我们尚未就可接受的 MMD 做出决定，但是很明显，它至少有数小时，也可能长达数天。

第三个权衡点是每张证书应该记入多少处日志。出于以下两点原因，使用更多的日志更好：首先，如果日志真的出现问题，客户端将不再信任它。如果证书的所有SCT都源于有问题的日志，那么证书将不再有效。其次，证书需要的 SCT 越多，则攻击者逃脱检测的难度就更大，因为攻击者将不得不同时控制认证机构（或认证机构的秘钥）以及所有所需的日志（或它们的秘钥）。

我们现在的想法是，每张证书应该至少在两处日志中记录，如果证书的有效期增加，则日志的数量随之增大——对于有效期超过 39 个月的证书，应在至少 5 处日志中记录。减少日志数量的原因并不复杂：TLS 握手的规模增大；创建证书所需的时间变长（虽然我们已经注意到，当日志的数量超过所需的 SCT 数量时，并行地向所有日志发出请求，并采用最先响应的 n 个日志通常相当快）；冗余的日志记录引起单个日志的规模和带宽要求增大。

最后，也就是第四个权衡点：日志中应该记入什么样的内容？一种诱人的回答是，“任何事物”，但是，只有当日志的规模可以管理时，日志才有用。有人需要检测日志。而且，倘若日志太大，检测它们不可行，那么日志还不如不存在。一种简单的解决方案是，只加入能够与客户端认可的认证机构相关联的证书。可以认为，把不符合上述条件的证书记入日志没有任何意义，因为浏览器不会接受它们。（倘若此类日志可从某种程度上提升自签名证书的作用，那也很不错；但由于存在滥发问题以及缺乏有效的方式吊销自签名证书，尚无人发现利用此类日志的方法；不过，在临近篇尾时，我们讨论了其他类型的日志）。另外，即使浏览器确实接受了此类证书，现在也没有可伸缩的方法来吊销它们。

生态系统。只有在受人监测时，日志才有用。所以，了解这个生态系统的参与者以及他们确切的行为相当重要。

只有在受人监测时，日志才有用。所以，了解这个生态系统的参与者以及他们确切的行为相当重要。

首先，证书必须以某种方式记入日志。虽然在最开始时认证机构可能会完成该项工作，但证书透明度的标准还允许在 TLS 扩展中包含 SCT。这一特性要求修改服务器软件。不过，Apache HTTPD 服务器已经实验性地支持该特性。给定支持 TLS 扩展的软件，网站运营方可以自己记录日志。而且，因为他们可以在需要的时候随时更新 SCT——在 SCT 固化成证书后，这无法实现——它们还可以减少使用的证书，降低证书可能被拒绝的风险。

其次，客户端必须检查它们所看到证书是否真的在日志内。因为要求中规定不能使用带外通信，所以这意味着在证书存在时相信日志，随后再要求提供入选证明来验证其真实性。

第三，相关方（例如，网站运营方、认证机构和研究人员）需要监测日志，确保认证机构无任何不当行为——例如，向站点颁发不属于它们的证书，或颁发扩展或标志集合不合规的证书，或颁发不符合标准的证书。监测还确保日志没有违反只能追加的属性，或违反它们的 MMD，或出现其他的不当行为。

最后，与日志互动的每个人应该互相检查，确保它们看到的所有日志均相同——也就是说，对不同的人而言，日志不应呈现不同的视图。否则，日志将能够说服客户端它在某个视图中记入了证书，而展现的视图中又未包含据称已经向网站颁发的那些证书。

流言。除了最后的任务之外，所有的任务都不复杂。监测日志，获得一致性和包含证明等等可以通过直接查询日志实现，但是检查一致的视图却更难。为了实现这一功能，各种日志客户端必须使用流言。长远来看，该功能可以通过多种协议实现——XMPP、SMTP 以及对等连接等——但是我们的第一个建议是通过 TLS 采用背驮方式传播流言。当客户端连接服务器时，它向服务器发送一些条目，而服务器可以验证或只是缓存这些数据；返回数据时，服务器发回从自己的缓存

中抽取的一些条目。这样便在客户端之间建立了高效的对等网络。

因为这种交流附在因其他目的而建立的连接（很可能是获取网页）之上，为了节省带宽，避免延迟过高，应该只发出一些条目。当专门为流言建立连接时（例如，直接与日志服务器建立连接，或与其他客户端使用某种对等协议），则不再需要担心上述问题；客户端和服务器可以选择发送大量的条目——甚至是所知道的所有事物。

它们需要发送什么样的条目呢？这是争论的（和仿真）的焦点，但是我们有理由相信，最低的需求是：STH（已签名的树根）。每个客户端应该能让自己确信，它和所有其他客户端看到的日志相同。由于日志通过 STH 汇总，所以很明显，客户端至少希望用流言传播它们。

STH 有些很好的性质便于流言传播。首先，它们已经被签名，所以恶意的参与者无法把垃圾数据注入协议。每个参与者可以轻易地拒绝并非源于日志的消息。其次，给定源于相同日志的两个 STH，可以证明它们之间的一致性，然后抛弃较旧的日志。这意味着，缓存的大小为 O（日志数量）。

更多的流言更好吗？通过流言传播最近的 STH 的 STH 一致性证明可能有用，进而可以降低日志的负载。服务器可能还希望通过流言传播它们自己的 SCT 入选证明以及其对应的 STH。

在书写本文时，流言传播的确切内容和时间仍然悬而未决。人们正在通过仿真的方式对其进行探索。

透明度的其他用途。证书透明度起初推动我们完成了可验证日志的工作。但是，它还有其他一些有益的应用：

▸ *二进制透明度。*它支持您用日志记录可从互联网上下载的应用。与证书透明度类似，二进制透明度并不能阻止恶意的二进制，但是它确实向用户保证，用户获取的二进制是公之于众的，可供任何人分析，这使得部署目标明确的恶意软件变得更难。

▸ *DNSSEC 透明度。*DNSSEC 是一个不错的方案，可用于替代基于认证机构的验证领域。但是，它的潜在不足之处也不少——特别是域名注册表和注册机构。DNSSEC 中保存的秘钥的透明度可以确保可监督秘钥是否用于正确的操作。

▸ *吊销的透明度。*一旦虚假的证书被确认后，便应该吊销该证书。在这一过程中，现有的机制同样无法摆脱欺诈——例如，为了实现恶意的目的，选择性地把吊销状态设为非吊销。

▸ *ID 到秘钥的映射。*例如，这可能包含用 PGP（完美隐私）发送的邮件，用 OTR（非公开的；见 https://otr. cypherpunks.ca/）发送即时消息的 ID。

▸ *受信的时间戳。*现在存在数字公证机构的协议，但是它们需要人们信任公证机构。记录所有时间戳的公证机构将不再需要被信任。

*其他的构件。*在思索吊销的透明度时，我和我的同事 Emilia Käsper 创造了一种新的构件：稀疏 Merkle 树（http://www.links.org/files/RevocationTransparency.pdf）。其背后的理念是，如果你想拥有一个可验证的映射关系，你可以把映射中的元素存储为规模巨大的 Merkle 树的叶子——比如，拥有 2^{256} 张叶子的树（即，深度为 256）。虽然在正常情况下无法计算这种树，但是我们观察到，大多数叶子为空，这意味着大多数的上一级拥有相同的值——两个空叶子的散列值；再上一级，上上一级等的情况也与此类似。这意味着，只要树是稀疏的，事实上可以计算树根并生成入选证明等。这种结构可用作可验证日志的附属，用于提供高效的、可验证的映射。

状态。谷歌正在积极的开发证书透明度。在生产环境中，我们有两个日志正在运行，计划年底运行第三个日志。其他机构（例如互联网协会（ISOC）、阿卡迈（Akamai）和各种认证机构）也正计划运营公共日志。对于所有的关键组件，我们都有开源的实现。Chrome 支持证书透明度，而且将把它作为对 2015 年 1 月起使用的扩展验证（EV）证书的强制要求。超过 94% 的认证机构（按颁发的证书数量衡量）已经同意在它们的 EV 证书中包含 SCT。

一旦我们观察到系统在 EV 证书上运行良好，我们便会计划在*所有*证书上推行证书透明度。我们还打算探索可验证日志和映射的一些其他用途。

queue.acm.org 上的相关文章

网络取证（Network Forensics）
Ben Laurie
http://queue.acm.org/detail.cfm?id=1016982

数据锁定的反例（The Case Against Data Lock-in）
Brian W Fitzpatrick and JJ Lueck
http://queue.acm.org/detail.cfm?id=1868432

十年中的OS访问控制可扩展性（A Decade of OS Access-control Extensibility）
Robert N.M.Watson
http://queue.acm.org/detail.cfm?id=2430732

参考资料

1. Comodo Group.Mar. 31, 2011 update; https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html.
2. Langley, A. Enhancing digital certificate security.Google Online Security Blog, 2013; http://googleonlinesecurity.blogspot.de/2013/01/enhancing-digital-certificate-security.html.
3. Langley, A. Further improving digital certificate security.Google Online Security Blog, 2013; http://googleonlinesecurity.blogspot.co.uk/2013/12/further-improving-digital-certificate.html.
4. Laurie, B. Improving SSL certificate security.Google Online Security Blog, 2011; http://googleonlinesecurity.blogspot.co.uk/2011/04/improving-ssl-certificate-security.html.

Ben Laurie 是软件工程师、协议设计师和密码学家。他是Apache软件基金会的创会理事、OpenSSL的核心组员、Shmoo Group的成员、Open Rights Group的理事、The Bunker Secure Hosting的安全理事、FreeBMD的创始人成员、剑桥大学计算机实验室的客座研究员以及FreeBSD的贡献者。Laurie 在伦敦为谷歌工作，现专注于证书透明度。

校对：宋方睿
译文责任编辑：陈文光

DOI:10.1145/2629572

即便你无法确定你要找的信息，你仍可使用这个地图查询界面来搜索世界。

HANAN SAMET，JAGAN SANKARANARAYANAN，MICHAEL D. LIEBERMAN，MARCO D. ADELFIO，BRENDAN C. FRUIN，JACK M. LOTKOWSKI, DANIELE PANOZZO，JON SPERLING， BENJAMIN E. TEITLER

利用空间同义词在地图上阅读新闻

你旅行吗？你想知道在此次旅行目的地及其附近正发生的事件吗？你想了解你离开的地方及其附近最近的新闻吗，特别是你曾居住或工作过的地方？如果你对上述任一问题给出了肯定的回答，那么我们的报亭（NewsStand）【指新闻的空间-文本聚合及其显示（Spatio-Textual Aggregation of News and Display）】应用和相关系统恰好符合你的需要。

报亭[46]是支持人们利用地图查询界面来检索信息的通用框架的一个示例应用。如上所述，在之前的30多年里，我们一直在马里兰大学开发我们称为“空间浏览器”的系统，而上述应用则是其中的一个变体（见Samet等人的两篇论文[39，41]）。地图查询界面的优点是，由于结合了查看时缩放尺度可变的能力，地图可为搜索过程提供内在的粒度，加快近似搜索。这种能力使之与广为流行的基于关键词的常规搜索方法区分开来。常规方法提供的近似搜索功能有限，其主要通过匹配关键词的子集实现。然而，用户通常无法确定应该使用哪个关键词，因此他们希望搜索时能考虑同义词。在名为“对空间数据的空间查询（spatial queries to spatial data）”的空间参照数据的查询方面，地图查询界面取得了一定的进展。我们考虑了指向位置的行为（例如通过定点设备进行妥善定位或采用适当的手势），并依据缩放比例解释该定位规格的精度。这等价于允许使用空间同义词。

支持使用空间同义词相当重要，因为它支持用户在不确定查询目标时或不确定查询应该返回的结果时搜索数据。例如，假设用户查询是“在曼哈顿举办的摇滚音乐会（rock concert in Manhattan）”。如果找不到在曼哈顿举办的此类事件，那么在哈莱姆、布鲁克林区或纽约市举办的摇滚音乐会都算是相当匹配的答案，因为它们对应于曼哈顿的空间同义词：哈莱姆在曼哈顿区之

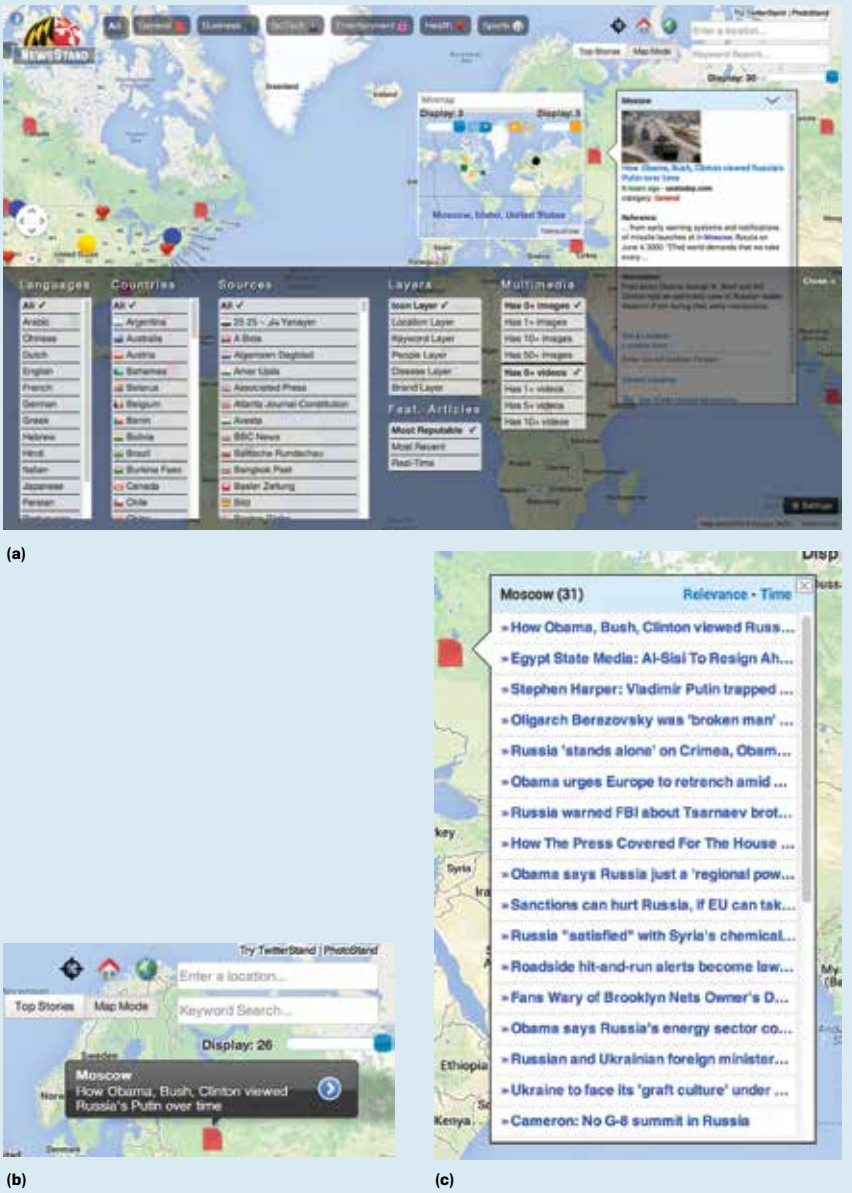
» 重要见解

- **报亭的地图查询界面监视着10,000多个发布时间间隔在数分钟之内的RSS新闻源的输出，并把文章与文中提到的地点关联。**
- **在结合查看和解释时缩放尺度可变的能力后，地图可为搜索过程提供内在的粒度，加快近似搜索，并允许使用空间同义词。**
- **对于移动设备的用户来说，地点的文本规范虽较几何规范更好，但却必须解决潜在的模糊性。**

插图由COHERENT IMAGES提供


BREAKING NEWS
Green Knoll Golf and Tennis
Best Buy
Commons Way
Somerset Corporate
Bond St
Birdsall Ln
Birdsall Ln
Country Club Rd
Golf Links Dr
Alletra Ave
206
202
22
28
287
Easton Turnpike

图 1.报亭的地图模式：（a）“2014年3月26日地点X处正在发生什么？”；（b）奥巴马/普京关系主题的代表性标题；以及（c）与莫斯科相关联的主题的代表性标题。



内；布鲁克林区则靠近该区且与该区的等级相同（两者均为纽约市下面的区）；而纽约市则包含了该区。常规的搜索引擎通过动态纳入从搜索 - 点击日志中收集的信息处理空间查询；据此，如果有足够多的用户在搜索曼哈顿时最终点击了与哈莱姆或纽约有关的网页，那么随着时间的推移，搜索引擎会推断出文档的空间范围最接近纽约，或与纽约相关。最近，搜索引擎（如谷歌的知识图和微软的 Satori）正在使用大型的知识库来理解关键词搜索查询的空间焦点，以及在一定程度上理解文档的空间焦点。尽管搜索引擎在理解文档的地点方面已经取得了上述进展，但是搜索引擎的主要工作原理仍然基于流行度。从这个角度来说，网页排名（PageRank）算法和点击日志确保向用户提供的网页会通过考虑了某种频率因素的测量值进行排序。具体而言，经典的网页排名算法使用了静态数据，但点击日志对应了动态数据。基于频率的方法确保其向某用户提供的结果与其他用户的结果相同。这种性质可以被刻画为“搜索的民主化”，就是说所有用户得到了相同的待遇。用更直白的方式来说，产生的效果是对用户不加区分，就是说他们得到相同的坏（或好）答案。换言之，使用网页排名算法和点击日志来对结果进行排序的效果（高效地选择向用户呈现的结果）是，如果没有人在之前曾搜索过某些数据（或空间意义上的近邻）或链接到该数据上，那么该数据将永远无法找到，因此也永远不会呈现给用户。在某些情况下，这种方法是可行的。然而，就同义词而言，这种方法对搜索结果的质量施加了相当强的负面影响，因为这意味着，对于不使用相同词语但内容等同的页面，倘若没有人链接到该页面，或点击空间邻区，那么搜索引擎将永远无法找到相似性。同样地，网页排名算法也永远不能找到类似的页面；在构建网页索引时它会在网络上抓取信息，但它却找不到有用的点击日志。

我们在马里兰大学中搭建的报亭和相关系统处理了空间查询中的同义词问题。注意，所有的空间查询均可归为两类：

*基于地点的查询*给出地点 X，传统上使用经纬度坐标值作为参数，并返回与 X 相关联的一组特征集合；以及

*基于特征的查询*利用特征 Y 作为参数，返回与 Y 相关联的一组地点集合。

这些查询还可用两个函数来刻画，其中一个函数为另一个函数的反函数。基于特征的查询也被称为“空间数据挖掘”。[3] 虽然特征通常为空间参照数据（例如作物类型、土壤类型、地带和速度限制）的特性（或称为属性），但是它们及其底层的空间参照数据域的解释可能更宽。报亭把它们转换成由新闻文章的集合组成的非结构化数据域，

其中新闻的地点通过文字标明；而各种特征则为主题。转换这些概念后，基于地点的查询会返回提及特定地点或区域 X 的所有主题和文章，而基于特征的查询则返回与主题 T 有关的文章或只在文章 Y 中提及的所有位置和区域。注意，报亭不需要用户提前指定 T。如果未指定，则主题会按重要程度排序，其中的重要程度可通过多种标准定义，包括但不限于包含的文章数量。下文举出了两个典型的查询示例：位置 X 处正在发生什么？主题 T 或文章 Y 的发生地点在哪儿？

它们的执行通过构建空间数据的索引来加速，最好在批量加载的过程中立即构建所有的索引[36]，如 Hjaltason 和 Samet 的工作[12]所述。在当空间数据拥有确切的地理信息和数值信息时，构建索引相对容易。然而，由于所有数据都是非结构化的，所以报亭中数据的描述方式并非如此。具体来说，位置和特征数据都只是词语的集合。从空间数据的角度来看，其中的某些数据可以解释（但并不要求这么做）为地点的名称。换言之，空间数据通过文本（称为“地名”）而不是几何数据描述，也就是说会有一些模糊性。这种模糊性既有好处，也有坏处。好处是，从几何的角度来看，文本规格从点和空间范围两方面解释了数据，这点与参数传递中的多态类型相似（它构成了面向对象编程语言中继承的基石）。例如，在地理上，某个城市可以用某个点（比如它的形心）或与其边界对应的区域来确定。选择何种方式取决于所激活的查询界面的缩放尺度。坏处是，我们无法确定搜索项是否一直是地理位置。例如，在“Michael Jordan”中，“Jordan”是指国家，河流，还是姓？上述解答过程称为“地名识别”。[18]不仅如此，如果它是地理位置，那么如果有很多名称相同的地理位置实例，它指哪个位置。例如，“伦敦”指英国伦敦市、加拿大安大略省伦敦市，还是其他的地方？上述解答过程称为“地名分辨”。[19]在部署报亭和相关系统时，如何正确无误（或基本无误）地弄清这些模糊点是我们面临的主要技术挑战。

报亭的用户界面

报亭的目标不仅是提供一种不同的新闻阅读过程，更重要的是另一种体验。在报亭中进行查询时，用户需要选择感兴趣的区域，然后找到相关的关联主题和文章（请访问 http://newsstand.umiacs.umd.edu 体验报亭的界面）。主题和文章的展示由地点和缩放尺度确定，它们共同决定了查询的空间范围，或者说感兴趣的区域。“感兴趣的区域”这一概念有两种解释方式，一种从内容角度，一种从新闻源角度。用最简单的方式来说，对于感兴趣的区域，系统展现了相关的文章，但并没有预先限制发表这些文章的新闻源的地点范围。其次，通过明确指定新闻源（比如《*纽约时报*》和《*华盛顿邮报*》）、语言、用文本方式指定空间区域（比如限定新闻源的范围为爱尔兰）、或在报亭的地图上圈出感兴趣的区域（比如覆盖爱尔兰和英国的方框），可以把新闻源限定在可用新闻源的一个子集内。用户也可以约束空间区域和新闻源；它们不一定要相同。这个功能相当有用，因为它允许用户了解世界的某个区域如何看待另一个区域发生的事情。例如，用户可能希望了解英文媒体如何看待和解读中东的动态。这种结果类似于情感分

图 2.报亭的头条模式：（a）查询“2014年3月26日主题 T 或文章 Y 正在何处发生？”时的截屏示例；以及（b）与奥巴马/普京关系主题相关联的图片的子集，其中重复和近似重复的图片已经置灰。

(a) (b)

析。其他的应用包括为投资者监控热点地区、国家安全以及得到疾病扩散的最新消息，参见 Lieberman 等人的论文。[24]

图 1a 为查询“2014 年 3 月 26 日地点 *X* 处正在发生的事件（What is happening at location X on March 26, 2014?）”时报亭输出的截屏。该图中使用了报亭的“地图模式”。*X* 是非洲、欧洲和美洲部分地区。图中包括了一则有关奥巴马 / 普京的文章片段，其中论及了莫斯科。我们把地图上的每个图标或标识称为“标记”，其代表了一组主题相同和 / 或不同的文章集合，其中所有文章共有的主要性质为它们均提及了对应的地图位置。标识的类型说明了新闻类别（比如普通新闻、商业、科技、娱乐、健康和体育），范围涵盖了与该位置关联的大多数文章主题。用户可以切换屏幕顶部的相应按钮状态选择其中的一个或多个类别。

图 1b 为一个信息提示框，包含了与莫斯科或奥巴马 / 普京关系相关的主要主题中的代表性文章。报亭通过对所有文章使用聚类过程获取这些主题。当用户把鼠标停在莫斯科上时，系统生成了该信息提示框。这种停留的行为还会让地图上与该代表性的文章相关联的所有其他地点的标记变成橘球。在本例中，这些地点部分与奥巴马 / 普京关系涉及的或受其影响的国家对应。某些地点没有出现在截图中，可能是因为它们处于当前可见的地图的地理范围之外（比如北美和远东）。

当用户把鼠标放在标记上时，系统会生成迷你地图和标题（未在此处展示），其中会包含大地图之外的感兴趣区域。用户的这种行为会让橘球会出现在适宜的位置上，展现了代表性文章的地理范围。迷你地图这一工具允许用户查看所选文章的地理焦点，同时不必离开主地图上的感兴趣区域。而且，它与当前的缩放尺度无关。

报亭通过抓取网页收集数据。它的主要数据源为世界上以真正简易聚合（RSS）订阅源的形式存在的成千上万个单独的新闻源。

主地图和迷你地图上的蓝球说明了与用户鼠标当前停留的位置（此处为莫斯科）名称相同的其他地点。在允许迷你地图中包含名称相同的其他地点后，迷你地图的地理范围可能会超出橘球的范围。蓝球支持检测地名分辨错误。

迷你地图上的黑球标出了用户鼠标当前停留的地点，即莫斯科。迷你地图上的上下箭头允许用户滚动查看橘球和蓝球，并输出对应的地点名称。滚动查看蓝球时，系统支持对地点名称的解释进行排序。迷你地图上的绿球和红球对应滚动过程中的当前蓝球和橘球。把鼠标放在迷你地图上的橘球上时，会出现地点的名称；而放在蓝球上时，因为所有的蓝球名称相同，系统会在迷你地图上同时显示该地点及其上级地点的名称（如“Moscow（莫斯科），ID，United States（美国）”）。

图 1c 中的信息提示框展现了代表性文章的标题，这些代表性文章属于与莫斯科关联的各主题，而莫斯科则是用户鼠标最近停留的位置。单击与这一地点关联的标题信息提示框中的 > 标记后，便可得到该图。单击标题列表中的某一标题后，系统会展示汇总信息提示框（见图 1a），并在旁边展示对应的迷你地图。它也是在用户把鼠标放在标记上后生成的。注意，当用户滚动查看主题中的标题时，迷你地图上的橘球（不是蓝球）会发生变化。汇总信息提示框还包含了相关图片、视频和其他文章的链接。单击汇总信息提示框上的标题后，系统会展现文章的全文。而且，如果语言不是英语，还会有一个通过翻译包（如谷歌翻译或微软翻译）把全文和 / 或标题翻译成英文的选项。

新闻源的域可通过语言、地理区域或国家以及特定的报纸加以限制，代表性的文章从这些源的文章

中抽取。该功能通过使用“settings（设置）”按钮（位于图 1a 中屏幕右下的角落）配置和选择合适的过滤器实现，如图 1a 中靠下的灰色部分所示。注意，用户还能通过地点或（多个）关键词进行搜索，并通过控制显示的滑动条改变展现的标记数量。

图 2a 为查询 2014 年 3 月 26 日“主题 *T* 或文章 *Y* 正在何处发生？（Where is topic T or article Y happening on March 26, 2014）”时，报亭输出的截屏。这是报亭的“头条模式”。*T* 为主题之一，其中的代表性标题展示在左侧底部的面板内，按重要性度量排序。重要性按照显著性、距现在的时间和频率定义。虽然本应考虑主题到达的速度 / 加速度，这是个更好的度量标准，因为主题最终会失去其时效性。显示的标题为曾经点击过的标题。当用户鼠标放在上面时，它会被突出显示（通过变灰），此处对应的是奥巴马 / 普京关系的主题。单击标题后，会出现与标题相关的详情（比如更为详细的描述，相关文档、图片和视频的数量），如图 2a 左侧顶部的面板所示。同时，还可通过继续点击鼠标访问这些内容。

把鼠标放在图 2a 左侧底部面板上并点击后，系统还会在地图（右侧面板）上与主题关联的主要地理位置处展示合适的标识（类别标记）。在本例中，这些地点部分对应与涉及奥巴马 / 普京关系，或受其影响的国家，包括美国和俄罗斯。把鼠标放在右侧面板的地图上后，会出现很多信息提示框和关联的迷你地图，其中迷你地图与提示框的语义相同，如图 1 中“*X* 处正在发生什么？（What is happening at X）”的查询所示。具体来说，橘球支持用户区分临近的多个地点（比如网球簇中的英国伦敦和温布尔顿），而蓝球则标出了名称相同的其他地理位置实例【如“美国宾夕法尼亚州莫斯科镇（Moscow, PA, United States）”】。

使用地图和头条模式后，用户可以获得与各簇关联的图片和视频集合。而且，报亭会检测出重复或近似重复的图片，在视图中予以隐藏。这是一种功能强大的特性；首先，系统使用了与文章相关联的词或其语义找出相似的图片，尔后通过经典的图像相似性方法（包括分层颜色直方图[5]和尺度不变特征转换算法，即 SIFT）检测出相似图像中的重复。[25] 图 2b 说明了锚定在莫斯科与奥巴马 / 普京关系主题关联的此类图像的子集。

正如之前论述的那样，报亭的最终目的是把地图作为展现与空间相关的信息的媒介，因此它不限于新闻文章；也就是说，它还可用于搜索结果、图片、视频和推文。它还支持新闻汇总，深层探索，甚至通过发现新闻中的模式进行知识获取。把主题和类别与组成的文章中所提及的地点相关联后，便可直接得到这一结果。例如，查询可以被链接在一起。从这个意义上讲，有趣的主题可能与法国巴黎相关联，但在浏览橘球时，相同的主题或许又会与英国伦敦相关联。此时，用户可把定点设备移到伦敦上面，单击后，可以找到提及伦敦的其他相关主题和其他地点，然后用户又可以通过在地图查询界面上移动而转换到那些地点。这种无限的链接只在地图模式下提供，因为此时查询是基于地点的；在头条模式下，因为查询是基于主题的，所以除非用户使用了关键词搜索，否则地图上的标记限于排名最高的主题所对应的地点。

报亭还支持计算发病中心的簇，即簇中与疾病名称对应的最常用搜索项【比如图 3 中“2014 年 3 月 26 日的欧洲（Europe on March 26, 2014）】。另外，用户也可使用相同的理念，在簇中找出与人名或品牌名对应的最常用搜索项。在分别把“层”的参数设置为“疾病”，“人”或“品牌”后，便可找到此类搜索项。

相关研究

很难把报亭与现有的新闻阅读器进行比较，因为所有流行的新闻阅读

图 3. 报亭截屏展示了2014年3月26日欧洲各国中提到某疾病名称的簇；用户把鼠标放在西班牙巴伦西亚上，疾病名称为乳腺癌。迷你地图上的橘球展现了世界上的其他地点，与那些地点关联的簇提到了乳腺癌。

器（比如 Pulse）尚未具有使用地图阅读新闻的功能。新闻阅读系统（比如微软必应新闻、谷歌新闻和雅虎新闻）用经典的线性方式展现新闻，把来自不同源的新闻按主题归类。从根据用户所在位置聚合相关文章和主题的角度来看，这些提供者均含有某些位置特征。聚合通常依照邮编或市 - 州规定完成。例如，对于邮编 20742，主题可能提到了“马里兰州帕克分校”。谷歌新闻好像实现了这种功能。至少据我们所知，在谷歌搜索中使用地名作为搜索词可以得到类似结果。例如，确定用户的邮编为 20742 后（比如，在缺少获得用户所在当地区域的其他规定时，使用用户的 IP 地址），谷歌新闻会返回提及“马里兰州帕克分校”或“马里兰大学”的主题，因为已知它们与该邮编相关联。另外，主题的结果列表主要基于新闻源（通常为报纸）的地点，而不是报道的内容。新闻源包含了组成主题的多篇文章。在上述示例中，展现的主题数量是有限的。除了排除与用户位置无关的主题外，使用这种限制并无其他特别的原因。还请注意，在这些示例中，在决定向用户展现的内容时，没有使用文章重要性的概念。

有趣的是，流行的新闻阅读器均没有使用地图展现文章，虽然它们只要在地图平台上采用糅合（mashup）便能实现这一功能。HealthMap[10] 确实使用地图来展现疾病的暴发，其中的地点从疾病报告的日期栏或 ProMed 报告的元数据处获取。使用地图展现疾病报告与报亭的“疾病层”功能相似（见图 3），不过报亭中的地点来源于文章的实际文本。它还与我们在抽取网络上的空间 - 文本助力文档检索（STEWARD）系统中的实现方式相似 [23]。该系统利用了 ProMed 报告，也可展示疾病随时间的传播情况。[16] 注意，虽然支持糅合（mashup）的地图平台能够放大，但除了报亭之外，尚没有地图平台把缩放与获取更多文章的能力相结合。

过去，有些系统试图理解和展现新闻文章中的地理位置，但其中的大多数已经无法找到，或无法访问。例如，路透社的 NewsMap、*华盛顿邮报*的 TimeSpace、BBC 的 LiveStats 以及 AP 的 Mobile News Network（移动新闻网络）均试图根据提交文章的新闻通讯社的地点把新闻文章与大致的地理位置关联起来。因此，向迈阿密新闻通讯社提交的文章将会与迈阿密所有的邮编关联。与报亭不同，AP Mobile News Network 似乎不打算分析单独的文章，进而确定主要的关联地点，或地理焦点，或文章中提到的其他重要地点。

把报亭和网络搜索及推荐系统的商业服务（比如评论站点 Yelp 和 TripAdvisor）进行比较后，我们也得出了一些有用的信息。区别在于，在那些系统中，在感知空间实体前，需要在系统的数据库中明确输入以地址或 GPS，或经纬度值记录的空间信息；因此，它们能支持对空间信息的探索。报亭则起着两种作用：发现输入数据中的空间信息，这些信息通过文本描述，通常具有模糊性（需要纳入其他信息，有些未在输入数据内）；以及探索性的作用，其中的功能与推荐系统的功能相似，虽然推荐系统对地图查询界面的重视程度较弱。

报亭的架构

在 Rudyard Kipling 于 1902 年出版的《*原来如此的故事（Just So Stories）*》中，他对理解新闻所需的关键因素做出了或许是最好的阐释：“我雇佣了六个诚实的仆人（他们教会了我所有我知道的事情）；他们的名字是什么、哪儿、什么时候、怎么样、为什么和谁（What，Where，When，How，Why，Who）。”报亭关注于“什么”和“哪儿”，但较少关注“什么时候”，“什么时候”通常指的最近的时间。此处，我们先关注“什么”，随后再关注“哪儿”。

报亭通过抓取网页收集数据。它的主要数据源为世界上以真正简易聚合（RSS）订阅源的形式存在的成千上万个单独的新闻源；RSS 是在线出版中广泛使用的 XML 协议，非常适于报亭，因为它只需要一个标题、简短描述以及每则已刊发新闻的网络链接。RSS 2.0 还允许加入可选的发布日期，这可帮助确定文章距现在的时间，或者说“新近程度”。报亭现在为 10,000 个新闻源构建了索引，每天处理约 50,000 篇新闻文章。它使用名为地理标记的过程确定文章中提及的地理位置，而且设法确定文章的地理焦点或中心点，即文中提到的关键地点。

报亭还根据内容相似性把新闻文章按主题归类（称为“聚类”），所以与相同事件有关的文章会被归为相同的簇中。聚类的主要目的是自动对新闻文章进行分组，对描述相同新闻事件的文章进行归类，形成名为“文章簇”的新闻文章集合（之前文中也将其称为“主题”或“簇”）。之后，每个簇便只包含当前得到的输入中与特定主题相关的文章。新闻文章进入此阶段后，报亭会把它们归入新闻簇中。这本质上是一次性的过程，也就是说，一旦文章加入簇后，它会一直在簇中存在。报亭永远不会重新处理文章或重新对文章进行聚类。因为进入报亭的文章吞吐率很高，而且报亭的文档聚类系统需要能够快速处理文章且保持优质聚类输出，所以这点符合我们的期望。这种版本的聚类算法具有“在线”的特性。

给定上述需求后，报亭使用了领导者—追随者聚类 [7] 算法。该算法允许从使用词频——逆文档频率

（TF-IDF）的搜索项 - 向量空间度量和[35]时间维度两个方面进行在线聚类。对于每个簇，报亭维护了一个搜索项形心和时间形心，分别对应于簇中所有搜索项 - 特征向量的均值和文章的发布时间的均值。对新闻文章 a 进行聚类时，报亭检查是否存在搜索项与时间形心至 a 的距离小于固定截断距离 ϵ 的簇。如果存在一个或多个备选聚类，则把 a 加入最近的备选簇，并更新簇的形心；否则，报亭创造一个新的簇，其中只包含 a。

报亭的在线聚类算法依照其“重要性”的概念对簇进行排序。“重要性”由以下几个因素决定：

*文章的数量。*簇中文章的数量；

*簇中唯一的新闻源的数量。*例如，如果多家新闻源报道了在加利福尼亚州欧文市发生的事件，特别是如果有一些位于相隔较远的洛杉矶（距欧文市约 50 英里），则该事件属于重要事件。

*簇的传播速度。*有关重要事件的新闻会在较短的事件内被多家新闻源报道；以及

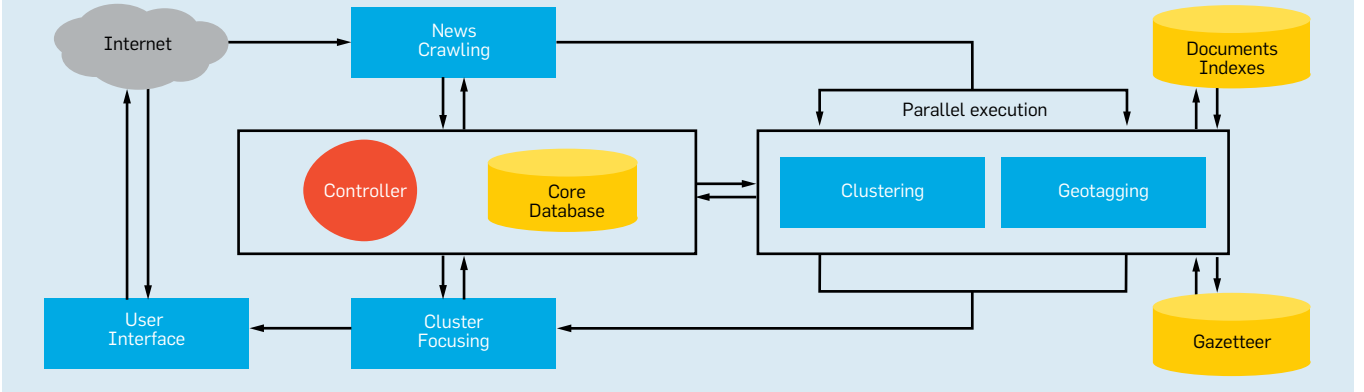
*添加的时间。*最近加入簇的时间。这是报亭用户可以设置的选项，可让系统忽略前三个因素。

当使用前三个因素对簇进行排名时，报亭必须选择簇的代表性文章，这是一种二级排序。该文章的性质可能因报亭用户的设置不同而不同，或是最近的文章，忽略对应簇的重要性（第四个因素），或是根据簇的重要性进行筛选；其中的选择范围或是声誉最好的新闻源的文章，或是最新的文章。在地图模式下，报亭在当前的观察窗口中展现的簇须含有最显著的主题，这点相当重要；仅仅在地图上展现排名最高的主题可能无法给广大的受众提供有用的结果，因为这些主题倾向于聚集在特定的地理区域中。这种情况反映了大型报社的新闻覆盖不均的情况，因为它们倾向于关注自己的地理区域。在报亭中，主题选择是在显著性和范围之间进行权衡。为了达到平衡，报亭把展现窗口细分为规则网格，要求每个网格区域内包含的主题数不多于最大主题数。展现的主题按显著性和距现在的时间降序选择，该方法确保热门主题在整个地图范围内良好分布。

报亭也能测定与簇关联的地理焦点或中心点；通过与地点特征对应的聚类过程，系统加快了这一测定过程。只要某个簇是最重要的簇之一，或其地理焦点与最重要的簇之一的焦点相同，则报亭会在该簇的地理中心点的位置处展现该簇，其中地点的数量通过地图右上角的滑动条调节。因此，与最重要的簇相关联的位置也是地图中含有数据的位置。这种展现通常会利用与新闻类别相对应的标识，如图 1 所示。然而，除了展现与簇相关联的类别标识之外，通过要求用户设置合适的“层”参数（如图 1a 所示），报亭还能展现与簇中最流行的搜索项（我们称之为“关键词”）相对应的文本。另外，通过设置层参数为“地点”，用户还能查看作为地理焦点的地点的实际名称。

在设计报亭的架构时，最重要的准则是规模化和对单篇文章的快速处理[20]（见图 4）。其他的目标包括，尽快（网上发布后几分钟内）展现最新的新闻，以及健壮，不易崩溃。通过把采集和处理细分为多个模块，报亭的架构满足了这些准则。其中的每个模块均可在分布式计算机集群中的不同计算节点上单独运行。图中勾勒出了计算管道中一连串模块对文章的处理过程。因为每个模块可能在不同的节点上运行，给定的文章可能会在系统的多个不同计算节点上处理。设计模块时，我们还支持任何模块的多个实例在一个或多个节点上同时运行。因此，依照其收到的新闻数量，报亭能够启动足够多的模块实例来进行处理。每个模块接收输入，然

图 4.报亭架构的高级概述我们把系统设计成管道，其中各单独的处理模块独立运行。中心控制模块通过把工作分配到另一模块并跟踪管道中的文章实现文章处理工作的编排。



后把输入存入作为同步点的 PostgreSQL 数据库。用户在报亭界面上的动作（比如缩放、平移和选择）会自动转化成 SQL 查询语句，PostgreSQL 数据库会返回查询结果。

地理标记

报亭从新闻文章中抽取地理位置（称为“地理标记”），并与地理信息检索的成果息息相关。在该领域的现有成果中，有很多处理了如何找出网站和单个文件的地理范围这一问题。在新闻文章的场景中，报亭区分了三种类型的地理范围：[26]

提供者。出版者的地理位置；

内容。文章或主题内容的地理信息；以及

服务。基于读者的所在地点。

报亭依据文章的内容确定文章的地理范围，同时还设法使用已知的提供者的范围以及通过学习得出的服务范围。

报亭扩展了我们之前在 STEWARD 中取得的地理标记成果[23]，以支持对暗网中的文档进行空间 - 文本查询。虽然 STEWARD 技术可用于任意的文档集合，但是报亭包含了其他的模块和功能。为提高新闻文章的处理效率，我们专门设计了这些模块和功能。STEWARD 处理各文档时，通常独立处理单个文档而不考虑和所有其他的文档的相关性。而把通常源于多个新闻源的文档归类为主题簇后，报亭利用了与主题相关的多个文档版本和实例，这样便可改进地理标记，让用户更容易获取相关文章。

地理标记包含两个过程：地名识别和地名分辨。地名识别涉及地理 / 非地理的模糊性，其中一个给定的短语可能指一个地理位置，也可能指其他类型的实体（比如，提到“华盛顿”时，需要确定它是地点，还是其他实体，比如人名）。别名的使用是第二个问题，其中多个名称指向同一个地理位置【比如“洛杉矶（Los Angeles）”和“LA”】。地名分辨又称为“地理名称的模糊性”或多义现象，涉及地理 / 地理的模糊性，其中给定的名称可能指多个地理位置中的任何一个。例如，“斯普林菲尔德（Springfield）”是美国很多城市的名称，包括马萨诸塞州的斯普林菲尔德市和伊利诺伊州首府斯普林菲尔德市。

图 5.居住在俄亥俄州哥伦布市附近的读者所拥有的地方词库的示例；注意，很多地方的地名与其他地区中名气更大的地名相同。

地名识别。现已可采用很多不同的方法进行地名识别，不过所有的方法均有一些共同的特点。识别的理念是在给定周围场景的情况下抽取“有用的”短语，或者最有可能提及地理位置和其他实体的短语。这些短语统称为文章的“实体特征向量”（EFV）。确定 EFV 时，最容易的方法是查找文章中已经在地名词典或地名和地点数据库中存在的短语。很多研究人员把这种方法作为他们的主要研究手段。[2] 具体来说，把地理信息与网页相关联的 Web-a-Where[2] 系统使用了一个小型的、结构良好的地名词典，其中包含了约 40,000 个地点。这个词典是通过收集人口数大于 5,000 的乡村和城市的名称而创建的。词典的规模严重限制了 Web-a-Where 实用的地理标记能力，因为这使得它无法识别人口较少的（往往是地方上的）地点。而这些地点在来自地方新闻源的文章中很常见。不仅如此，规模不大的地名词典也意味着 Web-a-Where 更容易出现地名识别错误。与使用较大的地名词典相比，它错失了分清地理 / 非地理的模糊性的机会。

为了处理较大地名词典中内在的地理 / 非地理模糊性，包括 Martins 等人[27]、Rauch 等人[33] 和 Stokes 等人[45] 在内的研究人员已经提出了多种启发式方法用于过滤可能错误的地名。MetaCarta[33] 识别了空间线索词（比如“city of”），以及特定格式的邮寄地址和地理坐标的文本描述。然而，在进行新闻文章的地理标记时，这种策略会引起严重的问题，因为每篇文章中通常都会包含报社总部的地址。由于 MetaCarta 主要关注更大的显著地点，这些格式良好的地理字符串在其地理标记过程中的作用太大，导致了很多地理标记错误。

其他地名识别的方法则扎根于自然语言处理中相关问题的解决方

案。例如，命名实体识别（NER）[47] 关注名词和名词短语，旨在从文章中找出与各种实体类别（如 PERSON、ORGANIZATION、 和 LOCATION）相对应的名词短语。标记为 LOCATION 的短语是最有可能成为地点的短语，被保存为实体特征向量的地理特征，而 ORGANIZATION 和 PERSON 短语则被保存为非地理特征。NER 方法可大致归类为基于规则的方法 [18,31] 或基于统计的方法。[17]

基于规则的解决方案以规则目录为特征，其中列出了地名可能出现的多种场景。另一方面，基于统计的解决方案依赖于标注后的文档语料库，使用这些语料库来通过类似隐马尔科夫模型（HMM）[47] 和条件随机场（CRF）的构件训练语言模型。[15] 在可以获得标注后的语料库时，HMM 和 CRF 被广泛使用。报亭的地名识别过程使用了 LingPipe 工作包 [4] 中的 NER 标记器。该标记器根据消息理解会议（MUC-6）和知名的 Brown 语料库提供的新闻数据进行训练。[9]

注意，NER 标记并不排除使用地名词典。与此相反，这些标记方法可作为过滤器或剪枝策略，用以控制对地名词典的查询量。缺点是，如果实体未被确认为潜在的地点，则会漏过该地点。这种情况偶尔会发生。报亭使用了最初从 100 多个地名词典中整理出的开源地名词典 GeoNames（http://geonames.org/），其中包括 GEOnet Names Server（GEOnet 名称服务器）和地理名称信息系统（Geographic Names Information System）。它现由世界各地的志愿者维护，包含了约 850 万个不同的地理位置的名称，其中约 550 万个名称是唯一的，而其他的名称用于进行地名分辨或解决地理 / 地理模糊性。由于报亭需要跟踪多语言下每个地点的名称，报亭的地名词典中包含了约 1630 万个地名。

最近使用报亭处理八百万篇文章的过程中，我们只碰到了约 60,000 个不同的地点，但有 40,000 多个面临地理 / 地理模糊性的问题，这使得地名分辨变得至关重要。地名词典还包含了有人居住的位置或区域的人口数量以及层次信息，包括包含该地点的国家和行政区划信息。这些信息在识别范围相当小的，地方上的地名时有用。我们把地名词典查找应用于每个地理特征 $f \in EFV$ 和匹配地点，以生成集合 $L(f)$，其中集合的数量与特征或 $|EFV|$ 的数量相同。

地名分辨。识别地名后，报亭使用地名分辨程序解决地理 / 地理模糊性。地理 / 地理模糊性分辨存在的问题与另一个更普遍的问题有关，即如何关联规范实体与文档中提及的每个名词短语，其又被称为“命名实体消歧”（NED）。为了消除名词短语的歧义，NED 采用了利用知识库（比如维基百科、DBpedia 和 Yago）匹配名词短语的方法。进行高级处理时，文档中提及的名词短语首先与多个备选实体进行匹配，然后根据知识库中这些实体的关联度进行消歧。例如，Milne 和 Witten[29] 使用了具有相关性度量的有监督学习方法，其中两篇维基百科文章的关联度依据两者均包含的导入链接数确定。类似的，Hoffart 等人 [13] 使用各种备选实体之间的“连贯性”来区分所有的名词短语。最近的某些研究已经设法把 NER 和 NED 模块整合为一个命名实体识别和消歧（NERD）模块 [34]，该模块扫描文档，然后输出其中提到的实体。

最简单的地名分辨策略是使用某种显著性度量（如人口）为每个识别出的地名分配一个默认的意义。包括 Amitay 等人，[2] Martins，[27] Purves 等人，[31] Rauch 等人，[33] 和 Stokes 等人 [45] 在内的很多研究人员已经结合其他的方法实现了这种策略。例如，根据给定地名的每种解释在语料库中出现的频率，MetaCarta[33] 用概率的形式为分配了“默认意义”。这种语料库由预先采集的有地理标注信息的文档组成。然后，它会根据其他的启发式方法（比如线索词和邻近地名的出现次数）改变这些概率。互联网空间感知信息检索（SPIRIT）项目 [31] 使用了与 MetaCarta 相似的技术。它查找了句子线索，在没有更强证据时，会为给定的地理参照分配“默认的意义”。

注意，使用基于语料库的默认意义和概率后，系统几乎无法识别文章中相对来说没有名气的地点参照（比如，世界上 2,000 多个名气较小的“伦敦”实例中的任意一个）。这需要选择当地报纸的文章作为正确的解释，因为预先创建的新闻文章语料库中极少会出现这些名气较小的地点。相比之下，报亭使用了我们称之为“地方词库”[22,32] 的概念。该词库与新闻源进行了关联，包含了位于新闻源的地理范围之内的地点集合。例如，居住于“俄亥俄州哥伦布市”的读者的地方词库包含了“都柏林”、“阿姆斯特丹”、“伦敦”、“特拉华”、“非洲（Africa）市 "、“亚历山大”、“巴尔的摩”和“不来梅”（见图 5）。对于哥伦布市区域以外的读者，由于其地方词库中没有包含这些位置名称，所以在名称相同时，他们可能先考虑更有名的地点。

使用地方词库与之前描述的使用提供者 - 和服务 - 范围对地理范围进行解释的情况类似。具体来说，报亭通过构建每个新闻源的文章语料库和收集语料库中提及的地方地理位置信息来学习自己的服务范围。该方法基于以下观察结果，书写新闻文章时假定了读者的位置。例如，当伊利诺斯州（比如芝加哥）的报刊文章提到地点“伊利诺斯州斯普林菲尔德市”时，限定词“伊

利诺斯州”或“IL”很有可能不会出现，因为读者可以自动做出正确的解释。另一方面，在讨论“斯普林菲尔德市”时，《*纽约时报*》则需要保留“伊利诺斯州”作为限定词，以避免可能出现误解。当用户在地图上进行放大，进而关注相对较小的地区时，地方词库非常有用，因为此时文章本质上更着重于地方特点。在这种情况下，有关提供者的知识在克服地理 / 地理模糊性时极为有用。

还可以把地方词库看成地名分辨使用的“辨别背景”。在地名分辨使用的相关流行策略 [2,27,31,45] 中，辨别背景被放在某个层次的地理本体范围内，此时需要找到可分辨文档内众多地名的地理区域。例如，Web-a-Where[2] 通过文档中多种形式的层次证据实现此类方法。证据包括最小辨别背景和邻近地名的包含关系（比如“马里兰州帕克分校”）。使用纳入地名词典内的层级结构以及各地名的置信度得分的简单评分算法后，系统找到了文档的地理焦点。Ding 等人 [6] 使用了类似的方法。MetaCarta[33] 和谷歌图书搜索没有使用计算地理焦点的概念，因此需要用户自己确定焦点。除了使用内容的位置外，Mehler 等人 [28] 还把文档与提供者的地点关联起来，有时候这等同于使用日期栏。注意，找到最小辨别背景背后的中心假设是，拟分析的文档只有一个地理焦点，这个地点在分辨该焦点内的地名时有用，但在分辨附带提及的较远的地名时没用。

还请注意，地方词库只是报亭使用的诸多地名分辨技术之一。由于事实上某些特性与多个纪录关联，即 $|L(f)| > 1$，所以需要该词库。具体来说，报亭通过启发式的过滤器分辨此类模糊的参照。这些过滤器会为依照人类阅读文章的方式为每个参照选择一组最可能的匹配。这些过滤器依赖于报亭最初的假设，即文章中的地点在地理距离、文章距离 [19] 和层次包含关系方面为彼此提供证据。“对象容器过滤器”便是其中的一种过滤器。通过指明包含关系的关键词或标点符号（比如“f_1 in f_2” 或者“f_1, f_2”），该过滤器找出了文章中被隔开的地理特征对 $f_1, f_2 \in EFV$。如果它发现了有地点对 (l_1, l_2) 符合 $l_1 \in L(f_1)$, $l_2 \in L(f_2)$ 且 l_1 在 l_2 之内，那么 f_1 和 f_2 被分别消歧为 l_1 和 l_2。举例来说，设 f_1 =“Brooklyn”，f_2“NYC.”。同时，设 $L(f_1)$ = {“Brooklyn, New York City,” “Brooklyn, Shelby County’’}，$L(f_2)$ = {“New York City, New York County,” “North Yorkshire County, U.K.”}。我们现在可以对 f_1 和 f_2 进行消歧，分别得到 l_1 =“Brooklyn, New York City”和 l_2 =“New York City, New York County.”。这种消歧得到了报亭观察结果的证明。在观察结果中，文章中出现的位置靠近、地理上邻近和层级关系明显的特征对不大可能是偶然发生的。在该策略的另一个例子中，当查询涉及多个地点列表时，报亭设法使用相近性、同级关系和显著性线索消歧。[1,21]

评价。为了获取报亭的地理标记的性能，可以通过把“层”参数设置为“地点”而不是“图标”使报亭展现地点的实际名称，而不是地点上的新闻类别图标。采用这种方法后，便可检测错误的地理 / 地理解释（比如把“洛杉矶”放在“智利”，而不是“加利福利亚”）以及把非地理名称归类为地理名称的错误（比如“南非”的“乔治”，而不是 2012 年在“佛罗里达州奥兰多市”发生的凯西·安东尼涉嫌杀女案审判中的“乔治·安东尼”），但反之不成立。

不仅如此，把鼠标放在地点 l 的名称 n 上时（在“地点”和“图标”层中），报亭会生成一个迷你地图，并在地图和迷你地图中用蓝球标记标出具有相同名称 n 的所有其他地点 k，使得至少有一个文章簇会与 k 相关联。这张迷你地图可以让报亭很快地发现地理标记错误。研究人员现在正在研究如何使用这种信息来学习得出更好的分类器。对于特定地名 n 的任何解释 k，只要一篇文章与解释 k 相关联，即使 k 可能不正确，系统也会认为 k 是 n 的解释，这样就把决定权放在了用户手里。基于 n 的任何解释 k 来标出系统认为提及特定地点 n 的所有文章后，蓝球可让报亭避免可能出现的地名分辨错误。根据报亭的一项假设，即至少有一篇文章与解释相关联（假设较低精度的地名识别会有 100% 的召回率），我们检查了所有提及 n 的文章以得出正确的解释。结果我们发现，对于在地名词典中列出的某地点，如果地点的解释精度较低但无任何遗漏，那么报亭在地名分辨时会达到 100% 的召回率。注意，从某种角度来说，报亭也正在对其响应进行排名，其中排名最高的响应与主地图上所查询的地点相关联，而排名较低的响应则与迷你地图相关联。

Lieberman 和 Samet 使用了人工制作的文章语料库进行了实验。他们的实验结果 [18] 说明，报亭的地名识别 [18] 和地名分辨 [19] 流程优于路透社的 OpenCalais 和雅虎的 Placemaker。它们均为不公开源码的商业产品，提供了公开的网络 API 支持自动对文档进行地理标记。同时，MetaCarta 系统 [33] 提供了相似的功能，可识别文本文档中的空间线索词（比如“city of"）、特定格式的邮寄地址以及用文本描述的地理坐标。

经验教训

构建报亭的经历教导我们，地名识别和辨别中的地理标记任务比我们最初预想的要复杂得多。例如，报亭的地理标记器本来可以使用更多

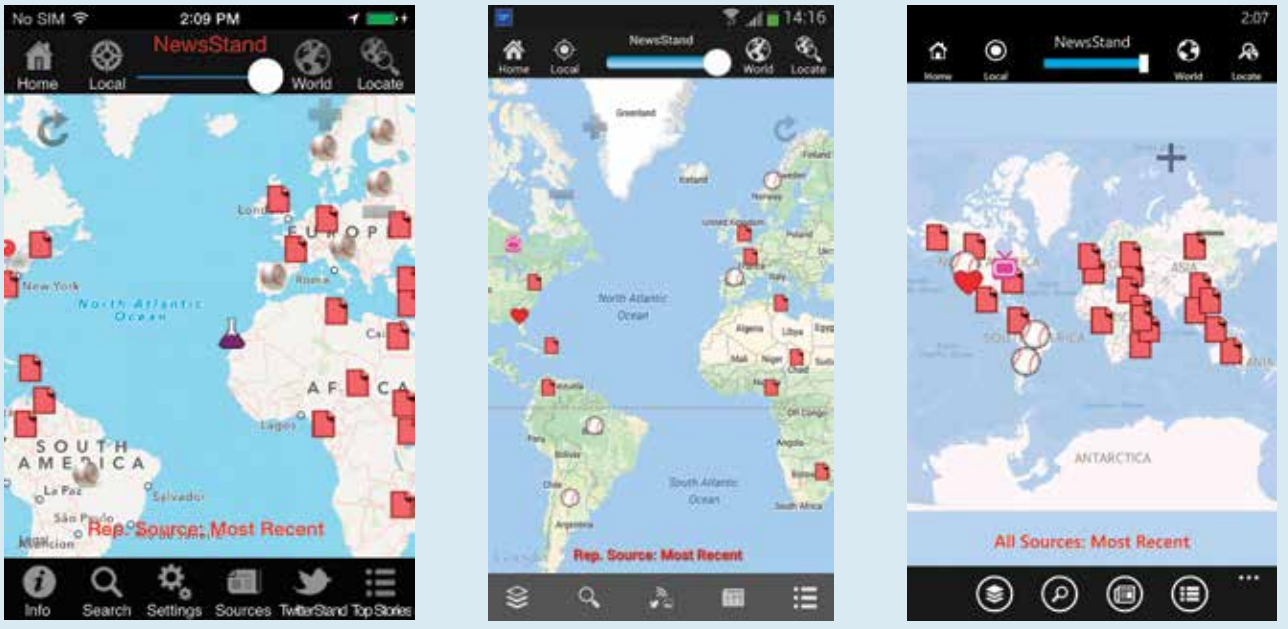
文档含有的语义提示来提高地理标记的正确性（比如地标和河流）。不仅如此，对TF-IDF框架进行修改，把各个空间上同义的搜索项合为一个搜索项而不是当成不同的搜索项后，也可以使用地理信息改进新闻文章的聚类。主要的难点在于评价报亭在上述任务中的性能。比较报亭和其他系统意味着不得不使用名为“语料库”的标准数据集。我们对地理标记任务的两个模块均进行了对比。我们把重点放在召回率而不是准确率上，最后得到了极好的结果。[18,19] 然而，这种评价方法有两个缺陷：数据集太小；而且由于新闻和语言一直变化，“语料库就如同僵化的库”。新闻数据的特征是流动的数据。评价更应该通过采样的方式进行，就如同检验 / 质量控制工作中那样。我们打算在未来这么做。

在网页浏览器中，报亭很好地使用了谷歌地图提供的地图 API 来展现主题。我们还对它进行了修改，使之可以适配必应地图和谷歌地球插件。尽管由于支撑平台的数量有限，插件导致了一些显示问题。报亭还被移植到带有支持手势的触摸屏界面的设备上（比如智能机和平板电脑）。虽然其中的用户界面有所不同，但用户可通过其中的网页浏览器进行使用 [42]。不仅如此，我们还开发了 [38] 在 iPhone、Android（安卓）和 Windows Phone 平台上使用的应用（见图 6）。报亭没有“公开”的 API，不过它的很多功能以及适配不同智能机平台的能力均使用了它的“私有”API。

由于基于浏览器的网络环境和移动设备的原生应用环境存在差异，所以需要改变用户行为或习惯。例如，网页上以地图为中心的应用最好只有一个页面，这意味着外部链接（比如通往报亭的新闻文章的链接）最好在单独的浏览器页签中打开，以保存报亭应用及其状态。如果新闻文章在相同的页签中打开，就无法实现这一功能。在单独的浏览器页签中打开外部链接也会导致不想要的结果。一个具体的例子是，用户不能使用“后退”键返回应用和应用之前的状态。与此相反，他们必须显式地关闭新打开的页签。此时，调用页签及其状态隐式地保存了。在原生的应用环境，此类问题不会发生。原生应用可以协调多个窗口之间的转换，从而提供更加友好的用户互动。不过，也牺牲了某些能力。比如在我们的例子中，一次只能打开一个链向新闻文章的外部链接。

在把报亭移植到多种不同的移动 / 智能机平台时，我们发现底层地图 API 的实现并没有遵守经典的制图原则。结果是，进行某些操作时（比如缩放和平移）会出现一致性问题。例如，一旦地点的名称出现在地图上后，在用户继续放大或平移时，只要该地点仍在窗口内，名称就应该能继续显示。[40] 有趣的是，移动和智能机平台上的某些地图应用不支持缩小后在整个屏幕上查看整个世界（比如谷歌地图和苹果地图的移动 / 智能机地图 API），因此即便整个世界在“当时”的地图 API 中已经存在，也需要继续平移才能看到世界的其他部分。[40] 在报亭中，这种现象尤其让读者厌烦，因为读者希望看到整个世界正在发生的事件。[40] 迷你地图部分缓解了这一问题，对于使用标题信息提示

图 6.报亭应用的截图：(a) iPhone、(b) Android（安装）、(c) Windows Phone 平台



框突出显示的特定文章，它用橘球标出了其中提到的所有其他地名。

在设计用户界面时，我们不得不考虑使用手势界面后无法在设备上悬停，这意味着在支持手势的平台上将不得不改变一些功能的实现方式。具体来说，当定点设备经过一个位置时，悬停支持用户观察当时正在显示或展开的现象的空间变异。手势界面要求使用轻拍（tap）或点击（click）来触发这种显示行为，因为手指在某个区域内的连续运动会被解释为一次轻拍或点击。因此，很难观察到空间变异。另一方面，缺少悬停意味着从地图位置 *l* 到另一地图位置 *b* 的过渡可以通过轻拍 *b* 处实现。相比之下，使用悬停进行从 *l* 到 *b* 的过渡时，可能需要采取某些特定的行为，而且这些行为会破坏系统的当前状态。

进行快速地图注记时，面临的设计挑战是把迷你地图放在靠近标题提示框和相关信息框的位置处。在注记的动态展现中，也会面临这种挑战（比如图 3 中的疾病名称，关键词以及人名和品牌名）。我们的目标是在平移和缩放时，按照互动的速度进行快速地图注记。这点可以通过为动态地图注记 [30] 开发且纳入 PhotoStand 系统的技术予以实现。[37]

结论

我们回顾了报亭系统的设计目标和功能。报亭系统使用地图来阅读网络中的新闻，并发挥了空间同义词的作用。报亭证明，从新闻文章中抽取地理内容会发掘之前未曾见过的信息维度，新闻确实可以被当成东南西北的首字母缩略词【NEWS（North，East，West，South）】。在互联网上，有地理标记的内容越来越流行，这使得在其他知识领域的系统中出现了与报亭相似的，令人叹服的应用。例如，情感 / 内容分析可以解释不同国家或不同语言的人群对同一篇新闻报道的不同理解方法，也可根据新闻、推文或其他数据摘要源进行热点分析。不仅如此，报亭还为新兴的计算新闻学领域做出了贡献。[8]

报亭现在为10,000个新闻源构建了索引，每天处理约50,000篇新闻文章。

未来的研究包括使用地图查询界面通过代表性的图片（比如 PhotoStand[37]）、视频或音频剪辑访问其他媒体。我们也正在设法纳入其他新闻源和信息源。例如，我们已经在报亭中纳入了 Twitter 的推文，由此创建了 TwitterStand 系统 [44]。该系统的理念是发掘大量的新闻文章作为某种类型的聚类用语料库，以便长度非常短、信息非常稀疏的推文使用已有的新闻簇进行聚类。在这一方法中，有趣的一面是由于推文长度相当短，它们往往很少有或没有地理内容。但是，当它们被聚类后，它们继承了地理信息，而这些地理信息则与推文关联的簇的地理焦点存在关联关系。我们从中发现的新结果是，焦点现在是用户的推文中涉及的地理区域，而不是用户发出推文时所在的地理区域（当发出推文的设备具有 GPS 功能时，很容易发现用户的所在位置）。在编写有关未来事件的推文时，这种焦点相当有用 [14]，但必须谨慎选择关注谁的推文。[11]

鸣谢

本文基于 Teitler 等人之前的论文。[46] 本文的研究资金部分来源于美国国家科学基金会（项目号 IIS-07-13501、IIS-08-12377、CCF-08-30618、IIS-10-18475、IIS-12-19023 和 IIS-13-20791）、美国住房和城市发展部政策发展和研究办公室、微软研究院、谷歌研究院、英伟达（Nvidia）、爱尔兰科学基金会 E.T.S. Walton 访问学者奖（E.T.S. Walton Visitor Award of the Science Foundation of Ireland）以及梅努斯镇爱尔兰国立大学地理计算国家中心。我们还要感谢 Larry Brandt、Jim Gray、Keith Marzullo、和 Maria Zemankova 的支持。

参考资料

1. Adelfio, M.D. and Samet, H. Structured toponym resolution using combined hierarchical place categories.In *Proceedings of the Seventh ACM SIGSPATIAL Workshop on Geographic Information Retrieval* (Orlando, FL, Nov. 5).ACM Press, New York, 2013, 49–56.
2. Amitay, E., Har'El, N., Sivan, R., and Soffer, A. Web-a-Where:Geotagging Web content.In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Sheffield, U.K., July 25–29).ACM Press, New York, 2004, 273–280.
3. Aref, W.G. and Samet, H. Efficient processing of window queries in the pyramid data structure.In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (Nashville, TN, Apr. 2–4).ACM Press, New York, 1990, 265–272.
4. Baldwin, B. and Carpenter, B. *Lingpipe*; http://alias-i.com/lingpipe/
5. Chum, O., Philbin, J., Isard, M., and Zisserman, A. Scalable near-identical image and shot detection.In *Proceedings of the Sixth ACM International Conference on Image and Video Retrieval* (Amsterdam, The Netherlands, July 9–11).ACM Press, New York, 2007, 549–556.
6. Ding, J., Gravano, L., and Shivakumar, N. Computing geographical scopes of Web resources.In *Proceedings of the 26th International Conference on Very Large Data Bases* (Cairo, Egypt, Sept. 10–14).Morgan Kaufmann, San Francisco, 2000, 545–556.
7. Duda, R.O., Hart, P.E., and Stork, D.G.*Pattern Classification, Second Edition.* Wiley Interscience, New York, 2000.
8. Essa, I. *Computation + Journalism:A study of Computation and Journalism and How They Impact Each Other*; http://www.computation-and-journalism.com/
9. Francis, W.N.A standard corpus of edited present-day American English.*College English 26*, 4 (Jan. 1965), 267–273.
10. Freifeld, C.C., Mandl, K.D., Reis, B.Y., and Brownstein, J.S.HealthMap:Global infectious disease monitoring through automated classification and visualization of Internet media reports.*Journal of the American Medical Informatics Association 15*, 2 (Mar. 2008), 150–157.
11. Gramsky, N. and Samet, H. Seeder finder:Identifying additional needles in the Twitter haystack. In *Proceedings of the Fifth ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (Orlando, FL, Nov. 5).ACM Press, New York, 2013, 44–53.
12. Hjaltason, G.R. and Samet, H. Speeding up construction of PMR quadtree-based spatial indexes.*Very Large Data Bases Journal 11*, 2 (Oct. 2002), 109–137.
13. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. Robust disambiguation of named entities in text.In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Edinburgh, Scotland, July 27–31).Association for Computational Linguistics, Stroudsburg, PA, 2011, 782–792.
14. Jackoway, A., Samet, H., and Sankaranarayanan, J. Identification of live news events using Twitter. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on Location-Based Social Networks* (Chicago, Nov. 1).ACM Press, New York, 2011, 25–32.
15. Lafferty, J.D., McCallum, A., and Peireira, F.C.N.Conditional random fields:Probabilistic models for segmenting and labeling sequence data.In *Proceedings of the 18th International Conference on Machine Learning* (Williamstown, MA, June 28–July 1).Morgan Kaufman, San Francisco, 2001, 282–289.
16. Lan, R., Lieberman, M.D., and Samet, H. The picture of health:Map-based, collaborative spatio-temporal disease tracking.In *Proceedings of the First ACM SIGSPATIAL International Workshop on the Use of GIS in Public Health* (Redondo Beach, CA, Nov. 6).ACM Press, New York, 2012, 27–35.
17. Leidner, J.L.*Toponym Resolution in Text:Annotation, Evaluation and Applications of Spatial Grounding of Place Names.* Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, U.K., Oct. 2006; https://www.era.lib.ed.ac.uk/bitstream/1842/1849/1/leidner-2007-phd.pdf
18. Lieberman, M.D. and Samet, H. Multifaceted toponym recognition for streaming news.In *Proceedings of the 34th International Conference on Research and Development in Information Retrieval* (Beijing, July 24–28).ACM Press, New York, 2011, 843–852.
19. Lieberman, M.D. and Samet, H. Adaptive context features for toponym resolution in streaming news. In *Proceedings of the 35th International Conference on Research and Development in Information Retrieval* (Portland, OR, Aug. 12–16).ACM Press, New York, 2012, 731–740.
20. Lieberman, M.D. and Samet, H. Supporting rapid processing and interactive map-based exploration of streaming news.In *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Redondo Beach, CA, Nov. 7–9).ACM Press, New York, 2012, 179–188.
21. Lieberman, M.D., Samet, H., and Sankaranarayanan, J. Geotagging:Using proximity, sibling, and prominence clues to understand comma groups.In *Proceedings of the Sixth Workshop on Geographic Information Retrieval* (Zürich, Switzerland, Feb. 18–19).ACM Press, New York, 2010.
22. Lieberman, M.D., Samet, H., and Sankaranarayanan, J. Geotagging with local lexicons to build indexes for textually specified spatial data.In *Proceedings of the 26th IEEE International Conference on Data Engineering* (Long Beach, CA, Mar. 1–6).IEEE Press, 2010, 201–212.
23. Lieberman, M.D., Samet, H., Sankaranarayanan, J., and Sperling, J. STEWARD:Architecture of a spatio-textual search engine.In *Proceedings of 15th ACM International Symposium on Advances in Geographic Information Systems* (Seattle, Nov. 7–9).ACM Press, New York, 2007, 186–193.
24. Lieberman, M.D., Sankaranarayanan, J., Samet, H., and Sperling, J. Augmenting spatio-textual search with an infectious disease ontology.In *Proceedings of the Workshop on Information Integration Methods, Architectures, and Systems* (Cancun, Mexico, Apr. 11–12).IEEE Computer Society, 2008, 266–269.
25. Lowe, D.G.Object recognition from local scale-invariant features.In *Proceedings of the Seventh International Conference on Computer Vision* (Corfu, Greece, Sept. 20–25).IEEE Computer Society, 1999, 1150–1157.
26. Markowetz, A., Brinkhoff, T., and Seeger, B. Exploiting the Internet as a geospatial database.In *Proceedings on the Workshop on Next Generation Geospatial Information* (Cambridge, MA, Oct. 19–21, 2003).
27. Martins, B., Manguinhas, H., Borbinha, J., and Siabato, W. A geo-temporal information extraction service for processing descriptive metadata in digital libraries. *e-Perimetron 4*, 1 (2009), 25–37.
28. Mehler, A., Bao, Y., Li, X., Wang, Y., and Skiena, S. Spatial analysis of news sources.*IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Sept.–Oct. 2006), 765–772.
29. Milne, D. and Witten, I.H.Learning to link with Wikipedia.In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (Napa Valley, CA, Oct. 26–30).ACM Press, New York, 2008, 509–518.
30. Nutanong, S., Adelfio, M.D., and Samet, H. Multiresolution select-distinct queries on large geographic point sets.In *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Redondo Beach, CA, Nov. 7–9).ACM Press, New York, 2012, 159–168.
31. Purves, R.S., Clough, P., Jones, C.B., Arampatzis, A., Bucher, B., Finch, D., Fu, G., Joho, H., Syed, A.K., Vaid, S., and Yang, B. The design and implementation of SPIRIT:A spatially aware search engine for information retrieval on the Internet.*International Journal of Geographical Information Systems 21*, 7 (2007), 717–745.
32. Quercini, G., Samet, H., Sankaranarayanan, J., and Lieberman, M.D. Determining the spatial reader scopes of news sources using local lexicons.In *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (San Jose, CA Nov. 3–5).ACM Press, New York, 2010, 43–52.
33. Rauch, E., Bukatin, M., and Baker, K. A confidence-based framework for disambiguating geographic terms.In *Proceedings of the HLT-NAACL Workshop on Analysis of Geographic References* (Edmonton, Canada).Association for Computational Linguistics, Stroudsburg, PA, 2003, 50–54.
34. Rizzo, G. and Troncy, R. NERD:A framework for unifying named entity recognition and disambiguation extraction tools.In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (Avignon, France, Apr. 23–27).Association for Computational Linguistics, Stroudsburg, PA, 2012, 73–76.
35. Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval.*Information Processing & Management 24*, 5 (1988), 513–523.
36. Samet, H. *Foundations of Multidimensional and Metric Data Structures.*Morgan Kaufmann, San Francisco, 2006.
37. Samet, H., Adelfio, M.D., Fruin, B.C., Lieberman, M.D., and Sankaranarayanan, J. PhotoStand:A map query interface for a database of news photos.*Proceedings of the VLDB Endowment 6*, 12 (Aug. 2013), 1350–1353.
38. Samet, H., Adelfio, M.D., Fruin, B.C., Lieberman, M.D., and Teitler, B.E.Porting a Web-based mapping application to a smartphone app. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Chicago, Nov. 2–4).ACM Press, New York, 2011, 525–528.
39. Samet, H., Alborzi, H., Brabec, F., Esperança, C., Hjaltason, G.R., Morgan, F., and Tanin, E. Use of the SAND spatial browser for digital government applications.*Commun.ACM 46*, 1 (Jan. 2003), 63–66.
40. Samet, H., Fruin, B.C., and Nutanong, S. Duking it out at the smartphone mobile app mapping API corral:Apple, Google, and the competition.In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems* (Redondo Beach, CA, Nov. 6).ACM Press, New York, 2012, 41–48.
41. Samet, H., Rosenfeld, A., Shaffer, C.A., and Webber, R.E.A geographic information system using quadtrees. *Pattern Recognition 17*, 6 (Nov./Dec. 1984), 647–656.
42. Samet, H., Teitler, B.E., Adelfio, M.D., and Lieberman, M.D. Adapting a map query interface for a gesturing touchscreen interface.In *Proceedings of the 20th International Word Wide Web Conference* (Hyderabad, India, Mar. 28–Apr. 1).ACM Press, New York, 2011, 257–260.
43. Sankaranarayanan, J., Samet, H., Teitler, B., Lieberman, M.D., and Sperling, J. TwitterStand:News in tweets.In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Seattle, Nov. 4–6).ACM Press, New York, 2009, 42–51.
44. Sarma, A.D., Lee, H., Gonzales, H., Madhavan, J., and Halevy, A. Efficient spatial sampling of large geographical tables.In *Proceedings of the ACM SIGMOD Conference* (Scottsdale, AZ, May 20–24). ACM Press, New York, 2012, 193–204.
45. Stokes, N., Li, Y., Moffat, A., and Rong, J. An empirical study of the effects of NLP components on geographic IR performance.*International Journal of Geographical Information Systems 22*, 3 (Mar. 2008), 247–264.
46. Teitler, B., Lieberman, M.D., Panozzo, D., Sankaranarayanan, J., Samet, H., and Sperling, J. NewsStand:A new view on news.In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Irvine, CA, Nov. 5–7).ACM Press, New York, 2008, 144–153.
47. Zhou, G. and Su, J. Named entity recognition using an HMM-based chunk tagger.In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, PA, July 6–12).Association for Computational Linguistics, Stroudsburg, PA, 2002, 473–480.

Hanan Samet（hjs@cs.umd.edu）是马里兰州马里兰大学帕克分校计算机科学系、自动化研究中心以及高级计算机研究所的杰出教授。

Jagan Sankaranarayanan（sjagan@gmail.com）是加利福尼亚州库比蒂诺NEC实验室的研究员；他在本文中的研究是他在马里兰大学帕克分校高级计算机研究所任助理研究员时进行的。

Michael D. Lieberman（mike.d.lieberman@gmail.com）是马里兰州劳雷尔（Laurel）市约翰霍普金斯大学应用物理实验室的研究员；他在本文中的研究是他在马里兰大学帕克分校攻读计算机科学博士时的一部分成果。

Marco D. Adelfio（marco@cs.umd.edu）是马里兰大学帕克分校计算机科学的在读博士。

Brendan C. Fruin（bcfruin@gmail.com）是华盛顿州西雅图Zillow公司的软件工程师；他在本文中的研究是他在马里兰大学帕克分校攻读计算机科学硕士时的一部分成果。

Jack M. Lotkowski（JackLotkowski@gmail.com）是马里兰大学帕克分校的本科生。

Daniele Panozzo（daniele.panozzo@gmail.com）是瑞士苏黎世瑞士联邦理工学院（ETH）的高级研究员；他在本文中的研究是他在马里兰大学帕克分校高级计算机研究所当进修生时进行的。

Jon Sperling（jonxsperling@gmail.com）是华盛顿特区美国住房和城市发展部政策发展和研究办公室的高级研究员。

Benjamin E. Teitler（bteitler@cs.umd.edu）在本文中的研究是他在马里兰大学帕克分校攻读计算机科学硕士时的一部分成果。

译文责任编辑：崔斌

DOI:10.1145/2661061.2661063

新的抽象对于实现 SDN 目标至关重要。

作者 MARTIN CASADO、NATE FOSTER 及 ARJUN GUHA

软件定义网络的抽象

软件定义网络 (SDN) 作为一种克服某些长期存在的网络难题的手段，近年来受到广泛关注。SDN 源自于两个简单想法：一是泛化网络硬件，使其提供一组标准的数据包处理功能，而不是一组固定的狭隘功能；二是将控制网络的软件与实现软件的设备解耦。这种设计有可能无需改变底层硬件，就能推动网络发展，并实现从特定应用程序相应抽象的角度表示网络算法。

图 1 比较了传统网络和 SDN 的体系结构。在 SDN 中，一台或多台控制机执行一个通用程序，这个程序通过计算一组数据包转发规则来响应某些事件，例如，网络拓扑的变化、终端主机发起的连接、流量负载的变动，或者来自其他控制机的消息。然后，控制机将这些规则推送到交换机，交换机使用专门的硬件有效实现所需的功能。

由于 SDN 没有规定如何实现控制机，因此，控制机可用于实现各种网络算法，包括最短路径路由之类的简单算法，以及流量工程之类的更复杂的算法。

人们已经利用 SDN 实现了很多新的应用程序，包括基于策略的访问控制、自适应流量监视、广域流量工程、网络虚拟化等等。[6,9,16,18–20,44] 原则上，可以在传统网络中实现任何这些应用程序，但是说起来容易做起来难：程序员必须设计新的分布式协议，还要解决实际问题，因为传统交换机不容易受第三方程序的控制。

早期 SDN 控制器平台开放了一个简陋的编程接口，这个接口提供的不过是底层硬件功能的瘦包装器。在存在更高级抽象的地方，这些抽象反映了传统网络中已有的结构，如拓扑或链路状态信息。但是现在，有越来越多的研究工作正在探索 SDN 如何改变控制算法的表示以及编写，使其变得更容易和更方便。现代操作系统提供了用于管理硬件级资源的丰富抽象，与此一样，我们相信，为了完全实现 SDN 的构想，我们需要类似的抽象。

这些抽象正是本文的主题。我们回顾了改进 SDN 编程模型和抽象

» 重要见解

- **SDN 是一种新的网络体系结构，它将控制网络的软件与实现软件的设备分离。**
- **通过全局显示网络状态，SDN 可大大简化很多网络算法的表达方式。**
- **SDN 还可能实现无需改变底层硬件，就能推动网络功能的发展。**
- **SDN 推动新的网络编程模型、系统抽象和验证工具的开发。**

图片由 AVN PHOTO LAB 提供

的近期研究以及正在进行的研究，重点放在以下方面：

全网结构：SDN 控制器是使用相对较小的紧耦合服务器集合构建的，这使其更容易接受可保持一致的全网结构（如拓扑、流量统计等）版本的分布式算法。

分布式更新：SDN 控制器管理整个网络，因此它们必须经常更改多台交换机的规则。在状态转换期间提供一致性保证的更新机制可简化动态程序的开发。

模块化组合：很多网络程序很自然地分成多个模块。提供组合编程接口的控制器使得以模块化组件的方式指定网络行为的正交方面变得容易。

虚拟化：将应用程序逻辑从物理拓扑中分离出来简化了程序、确保了隔离，并提供了可移植性。虚拟网络抽象还可以增强可伸缩性和容错。

形式验证：为了帮助程序员编写正确的程序，某些控制器提供了相应的工具，这些工具自动检查形式属性，并在出现意外错误时诊断问题。

这里，我们更加详细地探讨这些抽象。为了在共同的基础上展开讨论，作为 SDN 的具体例子，我们先介绍 OpenFlow。

OpenFlow

OpenFlow 规范除了定义了可用于与交换机通信的接口控制器外，还定义了交换机必须提供的标准功能集合：安装和删除转发规则的指令，以及关于数据流、拓扑和流量统计的通知。[31]

OpenFlow 交换机维护一张*转发表*，表中包含*优先规则*的列表。每条规则都有模式和操作，*模式*描述一组数据包，*操作*描述数据包的转换。当数据包到达交换机时，交换机查找模式与数据包首部匹配的规则，并应用关联的操作。如果有多条规则匹配，交换机应用优先级最高的规则的操作，而如果没有规则匹配，则交换机在 OpenFlow 消息中封装该数据包，并将其发送到控制器。控制器可以直接处理数据包，也可以将消息发回给交换机，指示它在转发表中安装或删除规则。表的最大大小由硬件限制决定，但是大多数交换机的空间可容纳至少是几千条规则。

为了支持流量监视，每条规则都有关联的计数器跟踪某些基本统计数据，如用该规则处理的所有数据包的数量和总量大小。控制器可以使用 OpenFlow 消息读取这些计数器。它们还可以配置交换机的物理端口，创建限制流量速率的队列；或者提供最低带宽保证（这是对于实现流量工程应用程序非常有用的功能）。

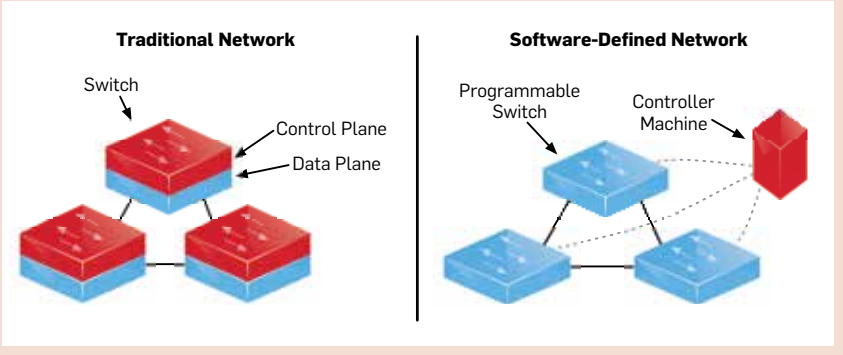
作为示例，请考虑附表：从上到下读，这些规则阻止所有 SSH 流量，转发目的地分别为主机 10.0.0.1 和 10.0.0.2 的非 SSH 流量并将其分别输出到端口 1 和 2，将所有其他流量转给控制器做进一步处理。

全网结构

SDN 的主要优点是，控制器可使用分布式算法计算全网结构，这些结构提供了全局的网络状态，而分布式算法可以强力保证这些结构在各控制器间的一致性。在传统网络中维护这些全网结构实际上并不可行，因为在传统网络中，控制权分散在更多的设备上，但是如果使用这些结构，很多应用程序的逻辑可以变得简单得多。例如，对表示拓扑的结构计算 Dijkstra 算法，可以实现最短路径路由。[41]

示例。为演示起见，请考虑维护生成树的任务，这棵生成树将网络中的交换机连接起来。这样的树可用于转发广播流量，而不会造成任何转发循环的危险。设计分布式算法来构造和维护生成树其难无比，因为它必须在任意拓扑中都能正确起作用，当出现设备或链路意外失效之类的事件时，它可以快速重新收敛到新树。

图 1.传统体系结构与软件定义体系结构。



OpenFlow 转发表示例。

优先级	模式	操作	计数器
30	TcpDstPort = 22	丢弃	⟨7156, 124⟩
20	IpDstAddr = 10.0.0.1	转发 1	⟨2648, 38⟩
10	IpDstAddr = 10.0.0.2	转发 2	⟨14184, 246⟩
0	*	控制器	⟨1686, 14⟩

传统解决方案。构建生成树的经典方法是使用*生成树协议*[36]——一种完全分布式协议，交换机按照此协议，使用成对通告报文，定期与其相邻的交换机交换信息。交换机运行分布式领头者选举协议共同选出一个根节点，然后从该节点开始逐渐构造生成树，启用和禁用链路来选择通向根的最短路径，并使用交换机标识符来打破僵局。请注意，生成树协议的实现需要邻居发现、领头者选举，以及实际树构造算法，但是因为这些组成部分都依协议而定，因此需要类似功能的其他协议无法轻易重用它们的逻辑。而且，当拓扑更改时，计算新树的时间随着最长无环路路径的规模而增大。

SDN 解决方案。大多数 SDN 控制器都提供了一套在很多应用程序中出现的通用功能，比如拓扑发现和链路故障检测，此外还维护跟踪网络状态信息（如主机位置、链路容量和流量矩阵等）的结构。存储这些信息的数据库通常称为网络信息库 (NIB)。[25] 使用 NIB 时，生成树的 SDN 实现可以远比其分布式实现来得简单：每当拓扑变化时，它只需使用 Prim 算法根据该拓扑计算生成树，然后在交换机上安装沿树转发的规则即可。

更丰富的应用程序。通过为程序员提供有关整个网络的状态的信息，NIB 还使得实现更丰富的应用程序变得容易，比如流量工程应用程序，这类应用程序在传统网络中很难实现。[11] 例如，B4 和 SWAN 系统使用 SDN 来平衡数据中心之间广域链路上的负载，从而达到比传统方法更高的利用率。[18,19] 这些应用程序需要分布式控制器，这些控制器通过 NIB 自动管理在多个控制器之间复制的数据。[26]

使用多个控制器解决了可伸缩性和容错等重要问题 — 例如，如果一个控制器的负载变高，或者其与交换机连接的链路失效，那么另一个控制器可以接手其工作。但是，由于控制器的数量通常很小，因此这些控制器可使用 Paxos 之类的算法 — 这类算法在完全分布式环境中不会扩展。因此，虽然控制器确实使用分布式算法，但是它们更加简单，而且通常比传统协议收敛更快，因为控制器比交换机少。

SDN 提供了全网结构，并使得通用分布式编程抽象可以实现一次，然后在多种应用程序中重用，所以大大简化了很多网络程序。

讨论。SDN 提供了全网结构，并使得通用分布式编程抽象可以实现一次，然后在多种应用程序中重用，所以大大简化了很多网络程序。这样的重用在传统网络中实际上不可能实现，因为传统网络的转发和控制紧密耦合在每台设备上，领头者选举等功能的实现束缚于特定协议，而设备有多种多样的 CPU、内存和存储能力。

分布式更新

在传统网络中，哪怕配置更新仅仅是最终一致，这通常也是可接受的。例如，如果网络配置由于链路失效而重新计算，那么数据包可能以原始状态通过交换机一次，然后以更新状态二次通过交换机。这可能导致转发环路或丢弃数据包等行为，但是由于大多数网络只提供最大努力递送，因此只要网络最终收敛到新的状态，那么状态转换期间发生的瞬时错误也是可接受的。但是，最终一致更新在 SDN 中并非始终够用。例如，SDN 控制器除了管理转发规则外，还可能管理过滤规则，这些规则对于确保不变量（如访问控制或共用网络的租户流量之间的隔离）至关重要。如果配置更新以仅仅是最终一致的方式传播到交换机，那么在状态转换期间，这些不变量可能很容易被更改。

程序员有时候可以绕开这些问题，他们可以小心对更新进行排序，让数据包只走那些其配置已完全传播到网络的路径。例如，程序员可能先更新入口交换机，然后检查网络内部所有部分更新的路径在状态

转换期间，是否通过其他方式都不可到达。但是，手工计算顺序非常复杂，并造成更新传播缓慢。近期的研究所调查的抽象不仅保证数据包不会“看到”部分更新的路径，而且还提供了处理分布式更新的通用机制。其想法是，给配置加上版本，然后仔细设计更新协议，协议确保每个数据包（或一组相关数据包）由同一个一致的版本进行处理。

*示例。*需要提供强一致性的配置更新是与传统网络的重大区别。为了证明它们不仅仅具有学术意义，请考虑下面的场景：

- **最短路径路由：** 网络最初配置为沿最短路径转发。然后运维人员决定关掉几台交换机进行维护。控制器生成一个新的全网配置，该配置沿另一组路径转发。网络应该一直提供连接，并且没有转发环路。
- **分布式访问控制：** 网络最初配置为过滤掉一组“禁止”的数据包，如果数据包不受禁止，就沿最短路径转发。由于过滤规则过大，一个转发表装不下，那么这些规则在网络中的多台交换机之间分摊。配置经过仔细构造，确保每个数据包通过相应的交换机，这些交换机包含必要的过滤规则。后来，运维人员决定重新安排规则，他保留原策略不变，但在不同的交换机上放置过滤规则。网络应该一直过滤掉禁止的数据包，而将其他数据包转发到目的地。
- **服务器负载平衡：** 网络最终配置为将传入的请求重定向到多个后端服务器副本。在某个时刻，多台服务器上线。然后控制器会生成在新的一组服务器之间平衡负载的新配置。网络应该一直将传入的流量转发到某台后端服务器，同时确保连接亲和性 —同一个连接中的所有数据包应该发送到同一台服务器。

在上述全部场景中，计算最初和最终配置比较简单，但在保留所需的不变量时在两者间转换则并不容易。尤其是，由于控制器无法原子化地更新整个网络的状态，因此通过网络的数据包势必由旧的、新的，甚至介于两者之间（混合包含新旧配置中的转发规则）的配置来处理。

除了版本控制的基本抽象外，控制器的状态更新子系统可以向应用程序开放多种一致性模型。

*更新抽象。*一致的更新抽象可以使控制器更新整个网络的转发状态，同时确保数据包不会经过处于两种状态之间的路径。抽象本身很容易描述：控制器程序指定被推入网络的状态的版本，而更新子系统保证通过网络的每一个数据包只“看到”一致的状态版本。除了版本控制的基本抽象外，控制器的状态更新子系统可以向应用程序开放多种一致性模型。

一个可能的模型是逐包一致性：即使用转发状态的单一版本处理每个数据包。[39] 也就是说，每个数据包要么用旧的全网配置进行处理，要么用新的配置，但是不能新旧混用。另一种模型是逐流一致性：即使用单一的配置版本处理每一组相关的数据包。[39] 其他扩展模型考虑带宽，并试图避免在状态转换期间形成额外的拥塞。[18,27]

*更新机制。*实现一致更新的一般机制是使用两阶段更新。顾名思义，两阶段更新分两步进行：控制器通过检测转发规则来修改新配置，使得这些规则只匹配标有对应于新版本的标记的数据包，然后在每台交换机上安装新配置；控制器更新网络外围的规则，给数据包标上新版本的标记，然后从每台交换机上卸载旧配置。虽然在状态转换期间，网络包含来自新旧配置的混合规则，但这些规则都有这样一个特性，即任何给定的数据包都将根据单一版本进行处理。类似的机制可用于实现逐流一致性。[39]

很多情况下，优化的机制可代替两阶段更新使用。例如，如果更新只是添加路径，那么只需更新影响这些路径的规则。同样，

如果更新只影响一部分交换机（并且策略具有不会将流量多次转发过这些交换机的特性），那么其他交换机就根本无需更新。这些优化的机制生成的消息更少、在交换机上占用的规则空间更少，或者比完整两阶段更新更快地完成状态转换。一致更新还可以逐步渐进地实现[21]，或者使某些数据包转移到控制器来实现。[31]

*讨论。*更新是任何 SDN 控制器的基本抽象。但是，尽管某些初步结果前景很好，但是还是有很多问题仍待解决。最明显的问题是效率：上面所述的机制需要大量规则空间和大量控制消息来实现状态转换。在大型网络中，这些机制的成本会令人难以承受。这里讨论的优化是很好的开端，但是需要更全面的研究。另一个重要问题是更新的响应能力。本节所述的抽象不保证更新需要多长时间才能完成。对于计划内变化，这可能是可接受的，但是在对故障作出反应时，快速响应非常重要。[38] 如果有一种抽象以更弱的保证换取响应更快的更新机制，那么研究这样的抽象会比较有趣。例如，有一种抽象只保证数据包最终抵达其最终目的地，并且不会走环路，这种抽象看上去很自然，并且为更有效的实现留出了空间。最终，它可能对综合处理特定于应用程序的不变量所发生的更新很有用。[28,35]

模块化组合

在操作系统中，进程允许多个用户共享同一台计算机上的可用硬件资源。除了与内存、锁、文件描述符和套接字等系统资源关联外，每个进程还与一个执行线程关联。操作系统要求进程间的所有交互通过规定明确的接口发生。例如，分配给一个进程的内存不能由另一个进程篡改，除非第一个进程显式共享该内存。虽然有人已将 SDN 控制器比作“网络操作系统”，但是当前控制器缺乏与进程相似的抽象。[13] 实际上，大多数控制器给予应用程序不受约束的访问权，使得应用程序可以访问网络中每台交换机上的转发表，这造成很难以模块化的方式编写程序。

这令人遗憾，因为网络编程应该很自然地适合于模块化。SDN 应用程序通常是用标准构建元素（如路由、广播、监视和访问控制）构造的。但是，大多数 SDN 控制器缺乏模块性，这迫使程序员在每个新应用程序中从零开始重新实现这些基础服务，而不是简单地从库中获取。

*示例。*下面的场景说明了为什么在当前 SDN 控制器中很难实现模性。

*转发和监视：*网络实现转发和流量监视。因为交换机表实现了这两种功能，所以必须小心构思规则，做到转发和监视某些数据包，但不转发和监视其他数据包。如果程序员并行执行标准的转发和监视程序，程序可能安装重叠的规则，而系统的整体行为将不可预测。

*有隔离转发：*网络分成两组主机。两组主机相互隔离，但是网络在同一组的主机对之间转发流量。与前面的例子相似，程序分解成两个正交的功能：隔离和转发。但是，程序员必须同时考虑两种功能，因为一个模块生成的规则可能很容易将流量转发到另一组的主机，从而违反预定的规则。

*低延迟视频和批量数据传输：*网络为视频会议应用程序提供低延迟服务，并且只要有足够的带宽，就允许备份应用程序沿多条不同的路径转发流量。程序员必须同时考虑这两种功能，以确保符合每个应用程序的服务级要求。

虽然这些示例涉及不同的应用程序，但是问题的原因都一样：允许程序直接操作低级网络状态，造成实际上不可能以模块化的方式开发 SDN 应用程序。

*编程语言抽象。*可使 SDN 应用程序更模块化的一种方法是，更改它们所用的编程接口。SDN 程序员可以使用编译成 OpenFlow 的高级语言，而不是显式管理交换机上的低级转发规则。这样的语言应允许程序员独立开发和测试模块，而不用担心出现意外的交互。程序员甚至可以将一个模块替换为提供相同功能的另一个模块。

NetKAT[2] 语言（及其前身 NetCore[14,32,33]）提供了一组高级编程构造，包括用于组合独立程序的运算符。在第一个示例中，可以使用其并集运算符组合转发和监视功能，这将产生所需要的既转发又监视的模块。NetKAT 编译器采用此策略，并生成可由其运行时系统安装在交换机上的等效转发规则。Maple 控制器[43] 允许程序员用 Java 或 Haskell 将模块编写为包处理函数，从而利用这些语言提供的模块化机制。Maple 使用运行时跟踪的形式记录程序决策，并创建优化的 OpenFlow 规则。

*隔离切片。*某些情况下，程序员需要确保合并的程序不会相互干扰。例如，在流量隔离场景中，两个转发模块必须互不干扰。

使用并集合并它们并不正确 — 这些模块可能相互发送数据包来进行交互。保证隔离的一种方法是使用这样的抽象：这种抽象允许程序并行执行，同时将每个程序限制在各自的网络“切片”中。FlowVisor 在控制器和交换机之间插入一个虚拟机管理程序，检视每个事件和控制消息，以确保程序及其流量限制在各自的网段内。[42] FortNOX 控制器还使用一种以基于角色的身份验证为基础的框架，提供了应用程序之间的强隔离。[37] NetKAT 的近期扩展提供了类似于切片的编程构造。[2,15]

*参与式网络。*多模块的合并行为有时会导致冲突。例如，如果一个模块保留链路上所有可用的带宽，那么其他模块将无法使用该链路。PANE 控制器[10]允许网络管理员指定特定于模块的配额，以及网络资源的访问控制策略。PANE 利用这种机制提供了一个允许终端主机应用程序请求网络资源的 API。例如，视频会议应用程序可以轻易修改为使用 PANE API 来为高质量视频通话保留带宽。PANE 确保其带宽请求不会超过管理员设置的限制，也不会造成其他应用程序缺乏资源。

*讨论。*将复杂应用程序分解成简单模块，这种抽象是 SDN 的关键技术。要是没有抽象，程序员就不得不以大而笨重的方式编写程序，同时开发、测试和推导程序的每一部分之间可能出现的交互。NetKAT 和 Maple 等高级语言提供的抽象、FlowVisor 和 FortNOX 之类的虚拟机管理程序，以及 PANE 之类的控制器，它们使得以模块化的方式构建应用程序成为可能。但是，虽然这些抽象的初步阶段很有前景，但是还有很多工作要做。例如，开发人员需要直观的推导原则，用于建立基于单独的模块构建的程序的属性 — 例如，一个模块是否可以由另一个模块所取代，而又不会影响整个程序的行为。他们还需要更好的表示方式和冲突解决方式，尤其是对于涉及安全性和资源限制的属性。

虚拟化

SDN 将控制网络的软件与底层转发元素分离。但是，它没有将转发逻辑与底层物理网络拓扑分离。这意味着实现最短路径路由的程序必须保留拓扑的完整表示，只要拓扑变化，就必须重新计算路径。为了解决这一问题，某些 SDN 控制器现在以虚拟网络元素的形式提供了用于编写应用程序的基本类型。将程序与拓扑分离还创造了使 SDN 应用程序更加可扩展、更加容错的机会。

*示例。*作为虚拟化的动机，请考虑下面的场景：

*访问控制：*访问控制通常是通过将 MAC 或 IP 地址之类的信息编码到配置中来实现的。遗憾的是，这意味着拓扑变化（比如，主机从一个位置移到另一个位置）可能破坏安全性。如果换成以连接到每台主机的虚拟交换机的形式配置访问控制列表，那么即使拓扑发生变化，策略仍保持稳定。

*多租户数据中心：*在数据中心中，人们常常希望能让多个租户对共享物理网络中的设备施加不同的策略。但是，重叠的地址和服务（以太网与 IP）导致转发表非常复杂，很难保证一个租户生成的流量会与其他租户隔离。若使用虚拟交换机，就可以给每个租户提供一个虚拟网络，他们可以随意配置这个虚拟网络，而不会干扰其他租户。

*向外扩展路由器：*在大型网络中，可能有必要让一组物理交换机看上去像一台逻辑交换机。例如，一大堆低成本的消费级交换机可以组合成一台运营商级别的路由器。除了简化各个应用程序的转发逻辑外，这种方法还可用于获得可伸缩性 — 因为这样的路由器只存在于逻辑层面，所以可以根据需要，用更多的物理交换机来动态扩大这样的路由器。

如上面的例子所示，因为转发逻辑与具体的物理拓扑分离，所以虚拟化可以使应用程序的可移植性和可伸缩性更强。

*虚拟化抽象。*SDN 的虚拟网络抽象最突出的例子是 VMware 的网络虚拟化平台 (NSX)。[7,9] Pyretic 控制器支持类似的抽象。[33] 这些控制器在虚拟层和物理层向程序员展示了同样的基础结构 — 一张表示网络拓扑的图；这使得针对物理网络编写的程序可以用在虚拟层，反之亦然。

为了定义虚拟网络，程序员指定逻辑网络中的元素与物理网络中的元素之间的映射。例如，为了根据一个任意拓扑创建一个“大交换机”，他们可以将物理网络中的所有交换机映射成一个虚拟交换机，并隐藏所有内部关系。[7,33]

*虚拟化机制。*虚拟化抽象很容易描述，但是它们的实现远没有那么简单。NSX 之类的平台所基于的控制器虚拟机管理程序将逻辑层的事件和控制消息直接映射到物理层，反之亦然。为了简化实现虚拟化所需要的簿记工作，大多数平台为传入的数据包标上一个标记（例如 VLAN 标记或 MPLS 标签），标记将数据包与一个或多个虚拟网络显式关联起来。

这些系统中的数据包处理分多步进行。首先，系统识别数据包的逻辑上下文 — 即，它在虚拟网络中的位置，由交换机和端口组成。其次，系统根据数据包逻辑上下文的策略处理数据包，这将把数据包重新放入另一个逻辑上下文（并可能生成额外的数据包）。最后，系统将数据包直接映射到物理层。虚拟机管理程序通常会生成同时实现所有三步的物理层转发规则。一个令人担忧的难题是物理交换机上可用的规则空间。根据虚拟网络的数量及其策略的大小，虚拟机管理程序可能无法容纳实现交换机上的这些策略所需要的全部规则。因此，与普通操作系统上的内存管理一样，虚拟机管理程序通常实现了某种形式的“分页”，将规则动态移入和移出物理交换机。

*讨论。*虚拟化抽象是现代 SDN 控制器的重要组成部分。将程序与物理拓扑分离简化了应用程序，还可以实现在多个不同的程序之间共享网络而没有干扰。但是，虽然多种生产用控制器已经支持虚拟化，但是还有很多问题仍待解决。有一个问题涉及在逻辑层应开放的详细

程度。当前的 SDN 虚拟化实现在逻辑层和物理层提供了相同的编程接口，忽略了链路容量、队列和本地交换机容量等资源。另一个问题是，如何将虚拟化与一致更新等其他抽象相结合。直接做这样的结合并非总是可行，因为这两种抽象普遍是用标记方案实现的。最后，当前平台不支持高效的嵌套虚拟化。语义上，没有很深刻的问题，但是使用虚拟机管理程序实现嵌套虚拟化存在实际后果。

将程序与物理拓扑分离简化了应用程序，还可以实现在多个不同的程序之间共享网络而没有干扰。

形式验证

今天的网络运维人员通常手工处理低级网络配置。不出意外，这会导致配置错误，造成很多网络不可靠、不安全。通过标准化网络硬件的接口，SDN 带来了巨大的机会，使人们可以开发出相应的方法和工具，让构建和运维可靠的网络变得更加容易。网络中会出现很多关键的不变量，下面将描述其中几个。使用形式建模网络和控制器状态的静态或动态工具，可以自动检查这些属性。

示例。很多网络属性都依拓扑而定，因此只有给定网络结构的模型，才可以声明和验证这些属性。

连通性：网络中任何主机发出的数据包最终会被递送到它们要到达的目的地，除非可能是因为拥塞或故障。

无环路：没有数据包会沿着环路转发回曾经处理过这个相同数据包首部和内容的结点。

中间站：不受信任的主机发出的数据包会经过一个中间盒，中间盒扫描恶意流量后，数据包才能转发到其所要的目的地。

带宽：网络提供在与租户间达成的服务级别协议中规定的最低带宽。

其他属性要么完全拓扑无关，要么适用于大的拓扑类别。这些属性体现了要在多种不同的网络上执行的应用程序的一般正确性准则。

访问控制：网络根据访问控制列表中指定的主机，阻止未经授权的主机发出的所有流量。

主机学习：控制器最终了解所有主机的位置，网络将数据包直接转发到其所要的目的地。

生成树：网络沿着包含每个交换机的树转发广播流量（如果网络是连通的）。

在传统网络中一直很难确定这两类属性，因为它们需要推导分散在很多异构设备上的复杂状态。近期多种建立在 SDN 提供的统一接口基础上的工具使得自动验证多种网络属性成为可能。

验证配置。验证无环路和连通性等属性需要对拓扑和交换机配置建模。标头空间分析 [23] 将交换机和拓扑建模为 n 维空间中的函数，空间中的点表示数据包首部的向量。此模型可用于生成覆盖整个配置中每条规则的测试数据包 [46]，扩展可以增量检查配置。[22] FlowChecker 基于类似的理念，但是将策略编码为二进制决策图。[1] Anteater[29] 将交换机配置编码为布尔 SAT 实例（在最初由 Xie 等人开发的编码基础上建立）。[45] VeriFlow[24] 开发了特定于域的表示和算法，用于实时检查属性，这一点很重要，因为 SDN 的转发行为可能会变化很快，尤其是在控制器要对不断变化的网络状况作出反应的时候。最后，NetKAT[2] 包含一套可靠、完整且可决定的等式推导系统，用于证明网络程序间的等效性。

验证控制器。除了可验证配置属性的工具外，近期的一些研究侧重于可验证控制程序本身的工具，通常侧重于拓扑无关的属性。NICE[5] 使用符号执行和模型检查的组合来验证多种重要的属性，包括是否存在竞争条件以及类似于交换机内存泄漏的缺陷。Scott 等人开发的另一个工具检查 SDN 控制器提供的抽象在交换机配置中是否正确

实现。[17] Guha 等人描述了使用证明助手确定控制器正确性的框架，以及针对 OpenFlow 的详细操作模型的 NetCore 语言机器验证实现。[14] VeriCon 显示，Hoare 式验证对于编写成简单命令式程序的控制器是可行的[3]，并已成功应用于改编自 SDN 文献的很多示例（例如，防火墙、路由算法等）。Nelson 等人提出了基于 Datalog 的 SDN 编程语言，名为 Flowlog，他们还使用该语言编写和验证了多种标准属性。[34] 由于 Flowlog 被设计成有限状态，因此很适合无需程序员提供的复杂断言的自动验证。

*讨论。*对于可严格保证联网系统的运行状况、性能、可靠性和安全性的工具，人们有着巨大的需求。SDN 标准化了控制网络的接口，因此人们可以构建相应的工具来对照精确的形式模型验证配置和控制器。这一领域可能出现的后续研究包括，开发自定义逻辑和决策过程来表示和检查属性、以更多特性（如延迟和带宽）丰富模型，以及将属性检查和调试工具更好地集成到 SDN 控制器平台。

相关研究

近年来，SDN 发展势头迅猛，但是 SDN 背后的理念则是建立在很多以前工作的基础之上。Tempest，[40] 一种在上世纪九十年代中期在剑桥开发的体系结构，是在 ATM 网络的背景下分离转发和控制的早期尝试。Tempest 的很多特点如今都可在 SDN 中找到，包括强调开放接口以及对虚拟化的支持。同样，IETF ForCES 工作组定义了控制器可用来管理单个网络中的多种异构设备的标准协议。[8] Soft-Router 项目从可扩展性、可伸缩性、可靠性、安全性和成本的角度研究了分离转发和控制的好处。[26]

AT&T 开发的路由控制平台[4]证明，逻辑集中化可用于大大简化路由算法，同时仍提供很好的性能。这些理念以后在 4D 平台[12]中得到扩展，该平台引进了管理面和控制面的差别。在这项研究中还可以看到，在 SDN 中使用全网数据结构表达算法，而不是使用分布式算法的好处。

距离 SDN 最近的前身是 Ethane[6]，一种旨在提供细粒度网内访问控制的系统。Ethane 提供了用于定义安全策略的高级语言以及一个控制器程序，该程序在可编程网络交换机中安装和卸载自定义转发规则来实现这些策略。NOX 控制器基于 Ethane[13]，而 Ethane 控制器用来与交换机通信的协议以后发展成 OpenFlow 标准第一版。[31]

对于可严格保证联网系统的运行状况、性能、可靠性和安全性的工具，人们有着巨大的需求。

总结

围绕 SDN 的很多初步工作都着重于体系结构问题 — 这使得发展网络和开发丰富应用程序成为可能。但是，这种新的软件生态系统的发展也带来了基本的新抽象的发展，这些新抽象充分利用以约束更少的状态分布模型在标准服务器上编写网络控制软件的能力。我们相信这些抽象对于实现 SDN 的目标至关重要，并可能证明是 SDN 传承最久远的传统。

鸣谢 本文作者希望感谢 Shrutarshi Basu、Andrew Ferguson、Anil Madhavapeddy、Mark Reitblatt、Jennifer Rexford、Mooly Sagiv、Steffen Smolka、Robert Soulé、David Walker 以及 *Communications* 的审阅者，感谢他们有益的评论。我们的研究得到了 NSF 资助项目 CNS-1111698、ONR 奖学金 N00014-12-1-0757、Sloan Research Fellowship 和 Google Research Award 的支持。

参考资料

1. Al-Shaer, E. and Al-Haj, S. FlowChecker:Configuration analysis and verification of federated OpenFlow infrastructures.In *Proceedings of SafeConfig*, 2010.
2. Anderson, C.J., Foster, N. Guha, A., Jeannin, J-B, Kozen, D., Schlesinger, C. and Walker, D. NetKAT:Semantic foundations for networks.In *Proceedings of POPL*, 2014.
3. Ball, T., Bjørner, N., Gember, A., Itzhaky, S., Karbyshev, A., Sagiv, M., Schapira, M. and Valadarsky, A. VeriCon:Towards verifying controller programs in software-defined networks.In *Proceedings of PLDI*, 2014.
4. Caesar, M., Caldwell, D.F., Feamster, N., Rexford, J., Shaikh, A.and van der Merwe, J.E.Design and implementation of a routing control platform.In *Proceedings of NSDI*, 2005.
5. Canini, M., Venzano, D., Perešíni, P., Kostic´, D. and Rexford, J. A NICE way to test OpenFlow applications. In *Proceedings of NDSI*, 2012.
6. Casado, M., Freedman, M.J.Pettit, J., Luo, J., McKeown, N. and Shenker, S. Ethane:Taking control of the enterprise.In *Proceedings of SIGCOMM*, 2007.
7. Casado, M., Koponen, T., Ramanathan, R. and Shenker, S. Virtualizing the network forwarding plane.In *Proceedings of PRESTO*, 2010.
8. Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W. Dong, L., Gopal, R. and Halpern, J. Forwarding and control element separation (ForCES), 2010.IETF RFC 5810.
9. Koponen, T. et al.Network virtualization in multi-tenant datacenters.In *Proceedings of NSDI*, 2014.
10. Ferguson, A.D., Guha, A., Liang, C., Fonseca, R. and Krishnamurthi, S. Participatory Networking:An API for application control of SDNs.In *Proceedings of SIGCOMM*, 2013.
11. Fortz, B., Rexford, J. and Thorup, M. Traffic engineering with traditional IP routing protocols.*IEEE Commun.* (Oct. 2002).
12. Greenberg, A.G., Hjálmtýsson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G.G., Yan, H., Zhan, J. and Zhang, H. A clean slate 4D approach to network control and management.*SIGCOMM CCR 35*, 5 (2005).
13. Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N. and Shenker, S. NOX:Towards an operating system for networks.*ACM SIGCOMM CCR 38*, 3 (2008).
14. Guha, A., Reitblatt, M. and Foster, N. Machine-verified network controllers.In *Proceedings of PLDI*, 2013.
15. Gutz, S., Story, A., Schlesinger, C. and Foster, N. Splendid isolation:A slice abstraction for software-defined networks.In *Proceedings of HotSDN*, 2012.
16. Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S. and McKeown, N. ElasticTree:Saving energy in datacenter networks.In *Proceedings of NSDI*, 2010.
17. Heller, B. et al.Leveraging SDN layering to systematically troubleshoot networks.In *Proceedings of HotSDN*, 2013.
18. Hong, C-Y, Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M. and Wattenhofer, R. Achieving high utilization with software-driven WAN.In *Proceedings of SIGCOMM*, 2013.
19. Jain, S. et al.B4:Experience with a globally deployed software defined WAN.In *Proceedings of SIGCOMM*, 2013.
20. Jose, L., Yu, M. and Rexford, J. Online measurement of large traffic aggregates on commodity switches.In *Proceedings of HotICE*, 2011.
21. Katta, N.P., Rexford, J., and Walker, D. Incremental consistent updates.In *Proceedings of HotSDN*, 2013.
22. Kazemian, P., Chang, M., Zeng, H., Varghese, G., McKeown, N. and Whyte, S. Real-time network policy checking using Header Space Analysis.In *Proceedings of NSDI*, 2013.
23. Kazemian, P., Varghese, G. and McKeown, N. Header space analysis:Static checking for networks.In *Proceedings of NSDI*, 2012.
24. Khurshid, A., Zhou, W., Caesar, M. and Godfrey, B. VeriFlow:Verifying network-wide invariants in real time. In *Proceedings of NSDI*, 2013.
25. Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T. and Shenker, S. Onix:A distributed control platform for large-scale production networks. In *Proceedings of OSDI*, 2010.
26. Lakshman, T.V., Nandagopal, T., Ramjee, R., Sabnani, K. and Woo, T. The SoftRouter architecture.In *Proceedings of HotNets*, 2004.
27. Liu, H.H., Wu, X., Zhang, M., Yuan, L., Wattenhofer, R. and Maltz, D. zUpdate:Updating datacenter networks with zero loss.In *Proceedings of SIGCOMM*, 2013.
28. Mahajan, R. and Wattenhofer, R. On consistent updates in software-defined networks.In *Proceedings of HotNets*, 2013.
29. Mai, H., Khurshid, A., Agarwal, R., Caesar, M., Godfrey, B. and King, S.T.Debugging the data plane with Anteater. In *Proceedings of SICOMM*, 2011.
30. McGeer, R. A safe, efficient update protocol for OpenFlow networks.In *Proceedings of HotSDN*, 2012.
31. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J. OpenFlow:Enabling innovation in campus networks. *SIGCOMM CCR 38*, 2 (2008).
32. Monsanto, C., Foster, N., Harrison, R. and Walker, D. A compiler and run-time system for network programming languages.In *Proceedings of POPL*, 2012.
33. Monsanto, C., Reich, J., Foster, N., Rexford, J. and Walker, D. Composing software defined networks.In *Proceedings of NSDI*, 2013.
34. Nelson, T., Ferguson, A., Scheer, M., and Krishnamurthi, S. Tierless programming and reasoning for software-defined networks.In *Proceedings of NSDI*, 2014.
35. Noyes, A., Warszawski, T., Cerny, P. and Foster, N. Toward synthesis of network updates.In *Proceedings of SYNT*, 2013.
36. Perlman, R. An algorithm for distributed computation of a spanning tree in an extended LAN.*SIGCOMM CCR 15*, 4 (1985).
37. Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M. and Gu, G. A security enforcement kernel for OpenFlow networks.In *Proceedings of HotSDN*, 2012.
38. Reitblatt, M., Canini, M., Foster, N. and Guha, A. Fattire:Declarative fault-tolerance for software-defined networks.In *Proceedings of HotSDN*, 2013.
39. Reitblatt, M., Foster, N., Rexford, J., Schlesinger, C. and Walker, D. Abstractions for network update.In *Proceedings of SIGCOMM*, 2012.
40. Rooney, S., van der Merwe, J.E., Crosby, S.A. and Leslie, I.M.The Tempest:A framework for safe, resource assured, programmable networks.*IEEE Commun.36*, 10 (1998).
41. Shenker, S., Casado, M., Koponen, T. and McKeown, N. The future of networking and the past of protocols. Invited talk at Open Networking Summit, Oct. 2011.
42. Sherwood, R. et al.Carving research slices out of your production networks with OpenFlow.*SIGCOMM CCR 40*, 1 (2010).
43. Voellmy, A., Wang, J., Yang, Y.R., Ford, B. and Hudak, P. Maple:Simplifying SDN programming using algorithmic policies.In *Proceedings of SIGCOMM*, 2013.
44. Wang, R., Butnariu, D. and Rexford, J. OpenFlow-based server load balancing gone wild.In *Proceedings of HotICE*, 2011.
45. Xie, G.G., Zhan, J., Maltz, D.A., Zhang, H., Greenberg, A.G., Hjálmtýsson, G. and Rexford, J. On static reachability analysis of IP networks.In *Proceedings of INFOCOM*, 2005.
46. Zeng, H., Kazemian, P., Varghese, G. and McKeown, N. Automatic test packet generation.In *Proceedings of CoNext*, 2012.

Martin Casado (mcasado@vmware.com) 是 VMware 的研究员、高级副总裁以及网络和安全的总经理，也是加州帕洛阿尔托 Nicira Networks 的联合创始人和 CTO。

Nate Foster (jnfoster@cs.cornell.edu) 是纽约州伊萨卡康奈尔大学的计算机科学助理教授。

Ajun Guha (arjun@cs.umass.edu) 是马塞诸塞州阿姆赫斯特麻省大学的计算机科学助理教授。

译文责任编辑：陈贵海

DOI:10.1145/2661432

如需查阅相关论文，请
访问 /10.1145/2661434

技术视角
从中间部位攻克问题

作者：Bart Preneel

下文展示了一种用于解决一大类组合优化问题的全新而简洁的算法。在研究 DES 算法的加强版本时，我们萌生了剖析方法的想法。DES 是由美国国家标准局（现为国家标准与技术研究所，简称 NIST）于 1977 年提出的密码算法，用于保护非保密的敏感美国政府数据。近三十年来，DES 已成为全球行的加密标准。IBM 设计了 DES，但美国国家安全局 (NSA) 决定将密钥长度限定在 56 位。这是一个争议性举动，因为 NSA（唯一）可借此通过搜索密钥空间来破解密码。

看上去增加密钥长度的直接方法是进行二次加密，即“Double-DES”或 $C = \mathrm{DES}_{K1}(\mathrm{DES}_{K2}(P))$，其中 P 和 C 分别是明文和密文，而 $K1$ 和 $K2$ 是两个不同的 k- 位密钥。Diffie 和 Hellman 在 1977 年证实了这种方法没有效果。让我们假定一些已知的密文和相应的明文（这是密码学的标准假定，在实践中经常遇到）。中间会合攻击的工作方式如下：首先，尝试首个密钥 $K1$ 的所有 2^k 值，计算 $A = \mathrm{DES}_{K1}(P)$ 的中间值，然后将配对 $(K1, A(K1))$ 存储在表格中。第二步，计算密码 $K2$ 的所有中间值，作为 $A' = \mathrm{DES}^{-1}_{K2}(C)$，并在表格中搜索值 A。如果找到匹配情况，那么候选配对 $(K1, K2)$ 将对照额外明文和密文进行检查。该算法需要约 2^k 加密和 2^k 内存。因此，金融行业已将 Single-DES 升级至使用三次加密的 Triple-DES。针对 Triple-DES 的中间会合攻击需要约 2^{2k} 加密和 2^k 内存。

本文作者探讨了这样一个自然的问题：使用 Quadruple-DES 是否会带来更好的安全性？初看之下，人们可能得出以下结论：破解 Quadruple-DES 需要 2^{2k} 加密和 2^{2k} 内存。但实际情况令人吃惊，通过应用递归可令破解 Quadruple-DES 的成本与破解 Triple-DES 的成本相当，方法为先猜测首个中间值，然后执行两个中间会合攻击。该想法由 Zhu 和 Gong 提出，用于分组加密算法 KATAN，作者的灵感源自 Isobe 关于分组加密算法 GOST 工作。然而，作者在 Crypto 2012 论文中进一步发展了这些想法，他们对用于任何 n 值的 n- 重加密案例的算法进行了优化和推广。而且，他们展示了剖析算法拥有多种用途。

本文作者探讨了这样一个自然的问题：使用 Quadruple-DES 是否会带来更好的安全性？

本文介绍了剖析算法的两种应用。第一个是魔方的解决方案。剖析解决方案与早期算法的复杂性相同，但无需使用群论性质。第二个是组合划分问题，该问题的一种变体（背包问题）在公钥加密和哈希函数的密码学构建中扮演重要角色。在这个例子中，更加复杂的策略被运用，以便将问题划分为不均匀的部分。

该剖析技术非常出色，源自一个简单的想法，即中间会合攻击。作者展示了该想法可以用一种简洁的方式加以推广，很多巧妙的优化都可以让想法得到最大限度利用。本文的主要贡献是作者意识到剖析算法可以广泛地用于组合优化问题。最后，作者提出的算法可降低内存需求，这从根本上确保了该算法的实用性。 C

Bart Preneel (bart.preneel@esat.kuleuven.be) 是 Katholieke 大学电气工程系 COSIC 研究组的教授。该大学位于比利时勒芬。

译文责任编辑：孙晓明

DOI:10.1145/2661434

剖析：解决双复合搜索问题的全新范式

作者：Itai Dinur、Orr Dunkelman、Nathan Keller 以及 Adi Shamir

摘要

组合搜索问题通常采用如下方式加以描述：一组可能的状态集合；一个动作列表，这些动作可能将各个当前状态映射到一些下一状态；以及一对初始和最终状态。算法问题在于找到可以将给定初始状态映射到期望最终状态的一组动作序列。我们在本文中介绍了双复合搜索问题的全新想法，并展示了如何通过使用一种叫做剖析算法的全新算法范式，来改善时间和空间复杂性以解决问题。为了证明我们的新范式具有广泛的适用性，我们展示了如何使用该方法来破解魔方，并以优于任何已知的相同用途的算法的算法来解决典型的 NP- 完全划分问题。

1. 引言

在设计高效算法时，一个核心问题是如何解决搜索问题：我们获得一对状态和一组可能的动作的集合，我们需要找出如何通过执行一些动作序列来完成从第一个状态到第二个状态的转换。在某些情况下，我们只希望确定是否存在这样的序列；而在其他情况下，这类序列显然是存在的，但我们需要找出最短的可行序列。这类搜索问题大多涉及到 NP 完全的判定问题，因此我们不期望找到任何多项式时间算法以解决所有实例。然而，我们希望找出一种全新指数时间算法，其指数应小于已知最好算法的指数。比如，我们无法利用穷尽密钥搜索算法在有效时间内破解秘钥为 $n = 100$ 未知位的加密方案，因为 2^n 的时间复杂性会令当前最大的数据中心难以应对。然而，如果我们能够找到运行时间为 $2^{n/2}$ 的更好的密码分析攻击，我们就能以适度的代价破解该方案，尽管该复杂度函数是指数量级的。在这类情形中，一个通常有用的方法是寻找攻击在时间与空间复杂性上的折衷方案：穷尽搜索需要花费大量时间，但消耗的内存却微不足道，因此，折衷方案十分有益，它会使用更多内存（以大型表格的预计算值的形式）来降低所需的时间（略过很多计算步骤）。对于那些在本文的拓展部分（参见 Dinur 等人的著述 [2]）阐述的原因，我们通常考虑算法所需时间和空间的乘积，因为我们尝试将相应复杂性评估降至最低。在上述案例中，以 $T = 2^n$ 的时间和 $S = 1$ 的空间来破解密码系统是不可行的，而以 $T = 2^{2n/3}$ 和 $S = 2^{n/3}$ 空间（其乘积 $TS = 2^n$ 和之前的结果相同）较上述方法略好，但几乎还是不可行的，以 $T = 2^{n/2}$ 时间和 $S = 2^{n/4}$ 空间（其乘积 $TS = 2^{3n/4}$ 的指数更小）是完全可行的。

一个典型的搜索问题是由条件 F（如，形式为子句合取的 CNF 布尔公式，）和候选解决方案 X（如，形式为针对所有 F 中的变量的 0/1 赋值）所定义的，目标是在所有可能的 X 中至少找到一个 X 可以满足条件 $F(X)$ 是正确的。这一表达式在某种意义上缺少内部结构，它使用 F 的完整描述和 X 的完整值来确定 $F(X)$ 是否得到满足。然而，我们通常能以更小决策序列来代替全有全无选择 X。比如，我们可以从所有变量的 0 赋值入手，我们可以在每个阶段决定翻转一个布尔变量中的当前值。在该过程中的任意中间点，F 均处于以下状态：部分子句得到满足，部分子句未得到满足。而我们的目标是找到可使所有子句同时得到满足的状态。一般而言，如果某搜索问题的解决方法可以利用一组基本动作序列来描述，且该动作序列可以将系统从某些初始状态经由一系列中间状态来达到最终期望状态，那么我们就称之为复合搜索问题。大多数搜索问题都含有此类复合结构，它可以采用图 1 中的*执行矩阵*来反映。该矩阵的行代表状态 S_0、S_1、……，图左侧的一组动作序列 a_1、a_2……则代表解决方案 X，其中的动作 a_i 实现了状态 S_{i-1} 到状态 S_i 的转换。在众多情况下，基本动作是针对状态的可逆操作，这样一来才有可能使用动作 a_i 将 S_{i-1} 映射回 S_i，a_i^{-1} 将 S_i 映射回 S_{i-1}。比如，在布尔公式的情况下，我们都可以翻转 X 中的任意变量的当前值，但是我们可以通过再次翻转相同变量来取消该效果，因此在该情况中，该动作及其反动作刚好是相同的。在本文中，我们仅考虑了

本文的先前版本曾发表于 *CRYPTO*, vol. 7417 (2012) *Lecture Notes in Computer Science*Springer, 719–740.

此类可逆转情况。在这类情况中，我们可以通过以下方式来分解搜索问题：对初始状态应用一些前进动作，对最终状态应用一些逆转动作，搜索可合并这些部分的中间会合 (MITM) 状态。我们的全新剖析方法甚至可以应用于一些涉及不可逆转动作的情况，但这会使动作的描述更加复杂化，也会略微增加动作的复杂性，这些影响已在 Dinur 等人 [2] 的论文中有所论述。

迄今为止，我们通过将解决方案流程划分为一系列基本动作来将执行矩阵划分成多个行。为了应用我们的最新剖析方法，我们还必须将执行矩阵划分成多个列，这可以通过将每个 S_i 状态划分为更小的区块 $S_{i,j}$ 来完成，我们将这些区块成为次状态，如图 2 所示。然而，只有子状态可以相互独立地操作的划分才对我们有用。我们称一个搜索问题带有*双复合结构*，如果该搜索问题可以被执行矩阵描述，且该执行矩阵中的动作 a_i 的知识使得子状态 $S_{i,j}$ 可以由 $S_{i-1,j}$，子状态 $S_{i-1,j}$ 可以由 $S_{i,j}$ 唯一的确定，即便是我们不了解矩阵中的其他子状态。这就意味着，如果我们选择执行矩阵内的任何尺寸的矩形，沿着上边缘的子状态 $S_{i-1,j}$、$S_{i-1,j+1}$、……$S_{i-1,k}$ 的知识和左侧的动作 a_i、a_{i+1}、……a_ℓ 的知识足以计算次沿着下边缘的次状态 $S_{\ell,j}$、$S_{\ell,j+1}$、……、$S_{\ell,k}$，反之亦然。

图 1.复合搜索问题的执行矩阵。

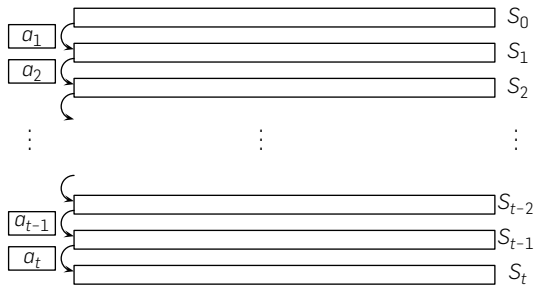
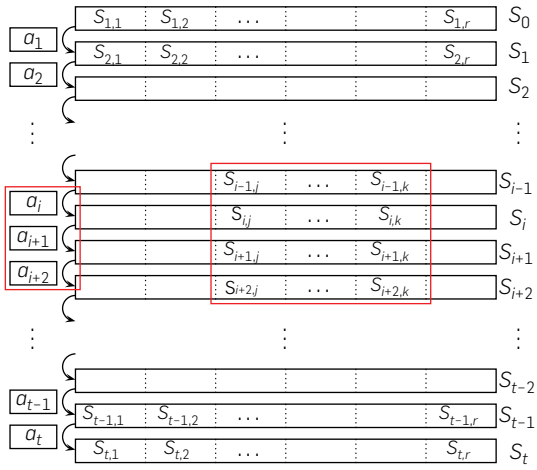


图 2.双复合搜索问题的执行矩阵。



并不是每个搜索问题都拥有此类双复合表达式，但我们在本文中展示了很多著名问题都能以这种方式表达。当某问题为双复合问题时，我么可以通过使用全新通用方法来提高解决效率。我们的主要发现为，在这种情况下，我们可以通过考虑仅在一个部分赋值的中间状态上去部分匹配正反两个方向的算法来改善标准的 MITM 算法 (标准 MITM 算法试图在一个完全的中间状态匹配正反两个方向)。此外，我们可以逆转 MITM 算法的逻辑。我们不必尝试从执行的两端向某恰好相同的中间状态聚合，相反，我们的入手点可以是以一切可行方式对该中间状态进行部分猜测，对于每种猜测值，我们都可以将问题分解为两个独立的子问题，方法为从该中间状态向两端推进（基于我们可以在双复合问题中确定较长动作序列对部分指定状态的效果)。我们可以部分猜测执行矩阵中的额外子状态，从而以递归方式解决任何子问题。我们将这种方法称为*剖析*算法，因为它代表的流程正如外科医生为实施手术而在患者身体各个特定部位上割开一系列小切口。比如，在第 4 节，我们展示了如何解决著名的组合划分问题：首先猜测执行 3/7 的动作后可能的状态的其中 2/7 的位。然后猜测执行 5/7 的动作后可能额外发生的 1/7 的位状态，并以此解决产生的次问题之一。

我们撰写本文的主要目的在于介绍一种全新理念，我们展示了如何以尽可能简单的方式来运用其解决多个著名搜索问题。我们有意识地忽略了几个棘手的问题，它们的正确处理方法难以阐述，我们也省略了几个可能的优化方法，这些方法可以进一步降低我们的算法的复杂性。在分析算法的运行时间时，我们为做到清晰明确，往往会假定我们需要解决的案例是随机选取的，我们尝试猜测的中间状态是均匀分布的。

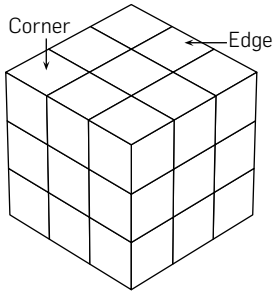
本文的组织结构如下。在第 2 节，我们描述了如何以尽可能少的步骤来将魔方作为双复合搜索问题予以解决。在第 3 节，我们展示了解决问题的两个方法：以近似于搜索空间大小的平方根的时间复杂性来解决；使用新剖析算法的最简单版本，以近似于搜索空间大小的四次方根的时间复杂性来解决。在第 4 节，我们描述了基本剖析方法的若干改进之处，并展示了如何利用其解决时间和空间复杂性相结合的被称为“组合划分”的另一种搜索问题。这类问题无法被已发表的任何算法解决，更适用于大规模的基于 FPGA 的硬件。第 5 节是本文结尾部分，我们提出了一些见解。

2. 将魔方表达为双复合搜索问题

在本节中，我们展示了如何通过构建双复合表达式来解决 3 x 3 x 3 标准魔方这一著名问题。[6] 我们可以假定，我们始终以固定方向来持有该魔方，白色中心颜色朝上，黄色中心颜色朝左，等等。在这 27 个子魔方块中，有 1 个位于魔方中央位置，它完全隐藏在其中，因此我们可以将其忽略。位于各表面中间部位的 6 个子魔方块不会在我们扭转任何一表面时发生移动，因此我们在状态表达式中也可以忽略它们。我们所采取的行动是将六个表面均扭转 90°、180° 或 270°（我们不得扭转中间的切面，因为这会改变以上定义的魔方的标准方向）。因此，我们拥有一个包含 18 个基本动作的指令集，它们可以应用于魔方的每个状态。请注意，在某种意义上，这些动作都是针对魔方状态的可逆转映射，90° 扭转的逆转效果相同于将相同表面扭转 270°，这两个动作都是可用的基本动作。

在 27-6-1=20 个子魔方块中，有 12 个带有两种可见颜色并且可以移动，我们称之为边缘子魔方块，8 个带有三种可见颜色的称为边角子魔方块（图 3）。每个此类子魔方块都可以用表面上的颜色组合来分别描述，比如蓝 - 白 (BW) 边缘子魔方块或绿 - 橙 - 红 (GOR) 边角子魔方块。此外，魔方的每个位置都可以用相应的表面来描述，如上 / 下、左 / 右、前 / 后的组合。因此，我们能以长度为 20 的向量来描述魔方的任何状态，第 i 个元素描述了第 i 个子魔方块的当前位置（如，向量上的第一个元素始终代表蓝 - 绿边缘子魔方块，并表明其目前位于上前位置）。为了完成该规范，我们还需要选择一些标准的颜色方向，请注意，边缘子魔方块可以为标准状态（如 BW），也可以为逆转状态（如 WB），每个边角子魔方块都可以是三种可能的方向之一（如 GOR、ORG 或 RGO）。还需要注意，任何可能的动作都只能将边缘子魔方块与边缘子魔方块相互移动，或者将边角子魔方块与边角子魔方块相互移动。如果我们使用状态向量中的前 12 个位置来描述 12 个边缘子魔方块（处于某种固定顺序）的当前位置，而每个位置上的元素都可以用 1 到 24 之间的数字来描述（规定其目前处于 12 个可能的位置中的一个以及 2 个可能的方向中的一个）。同样，当我们使用状态向量中的后 8 个位置来描述 8 个边角子魔方块（处于某种固定顺序）的当前位置时，每个元素仍然包含着一个介于 1 到 24 之间的数字，这次需要规定的是其所处于的 8 个可能的位置中的一个以及 3 个可能的方向中的一个。因此，我们可以使用长度为 20、每个元素介于 1 到 24 之间数字的向量来描述魔方的任何状态。18 个基本动作中的任何一个动作都可以改变该向量上的 8 个元素，方式为将 4 个边缘子魔方块和 4 个边角子魔方块移动到新的位置和方向，其余 12 个元素则保持不变。

图 3.魔方。



现在，魔方解决问题可以表现为以下搜索问题：我们得到第一个长度为 20 的向量（代表魔方的初始被打乱的状态）和第二个长度为 20 的向量（代表魔方的标准未打乱状态）。我们想找到一组基本动作序列（扭转表面的方式）来将第一个向量转变为第二个向量。找到一些序列比较容易，趣味数学著述中描述的很多算法都可以达到该目的（可参见 Slocum 的著述 [6]）。然而，这些算法一般非常浪费，需要使用 50 到 100 个动作，才能将子魔方块依次移动到处于固定顺序的正确位置，并且忽略了这些动作对其他子魔方块造成的影响。一些算法使用较少动作，但更难以描述，因为它们需要对整个状态进行具体的案例分析，然后才能选择首个动作。最近，Davidson 等人在其著述中 [1] 证明一项结论，如果我们希望解决*任意*处于可解状态的魔方，那么有 20 个动作是充要的。但这仅是一个存在性结果，而且作者并未展示找到这种序列的有效方式。请注意，不同的 20 个基本动作的序列有 $18^{20} = 12748236216396078174437376 \approx 2^{83}$ 种序列，但是我们可以将搜索空间的大小略微缩减至 $8 \times 15^{19} = 399030807609558105468750 \approx 2^{78}$，如果我们意识到连着扭转相同表面两次是无意义的。然而，经缩减后的大小仍然无法在可行时间内被穷尽搜索。

为展示我们的搜索问题表达式为双复合表达式，我们假定我们了解某特定子魔方块的当前位置和方向（即我们知道执行矩阵中的 $S_{i-1,j}$ 值为介于 1 到 24 的数字），而且我们应用了一些已知表面扭转动作。我们随后可以唯一确定该特定子魔方块的新位置和方向（即 $S_{i,j}$ 的值），即便我们根本不知道任何其他子魔方块（即执行矩阵中的一切其他 $S_{k,\ell}$ 值）的当前位置和方向。请注意，魔方状态的很多其他自然表达式并不含有此类双复合结构。比如，如果我们将状态向量上的第一个元素与特定的魔方位置（如上 - 前）相联系，并使用它来指示目前位于其中的边缘子魔方块（如 BW）

及其方向，那么关于第一个状态的该元素的知识并不能告诉我们，如果我们将上表面扭转 90° ，哪个边缘子魔方块（如 GR）会在上 - 前位置取代它。这类表达式要求了解执行矩阵中的其他列的知识，这取决于哪个动作应用于该状态，因此我们无法在新的剖析方法中运用该表达式。

如第 3 节所示，我们可以利用双复合表达式来为任何给定的魔方初始状态来寻找一组最多由 20 个表面扭转动作组成的序列，方法为使用完全可行的时间复杂性和空间复杂性组合，时间复杂性为搜索空间大小的平方根（即大约 2^{39} 个步骤或标准的个人电脑上的数分钟时间），空间复杂性约为该数字的四次方根（即大约 $2^{19.5}$ 内存位置或数百万字节）。结果算法完全通用，不必使用除双复合问题以外的其他问题的具体信息，并且复杂性与先前已知最佳魔方解决方法（于 25 年前设计，参见 Fiat 等人 [3] 著述）相当，后者专属性极强，并取决于由魔方定义的一套排列的群论性质。

3. 基本的剖析方法

我们现在假定，我们得到魔方的初始状态向量 S_0 和最终状态向量 S_ℓ，我们的目标是找到一系列基本动作 a_1、a_2、……a_ℓ 来将初始状态转化至最终状态。如前一节所述，我们知道 $\ell = 20$ 足以找到一种解决方案，因此我们的目标是找到 a_1、a_2、……a_{20}。

我们的剖析算法是在传统 MITM 算法基础上的扩展，后者由 Horowitz 和 Sahni [5] 在 1974 年提出，旨在解决背包问题。我们可以将 MITM 算法应用于几乎任何含有可逆动作的复合问题。当搜索空间的大小为 2^n 时，MITM 算法需要 $2^{n/2}$ 时间和 $2^{n/2}$ 空间。为确保完整性，我们在下文描述了如何将该算法应用到魔方环境，其搜索空间拥有约 2^{78} 个状态。在这种情况下，时间复杂性 $2^{78/2} = 2^{39}$ 是可行的，但空间复杂性 $2^{78/2} = 2^{39}$ 的随机访问内存位置太过高昂。

该算法的完整信息可参见图 4。该算法的第一步是对所有可能的 $20/2 = 10$ 动作序列 a_1、……a_{10} 进行迭代。我们拥有 $18 \times 15^9 \approx 2^{39}$ 个此类序列，我们将每个序列的动作都应用于 S_0，并获得 S_{10} 的可能值。我们将 a_1、……a_{10} 的 2^{39} 个值存储在相应的 S_{10} 值的旁边的列表中，并根据 S_{10} 来对该列表进行排序 [a]。随后，我们对所有可能的动作向量 a_{11}、……a_{20} 进行迭代。此类动作向量大约有 2^{39} 个，我们将每个向量对应的逆转动作应用于 S_{20}，并获得了 S_{10} 的可能值。现在，我们从经排序的列表中寻找 S_{10} 值，对于每个匹配情况，我们都从列表中获得相应的 a_1 值、……a_{10} 值，并输出 a_1、a_2、……a_{20} 作为解决方案。

[a] 为简单起见，我们在复杂性分析中忽略了对数因素，因此我们假定排序是线性时间操作。

图 4.解决魔方的中间会合程序。

MITM- 魔方算法

```
Input:Initial state S_0 and final state S_20
for all a_1, a_2, ..., a_10 do
   Compute S_10 = a_10(...(a_2(a_1(S_0)))...)
   Store (S_10, a_1, a_2, ..., a_10) in a list L
Sort L according to the value of S_10 in each entry (under
some lexicographical order)
for all a_11, a_12, ..., a_20 do
   Compute S_10 = a^-1_11 (...(a^-1_19(a^-1_20(S_20)))...)
   Search for S_10 in L
   if S_10 is found then
      return the associated a_1, a_2, ..., a_10 and a_11, a_12, ..., a_20
      as a solution
```

MITM 算法需要约 2^{39} 内存单元以存储经排序的列表，其时间复杂性约为 2^{39}，这是用于对每个动作向量 a_1、……a_{10} 和 a_{11}、……a_{20} 进行迭代所需要的时间。

3.1. 使用剖析来改进 MITM 算法

在本节中，我们展示了如何使用全新的剖析方法的基本版本来改进关于魔方的经典 MITM 算法。这里的主要想法是通过对中间状态 S_{10} 某些部分的所有可能值进行迭代来“剖析”中间的执行矩阵。我们会对需要进行迭代的部分 S_{10} 的大小作出选择，确保其包含约 $2^{n/4} = 2^{19.5}$ 个部分状态。由于 S_{10} 表达为长度 20 的向量，每个元素都能获得 24 个值，我们选择对前 4 个元素进行迭代，以假定约 $24^4 \approx 2^{18.5}$ 个值。对于每个此类 S_{10} 部分值，我们使用该问题的双复合结构来单独地处理图 5 所示的红色和蓝色矩形这两个部分执行矩阵，并在最后合并部分解决方案以获得完整的动作向量。

该算法的完整信息可参见图 6。我们使用一个外循环对 $S_{10,1}$、$S_{10,2}$、$S_{10,3}$ 以及 $S_{10,4}$ 的所有可能值进行迭代。假定该值被准确猜出，我们首先关注执行矩阵的上部，并找到可将 $S_{0,1}$、$S_{0,2}$、$S_{0,3}$、$S_{0,4}$ 转换到 $S_{10,1}$、$S_{10,2}$、$S_{10,3}$、$S_{10,4}$ 的所有部分动作向量 a_1、……a_{10}。对这个较小的执行矩阵使用简单的 MITM 算法即可完成该任务。对于我们使用 MITM 算法得到的每个解决方案 a_1、……a_{10}，我们都会将它的动作应用于完整状态 S_0，获得完整状态 S_{10} 的候选值，并将其存储在列表中的 a_1、……a_{10} 的旁边。使用 MITM 算法完成填充列表的任务后，我们根据 S_{10} 的值对列表进行排序（如采用一些词典编纂顺序）。

现在，我们关注执行矩阵的底部，并找可将 $S_{10,1}$、$S_{10,2}$、$S_{10,3}$、$S_{10,4}$ 转换到 $S_{20,1}$、$S_{20,2}$、$S_{20,3}$、$S_{20,4}$ 的*所有*部分*动作向量*。我们采取的思路与之前部分的相同，也就是说，我们对执行矩阵的底部执行了 MITM 算法。对于我

图 5.魔方执行矩阵的剖析算法

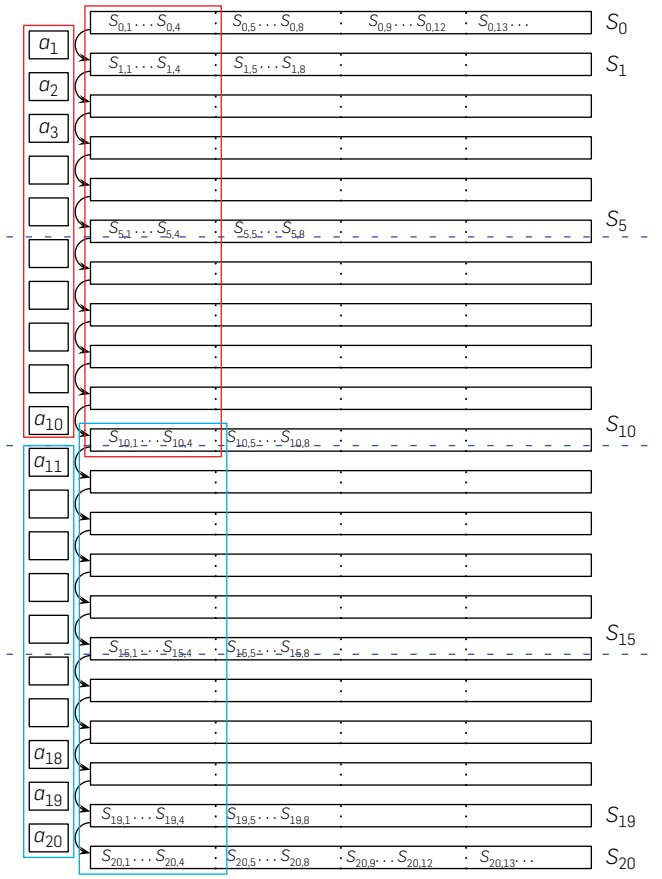


图 6.使用剖析算法解决魔方问题

魔方剖析算法

Input:An initial state S_0 and a final state S_{20}
for all $S_{10,1}$, $S_{10,2}$, $S_{10,3}$, $S_{10,4}$ do
Obtain all candidate (a_1, a_2, …, a_{10}) satisfying $S_{10,1}$, $S_{10,2}$, $S_{10,3}$, $S_{10,4} = a_{10}(\ldots(a_2(a_1(S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4})))\ldots)$ by calling *PartialMITM*($S_{0,1}$, $S_{0,2}$, $S_{0,3}$, $S_{0,4}$, $S_{10,1}$, $S_{10,2}$, $S_{10,3}$, $S_{10,4}$)
for all obtained a_1, a_2, … a_{10} do
Compute $S_{10,5}$, $S_{10,6}$, $S_{10,7}$, $S_{10,8} = a_{10}(\ldots(a_2(a_1(S_{0,5}, S_{0,6}, S_{0,7}, S_{0,8})))\ldots)$
Store ($S_{10,5}$, $S_{10,6}$, $S_{10,7}$, $S_{10,8}$, a_1, a_2, …, a_{10}) in L_{10}
Sort L_{10} according to the values of $S_{10,5}$, $S_{10,6}$, $S_{10,7}$, $S_{10,8}$ in each entry (under some lexicographical order)
Obtain all candidates a_{11}, a_{12}, …, a_{20} satisfying $S_{20,1}$, $S_{20,2}$, $S_{20,3}$, $S_{20,4} = a_{20}(\ldots(a_{12}(a_{11}(S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4})))\ldots)$
by calling *PartialMITM*($S_{10,1}$, $S_{10,2}$, $S_{10,3}$, $S_{10,4}$, $S_{20,1}$, $S_{20,2}$, $S_{20,3}$, $S_{20,4}$)
for all obtained a_{11}, a_{12}, … a_{20} do
Compute $S_{10,5}$, $S_{10,6}$, $S_{10,7}$, $S_{10,8} = a^{-1}{}_{11}(\ldots(a^{-1}{}_{19}(a^{-1}{}_{20}(S_{20,5}, S_{20,6}, S_{20,7}, S_{20,8})))\ldots)$
Search for $S_{10,5}$, $S_{10,6}$, $S_{10,7}$, $S_{10,8}$ in L_{10}
if $S_{10,5}$, …, $S_{10,8}$ are found then
Obtain the associated a_1, a_2, …, a_{10} from L_{10}
if $S_{20} = a_{20}(\ldots(a_2(a_1(S_0)))\ldots)$ then
return a_1, a_2, …, a_{20} as the solution

部分 MITM 程序

Input:An partial initial state $S_{0,1}$, $S_{0,2}$, $S_{0,3}$, $S_{0,4}$ and a partial final state $S_{10,1}$, $S_{10,2}$, $S_{10,3}$, $S_{10,4}$
for all a_1, a_2, …, a_5 do
Compute $S_5 = a_5(\ldots(a_2(a_1(S_{0,1}, S_{0,2}, S_{0,3}, S_{0,4})))\ldots)$
Store ($S_{5,1}$, $S_{5,2}$, $S_{5,3}$, $S_{5,4}$, a_1, a_2, …, a_5) in a list L_5
Sort L_5 according to the values of $S_{5,1}$, $S_{5,2}$, $S_{5,3}$, $S_{5,4}$ in each entry (under some lexicographical order)
for all a_6, a_7, …, a_{10} do
Compute $S_{5,1}$, $S_{5,2}$, $S_{5,3}$, $S_{5,4} = a^{-1}{}_6\ (\ldots(a^{-1}{}_9\,(a^{-1}{}_{10}\,(S_{10,1}, S_{10,2}, S_{10,3}, S_{10,4})))\ldots)$
Search for $S_{5,1}$, $S_{5,2}$, $S_{5,3}$, $S_{5,4}$ in L_5
if $S_{5,1}$, …, $S_{5,4}$ are found then
Obtain the associated a_1, a_2, …, a_5 from L_5
return a_1, a_2, …, a_{10} as a candidate solution

们获得的每个解决方案 a_{11}、……a_{20}，我们将其逆转动作应用于 S_{20}，并获得 S_{10} 的值。然后，我们检查了 S_{10} 在经排序的列表中的匹配情况，对于每个匹配情况，我们都输出一个完整解决方案 a_1、a_2、……a_{20}。

为分析该算法，我们对 $S_{10,1}$、$S_{10,2}$、$S_{10,3}$、$S_{10,4}$ 设定了固定值，并评估了我们期望的（较小的）执行矩阵上部的解决方案的平均数量。也就是说，我们计算了可将 $S_{0,1}$、$S_{0,2}$、$S_{0,3}$、$S_{0,4}$ 转换到 $S_{10,1}$、$S_{10,2}$、$S_{10,3}$、$S_{10,4}$ 的动作向量 a_1、……a_{10} 的预期数量。首先，我们注意到，可能的动作向量 a_1、……a_{10} 的数量约为 2^{39}。每个此类动作都可以将 $S_{0,1}$、$S_{0,2}$、$S_{0,3}$、$S_{0,4}$ 转换到一个任意的部分状态，且该部分状态匹配 $S_{10,1}$、$S_{10,2}$、$S_{10,3}$、$S_{10,4}$ 的可能性约为 $1/(24^4) \approx 2^{-18.5}$（与 $S_{10,1}$、$S_{10,2}$、$S_{10,3}$、$S_{10,4}$ 的可能值的数量成反比）。因此，解决方案（存储在经排序的列表中）的预期数量为 $2^{39} \times 2^{-18.5} = 2^{20.5}$。

一般而言，MITM 算法的时间复杂性约等于搜索空间的平方根，因此执行矩阵上部的时间复杂性约为 $2^{39/2} = 2^{19.5}$。然而，由于在这种情况下，我们无法将该问题分成两个完全相同大小的部分，我们预期会有 $2^{20.5}$ 个解决方案（用于列举并存储），因此其时间复杂性略微上涨至 $2^{20.5}$。这也是针对执行矩阵底部的 MITM 算法的预期时间复杂性（尽管我们不在这里存储解决方案，但会立刻检查每个解决方案）。由于我们拥有一个执行 $24^4 \approx 2^{18.5}$ 时间的外循环，完整算法的预期时间复杂性约为 $2^{18.5+20.5} = 2^{39}$。预期的内存复杂性为 $2^{20.5}$，以便存储针对上部的 MITM 解决方案（请注意，我们重复使用该内存来猜测 $S_{0,1}$、$S_{0,2}$、$S_{0,3}$、$S_{0,4}$）。

4. 改进的剖析方法

进一步观察第三节展示的算法，您会发现该算法以不同方式处理执行矩阵的上部和底部。事实上，来自上部的建议被存储在一个表格中（图 6 中的例子中的 L_{10}），来自底部的建议则动态地对比表格值进行检查。因此，尽管上部建议数量的上限受制于算法可使用的

内存大小，底部建议数量可为任意大，并且以任意顺序动态地生成。

这意味着，执行矩阵的非对称分割（底部明显大于上部）可改善算法的效果。

在本节中，我们证明了事实的确如此。用于破解第三节中的魔方的算法在 PC 上具有可行性，因此进一步改进该算法的价值并不大，我们选择了另一个经典搜索问题*组合划分问题*作为本节的运行示例。

该问题定义如下：我们被给定一组 n 个整数，$U = \{x_1, x_2, \ldots, x_n\}$。我们的目标是将 U 分为两个互补子集 U_1 和 U_2，两者的元素之和相同，也就是

$$\sum_{x_i \in U_1} x_i = \sum_{x_j \in U_2} x_j. \qquad 1.$$

组合划分问题已被证明是 NP 完全的[4]，因此，我们通常不对亚指数解决方案抱有希望。即便如此，我们仍可以采用各类方法来高效地找到不同情况下的解决方案，当存在很多划分 (U_1, U_2) 可以满足方程 (1) 时更是如此。我们因此考虑了该问题的最难实例，其中每个 x_i 都在二进制表示法（即 $x_i \approx 2^n$）中拥有 n 位数。在这种情况下，最多只有数个解决方案 (U_1, U_2) 可能存在，而且 (目前) 尚无任何亚指数时间算法。为简单起见，我们侧重于该问题的*模块化*变体，方程 (1) 将稍加改动为

$$\sum_{x_i \in U_1} x_i \equiv \sum_{x_j \in U_2} x_j \pmod{2^n}. \qquad 2.$$

作为一个具体的数值例子，考虑案例 $n = 112$。在该情况下，检查所有的 2^{112} 个可能的划分当然是根本不可行的。标准的 MITM 算法可以将时间复杂性缩减至 2^{56}，同时将空间复杂性提高至 2^{56}，这在目前也是不可行的。如上所示，该问题可以表达为一个双复合问题，因此，可以使用第 3 节中的方法来获得更好的折衷的结果，即 $T = 2^{56}$，$S = 2^{28}$。尽管，这些数字近乎实用，但由于所需内存相当大，所以在计算中无法使用 FPGA，因此这也是基本不可行的。我们在下文展示了，借助一个*非对称*的剖析算法变体，我们能够获得时间复杂性 $T = 2^{64}$（仅仅大了 2^8 倍）和空间复杂性 $S = 2^{16}$（小了 2^{12} 倍），这样一来，就可以使用内存受约束的 FPGA 来大幅提高计算速度。请注意，非对称剖析算法优于对称剖析算法，根据复杂性测算 $S \times T$，二者存在 16 的因子差别（复杂性分别为 2^{80} 和 2^{84}）。该因子尽管并非很大，但可在实际情形中造成差异。

4.1 将组合划分表达为双复合搜索问题

为了将剖析算法应用于组合划分问题，我们必须找到一种方式来将该问题表达为双复合搜索问题。

首先，我们将它表达为复合问题。我们将选择划分 (U_1, U_2) 的问题视为一个长度为 n 的基本决策序列，其中第 i 个决策为是否分配 $x_i \in U_1$ 或 $x_i \in U_2$。我们引入一个计数器 C，并在最初将其设置为 0，然后在第 i 个步骤，如果选择为 $x_i \in U_1$，则 C 被 $C + x_i \pmod{2^n}$ 代替；如果选择为 $x_i \in U_2$，则 C 被 $C - x_i \pmod{2^n}$ 代替。请注意，在第 n 步以后的 C 值为 $\sum_{x_i \in U_1} x_i - \sum_{x_j \in U_2} x_j \pmod{2^n}$，因此，这一系列选择可以导出期望的解决方案，如果 C 最终值为零。

在该表达式中，划分问题具备了复合问题的一切要素：初始状态（$C_{\text{初始值}} = 0$），最终状态（$C_{\text{最终值}} = 0$），一组由 n 个步骤组成的序列，在每个步骤我们都从两种可能的基本动作中选出一种。我们的目标是找到一组可以使初始状态导向最终状态的选择序列。就执行矩阵而言，我们将 S_i 定义为 i 步（n 位二进制数）以后的 C 值，将 a_i 定义为可将 S_{i-1} 转换到 S_i 的动作，其可能值为 $C \leftarrow C + x_i \pmod{2^n}$ 或 $C \leftarrow C - x_i \pmod{2^n}$。

第二步，我们将该问题表达为双复合问题。我们在此处依据的主要观察结果是，对于任意两个整数 a 和 b，第 m 个最低有效位 (LSB) $a + b \pmod{2^n}$ 仅取决于 a 的 m 个最低有效位 (LSB) 和 b 的 m 个最低有效位 (LSB)（而不是它们的其他数位）。因此，如果我们知道 S_{i-1} 的 m LSB 以及动作 a_i，我们就可以计算 S_i 的 m LSB。

根据这个观察，我们将 $S_{i,j}$ 定义为第 j 个 S_i 的 LSB。这可以导出 n x n 的执行矩阵 $S_{i,j}$，$i, j \in 1, 2, \ldots, n$，属性为如果我们选择执行矩阵（*包含矩阵最右侧列*）中的任何矩形，沿上部边缘的子状态的知识 $S_{i-1}^{j}, S_{i-1}^{j+1}, \ldots, S_{i-1}^{k}$ 和右侧的动作 a_i、a_{i+1}、……a_ℓ 的知识足以计算沿底部的次状态。$S_\ell^j, S_\ell^{j+1}, \ldots, S_\ell^k$

请注意，我们的执行矩阵满足的条件比双复合问题定义中给定的条件更弱，因为在我们的情况中，“矩形”属性仅适用于特定类型的矩形而不是所有的矩形。然而，正如下一分节所述，这种较弱的属性足以应用我们提出的一切剖析算法。[b]

4.2. 适用于复合划分问题的剖析算法

该算法的基本思想是将状态矩阵划分成各自包含 $n/7$ 步骤的 7 个 (!) 部分，其中 3 个部分属于上部 S^t，4 个部分属于底部 S^b。划分通过列举状态 $S_{3n/7}$ 的 $2n/7$ LSB 来获得。

[b] 为简单起见，我们忽视了进位问题。我们注意到，为了获知用于执行剖析算法所需的所有进位，我们从最低有效位到最高有效位猜测 $S_{i,j}$。如欲了解更多信息，请参见本文的拓展部分。[2]

对于这些位的每个v值，我们在上部执行了一个简单的MITM算法，并产出约$2^{3n/7} \times 2^{-2n/7} = 2^{2n/7}$个可能的动作$a_1$、$a_2$、……$a_{3n/7}$的组合，从而导出状态$S_{3n/7}$，其$2n/7$ LSB相当于向量v。对于每一个上述组合，我们计算了状态$S_{3n/7}$的完整值。$S_{3n/7}$的结果值连同相应的组合a_1、a_2、……$a_{3n/7}$都被存储在列表中。

然后，我们考虑底部，并应用第3节中描述的剖析算法（因此，我们将其划分为分别含有$n/7$个步骤的4个区块）。这产生了$2^{4n/7} \times 2^{-2n/7} = 2^{2n/7}$个可能的动作$a_{(3n/7)+1}$、$a_{(3n/7)+2}$、……$a_n$的组合，它们可以导出（以逆转方向）状态$S_{3n/7}$，该状态的$2n/7$ LSB相同于向量v。对于每一个上述组合，我们计算了$S_{3n/7}$的完整值，并将其与表格中的值对比。如果存在匹配情况，就意味着相应的序列$\{a_1$、a_2、……$a_{3n/7}\}$和$a_{(3n/7)+1}$、$a_{(3n/7)+2}$、……a_n匹配，从而得出该问题的解决方案。请注意，如果存在任何解决方案，那么我们的方法必须将其找出，因为它实际上检验了所有可能的动作组合（尽管以一种复杂的方式）。该算法的伪代码可参见图7。

该算法的内存复杂性为$O(2^{n/7})$，正如上部的标准MITM算法和底部的剖析算法都带有该复杂性。（底部的算法的复杂性为$(2^{4n/7})^{1/4} = 2^{n/7}$。）

时间复杂性为$2^{4n/7}$。实际上，状态$S_{3n/7}$中的枚举是针对$2^{2n/7}$值而执行的，上部的标准MITM算法和底部的剖析算法都需要$2^{2n/7}$的时间，而余下的$2^{2n/7}$个可能的动作$a_{(3n/7)+1}$、$a_{(3n/7)+2}$、……a_n的组合都将立刻被检查。这使得时间复杂性为$2^{2n/7} \times 2^{2n/7} = 2^{4n/7}$。

在$n = 112$的特殊情况下，7个区块各自包含16个步骤，枚举针对状态S_{42}的28个LSB执行，内存复杂性为$2^{n/7} = 2^{16}$，时间复杂性为$2^{4n/7} = 2^{64}$。

4.3. 高级剖析算法

第3节和本节所展示的算法是两种最简单的剖析算法，它们反映了剖析算法背后的基本思路。在本文的拓展部分，[2]我们展示了更多高级剖析算法，包括将矩阵划分为特殊数字部分（如11和29），并显示了此类选择在我们的总体框架内的最佳性。

迄今为止，我们仅考虑了不允许失败的搜索算法（即，如果该问题存在任何解决方案，那么我们的总会找出所有的解决方案。如何实例不是随机选择的，或解决方案数量过多，则运算时间可能超出预期水平。）在Dinur等人的著述中，[2]我们还考虑了可能无法找到存在性低的解决方案的算法，并展示了如何改进算法在这种情况下的效率，方法为结合*并行碰撞搜索*这一经典方法，该方法由Wiener和van Oorschot在1996年设计。[7]

图7.使用剖析算法解决划分问题

划分剖析算法

```
Input:U = {x_1, x_2, ... x_n}
for all S_{3n',1}, S_{3n',2}, ..., S_{3n',2n'} (2n' LSBs of S_{3n'}) do
  call PartialMITM(S_{0,1}, S_{0,2}, ..., S_{0,2n'}, S_{3n',1}, S_{3n',2}, ..., S_{3n',2n'}, 3n')
  for all obtained a_1, a_2, ..., a_{3n'} do
    Compute the 5n' MSBs of S_{3n'} = a_{3n'}(...(a_2(a_1(S_0)))...)
    Store (S_{3n'}, a_1, a_2, ..., a_{3n'}) in L_{3n'}
  Sort L_{3n'} according to the values of S_{3n'}
  for all S_{5n',1}, S_{5n',2}, ..., S_{5n',n'} do
    call PartialMITM(S_{3n',1}, S_{3n',2}, ..., S_{3n',n'}, S_{5n',1}, S_{5n',2}, ..., S_{5n',n'}, 2n')
    for all obtained a_{3n'+1}, a_{3n'+2}, ..., a_{5n'} do
      Compute the n' bits S_{5n',n'+1}, S_{5n',n'+2}, ..., S_{5n',2n'} = a_{5n'}(...(a_{3n'+2}(a_{3n'+1}(S_{3n',n'+1}, S_{3n',n'+2}, ..., S_{3n',2n'})))...)
      Store (S_{5n'}, a_{3n'+1}, a_{3n'+2}, ..., a_{5n'}) in L_{5n'}
    Sort L_{5n'} according to the values of S_{5n'}
    call PartialMITM(S_{5n',1}, S_{5n',2}, ..., S_{5n',n'}, S_{7n',1}, S_{7n',2}, ..., S_{7n',n'}, 2n')
    for all obtained a_{5n'+1}, a_{5n'+2}, ..., a_{7n'} do
      Compute S_{5n',n'+1}, S_{5n',n'+2}, ..., S_{5n',2n'} = a^{-1}_{5n'+1}(...(a^{-1}_{7n'-1}(a^{-1}_{7n'}(S_n)))...)
      Search for S_{5n'} in L_{5n'}
      if S_{5n'} value is found then
        obtain a_{3n'+1}, a_{3n'+2}, ..., a_{5n'} from L_{5n'}
        for all obtained a_{3n'+1}, a_{3n'+2}, ..., a_{5n'} do
          Compute S_{3n',2n'+1}, S_{3n',2n'+2}, ..., S_{3n',7n'} = a^{-1}_{3n'+1}(...(a^{-1}_{7n'-1}(a^{-1}_{7n'}(S_{7n'})))...)
          Search for S_{3n'} in L_{3n'}
          if S_{3n'} value is found then
            obtain a_1, a_2, ..., a_{3n'} from L_{3n'}
            return a_1, a_2, ..., a_{7n'} as a solution
```

部分MITM程序

```
Input:A partial state S_{0,1}, S_{0,2}, ..., S_{0,tn'}, a partial state S_{(t+1)n',1}, S_{(t+1)n',2}, ..., S_{(t+1)n',tn'}, and "distance" (t + 1)n'
for all a_1, a_2, ..., a_{n'} do
  Compute S_{n',1}, S_{n',2}, ..., S_{n',tn'} = a_{n'}(...(a_2(a_1(S_{0,1}, S_{0,2}, ..., S_{0,tn'})))...)
  Store (S_{n',1}, S_{n',2}, ..., S_{n',tn'}, a_1, a_2, ..., a_5) in a list L_{n'}
Sort L_{n'} according to the values of S_{n',1}, S_{n',2}, ...S_{n',tn'}
for all a_{n'+1}, a_{n'+2}, ..., a_{(t+1)n'} do
  Compute S_{n',1}, S_{n',2}, ..., S_{n',tn'} = a^{-1}_{n'+1}(...(a^{-1}_{(t+1)n'-1}(a^{-1}_{(t+1)n'}(S_{(t+1)n',1}, S_{(t+1)n',2}, ..., S_{(t+1)n',tn'})))...)
  Search for S_{n',1}, S_{n',2}, ..., S_{n',tn'} in L_{n'}
  if S_{n',1}, S_{n',2}, ..., S_{n',tn'} are found then
    Obtain the associated a_1, a_2, ..., a_{n'} from L_{n'}
    return a_1, a_2, ..., a_{(t+1)n'} as a candidate solution
```

5. 结论

我们在本文介绍了双复合搜索问题的想法，并设计了一种叫做剖析算法的全新算法，以改善解决此类问题的时间和空间复杂性。我们使用这种方法解决两种标准类型问题（魔方和组合划分），以此展示它的应用方式。然而，这些方法的一些最激动人心的用途在于

密码分析，这不在本文讨论范围内。比如，很多银行使用 *Triple-DES* 这种传统密码技术，该方法使用三种相互独立的密钥对敏感财务数据进行三次加密。人们自然会产生这样的问题，使用 *Quadruple-DES*（使用四种相互独立的密钥进行四次加密）是否可以显著提高安全性，因为其密钥更长，加密过程更复杂。通过使用我们的全新剖析方法，我们展现了一种令人出乎意料的结果，找出 Quadruple-DES 的全部密码与找出较简单的 Triple-DES 加密方案的全部密钥所需的时间和空间复杂性是基本相同的，因此，将 Triple-DES 升级至 Quadruple-DES 并不会显著提高安全性。

致谢

第二作者 (O.D.) 得到了以色列科学基金的部分支持（授权编号：827/12）以及德国和以色列基础科学研究和发展的部分支持（授权编号：2282-2222.6/2011）。第三作者 (N.K.) 得到了 Alon 奖学金的支持。

参考资料

1. Davidson, M., Dethridge, J., Kociemba, H., Rokicki, T. God's number is 20, 2010. http://cube20.org.
2. Dinur, I., Dunkelman, O., Keller, N., Shamir, A. Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. In *CRYPTO*, R. Safavi-Naini and R. Canetti, eds. Volume 7417 of *Lecture Notes in Computer Science* (2012). Springer, 719–740.
3. Fiat, A., Moses, S., Shamir, A., Shimshoni, I., Tardos, G. Planning and learning in permutation groups. In *FOCS*. IEEE Computer Society, 1989, 274–279.
4. Garey, M.R., Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
5. Horowitz, E., Sahni, S. Computing partitions with applications to the knapsack problem. *J. ACM 21*, 2 (1974), 277–292.
6. Slocum, J. *The Cube: The Ultimate Guide to the World's Bestselling Puzzle—Secrets, Stories, Solutions*. Black Dog & Leventhal Publishers, 2011.
7. van Oorschot, P.C., Wiener, M.J. Improving implementable meet-in-the-middle attacks by orders of magnitude. In *CRYPTO*, N. Koblitz, ed. Volume 1109 of *Lecture Notes in Computer Science* (1996). Springer, 229–236.

Itai Dinur 和 Adi Shamir ({itai.dinur, adi.shamir}@weizmann.ac.il)，计算机科学系，魏兹曼研究所，雷荷屋，以色列。

Orr Dunkelman (orrd@cs.haifa.ac.il)，计算机科学系，海法大学，以色列。

Nathan Keller (nathan.keller@biu.ac.il)，数学系，巴伊兰大学，以色列。

译文责任编辑：孙晓明