服务计算技术与系统教育部重点实验室
集群与网格计算湖北省重点实验室

并行與分布式計算通訊

BING XING YU FEN BU SHI JI SUAN TONG XUN

2018年第4期 总第35期 2018年12月

地 址：武汉市华中科技大学东五楼二楼
邮 编：430074
电 话：（027）87541924 或 87543529
传 真：（027）87557354
E-mail：www.hust@hust.edu.cn
网 址：http://grid.hust.edu.cn


CNCC
2018中国计算机大会
会士论坛
计算科学推动学科交叉创新


CNCC
2018中国计算机大会
大数据时代的计算机教育


CNCC
CPU及xPU芯片未来之路
金海 华中科技大学
孙贤和 美国伊利诺伊理工学院
黄畅 地平线联合创始人
陈云霁 中科院计算所
2018中国计算机大会
CNC


计算机大会

2018年度实验室学术成果展示

白驹过隙，时光荏苒。从“你好！2018”到“再见！2018”，从年初的信心满满到年终的欢欣喜悦。这一年，实验室全体教师和学生围绕年初既定目标“撸起袖子加油干”，取得了丰硕的成果。经统计，2018年实验室师生发表TOP80高档次会议论文18篇，再创新高；发表IEEE/ACM Transactions期刊论文15篇，CCF B类会议及其他SCI论文21篇。

在ACM Multimedia 2018会议上，马晓静等人围绕同态加密中密文膨胀效应问题，提出了一种同态域下的JPEG解压缩算法。在WWW 2018会议上，黄宏等人针对单一平台上用户的浏览与购买行为难以捕捉的问题，提出了跨平台的用户购买框架。在INFOCOM 2018会议上，黄钰瑶等人针对传统标准物理干扰absMAC层实现的不精确性问题，提出了一种用于GLB的随机分布式算法；公绪辉等人提出了针对隐私保护数据聚合中通信和计算复杂度大的问题，提出了一种不需要可信任第三方参与的协议；牛轶佩等人提出了一种面向微服务的负载均衡策略；曾超冰等人提出了一种面向虚拟网络功能性能干扰的测量方法；费新财等人提出了一种基于主动预测的自适应虚拟网络功能扩展和流量调度；在ICNP 2018会议上，林昌富等人针对任务依赖关系导致的任务级联等待问题，提出了一种低延迟和高容错的分布式流处理系统Ares；在ICDCS 2018会议上，柳密等人针对跨进程操作符间通信开销大的问题，提出了一种低延迟的数据流处理系统TurboStream；戴小海等人针对不同区块链中的数据孤岛问题，提出了一种面向跨链交互的区块链新型架构；李逍遥等人针对当前软件网络功能深度包处理时资源消耗大的问题，提出了一种基于动态硬件库的FPGA-CPU协同设计框架DHL；姜炜祥等人针对非IT设施难以进行细粒度的能耗测量及有效的能耗管理问题，运用经济学中著名的夏普利值方法为数据中心中大量的动态虚拟机的非IT能耗实现公平高效的计量。在PACT 2018会议上，姚鹏程等人针对图计算中的数据冲突问题，提出了一种基于并行累加的图计算加速器AccuGraph；郑龙等人针对多线程程序中数据竞争调试与诊断的问题，提出了面向数据竞争的并发调试系统RaceDebugger。在NDSS 2018会议上，李珍等人针对软件漏洞自动检测中的误报和漏报的问题，提出了一种基于深度学习的漏洞检测系统同时兼顾误报和漏报的有效平衡。在USENIX ATC 2018会议上，张宇等人针对大量并发图分析问题，提出了一种关联性感知的并发图处理系统。在CGO 2018会议上，郑龙等人提出了一种基于分布式图计算的并发调试框架。在ACSAC会议上，刘长鸣等人针对软件并发漏洞的特点，结合经典的模糊测试工具AFL，提出了一个轻量级针对并发的模糊测试框架。

在TNSM期刊中，袁斌等人针对错误的内部交换机会导致控制器无法得到正确的网络状态信息问题，提出了基于拜占庭协议的错误交换机容忍机制。在TKDD期刊中，黄宏等人探究了闭环三角形的形成是否会影响原有三角形的交互强度。在ACM Computing Surveys期刊中，郑志高等人综述了GPU图计算的关键问题，全面总结了当前的研究现状和进展，详细分析了GPU图计算存在的挑战性问题，并深入探讨了未来的研究方向。在ToN期刊中，华强胜等人针对无线干扰环境下的无线网络中信息交换问题，提出了基于SINR模型的多跳无线网络中稳定局部广播协议。在TPDS期刊中，王娜等人针对动态图中算法执行效率低下的问题，提出了一种基于匹配的并行核值维护方法；王新猴等人针对云无线网和边缘云计算环境下的利润权衡问题，拓展了标准李雅普诺夫技术，设计了能够在多个时间片之间进行在线决策的VariedLen算法；张宇等人提出了一种基于磁盘的有向图处理方法；林敏豪等人针对现有数据竞争检测同步分析开销大的问题，提出了一种基于Pending Period的数据竞争检测范式。在TACO期刊中，刘博等人针对GPU DRAM容量无法满足训练过程中内存消耗日益增长的要求，提出了一种基于模型层敏感的内存复用方法Layrub；郑龙等人针对图数据复杂依赖导致的同步和通信开销大的问题，提出了一种基于符号执行的图计算系统SymGraph。在TII期刊中，贺双洪等人针对传感器设备数据所面临的安全性问题，提出了一种面向云协同无线传感器网络的轻量级可搜索公钥加密方案。在TSC期刊中，陈汉华等人在线社交网络搜索开销大的问题，提出了一种基于摘要索引的在线社交网络搜索机制。在TMPECS期刊中，陈姝彤等人提出了面向数据中心废热利用的高效在线交易与双赢激励机制。在TKDE期刊中，张宇等人提出了一种基于交替式数据处理的图计算优化方法；石宣化等人针对同步处理模型存在通信开销大的问题，提出了一种基于GPU的异步图计算系统Frog。

本期季刊以实验室2018年学术成果作为主题，展示实验室本年度发表的高档次会议及期刊文章。“宝剑锋从磨砺出，梅花香自苦寒来”；回首2018，我们成果颇丰，这是老师和同学们辛勤工作的智慧结晶；展望2019，我们亦胸有成竹，让我们携手共进，奋力开创实验室工作新局面，聚力谱写实验室成果新篇章！

郑　龙

2018年12月31日

主　编：金　海
本期执行主编：郑　龙
编　委：陈汉华、代炜琦、丁晓锋、耿　聪、顾　琳、胡　侃、华强胜、黄　宏、蒋文斌、廖小飞、刘方明、刘海坤、刘英书、陆　枫、黄　宏、吕新桥、马晓静、羌卫中、邵志远、石宣化、王多强、吴　松、肖　江、谢　夏、徐　鹏、余　辰、于东晓、袁平鹏、章　勤、张　宇、赵　峰、郑　龙、郑　然、邹德清
（按姓氏拼音排序）
责任编辑：吴　未

地　址：武汉市华中科技大学东五楼二楼
邮　编：430074
电　话：（027）87541924 或 87543529
传　真：（027）87557354
E-mail：wwuhust@hust.edu.cn
Homepage：http://grid.hust.edu.cn
（此刊仅供内部交流学习）

卷首语

热点

封面人物

专栏

声音

动态

推荐

交流

边缘计算概览

（高 彬 整理）

伴随着数据越来越多的在网络边缘诞生，边缘计算应运而生，以解决传统云计算面临的问题。本文将对边缘计算这一代新兴云计算范式进行概述。

1. 边缘计算及其必要性

边缘计算是指在靠近物或数据源头的一侧，采用网络、计算、存储、应用核心能力为一体的开放平台，就近提供最近端服务。边缘计算的诞生的必要性有以下原因。首先是传输速度瓶颈。例如自动驾驶汽车每秒钟生成1000MB的数据，安全驾驶的时延要求为ms级，数据发送到传统云端进行处理，响应时间不能忍受，此时需要边缘云来提供支持。其次是设备能量受限。物联网中的大多数设备都是能量受限的东西，无法支撑通信模块长时间传输大量数据至传统云端，因此将数据卸载到边缘节点更加节能。此外，还有隐私保护要求。将数据传输到公有云，会增加数据被泄露的可能性。将数据传输到可控的边缘节点进行处理将比上传到传统公有云更好的保护用户隐私。

2. 边缘计算的关键指标

首先是延迟。边缘云更靠近用户，避免了广域网中的远距离传送，降低了传输时延。其次是带宽。边缘计算在靠近用户的地方进行，对于此类短距离传输，可以建立高带宽的网络接入点，从延迟角度来看，高带宽可以大大减少大数据类应用的传输时间，边缘云也提供计算能力，大部分数据都可以使用边缘节点进行预处理，只传输中间数据，减少带宽浪费。最后是能量。边缘计算场景中，计算能耗可以估计为计算成本与传输成本，多层传输会显著增加开销，需要合理的任务卸载策略降低能量消耗。

3. 边缘计算未来的挑战和机遇

第一个是任务卸载。在边缘计算的新型结构下，如何缓存数据，如何放置计算任务，如何同步数据等都成为需要亟待解决的问题。第二个移动性。在边缘云场景中，设备往往具备移动性，边缘设备不断在各个边缘云中进行切换，如何感知设备位置，如何迁移任务，如何选择接入点等都是未来研究需要关注的方面。第三个是边缘计算平台。边缘计算中，计算并不只是在传统云中发生，还有可能发生在设备、边缘云中，如何在边缘计算范例中跨多层进行协作，也是边缘计算平台应该提供的特性。第四个是隐私和安全。用户数据存储在用户的边缘端具有更好的安全性，但是如何使用这些数据，如何将其提供给服务商以获取必要的服务需要被解决，应该开发更多的工具和方法以保障边缘的安全性。

中国的Infervision帮助医院从图像中检测癌症

（赵 祥 整理）

总部位于北京的Infervision是全球少数几个人工智能初创公司之一，它们通过深度学习来改善医学成像分析，这种技术同样为人脸识别和自主驾驶提供动力。

这家初创公司迄今已从红杉资本(Sequoia Capital China)等主要投资者那里筹集了7,000万美元，其始于找出中国常见的致死原因肺部癌细胞。本周，在芝加哥召开的北美放射学会年会上，这家成立了三年的公司宣布，将计算机视觉能力扩展到其他与胸部相关的疾病，如心脏钙化。

Infervision的创始人兼首席执行官陈宽表示，通过增加AI工作的场景能够为医生提供更多的帮助。虽然医生可以通过一次图像扫描发现几十种疾病，但是AI需要学会如何一次识别多个目标物体。但是，机器在其他方面已经超过了人类。首先，它们速度更快。医生通常需要15到20分钟来仔细检查一幅图像，而Infervision的AI可以在30秒内处理这些图像并汇总报告。

人工智能也解决了长期存在的误诊问题。中国临床报纸《医学周刊》报道说，在诊断煤矿工人常见的黑肺病时，不足五年经验的医生正确回答了44%。浙江大学在1950年至2009年

间对尸体解剖进行检查的研究发现，总的临床误诊率平均为46%。这位创始人声称他的公司能够提高20%的准确率。

像任何深度学习公司一样，Infervision需要利用来自不同来源的数据持续训练其算法。截至本月，这家初创公司与280家医院合作，其中20家在中国境外，并且每周稳步增加12家新的合作伙伴。报告还称，中国70%的顶级医院使用肺部专用AI工具。

理解 Ethereum 第二层扩容方案： State Channels 、 plasma 以及 Truebit

（郝威峰 整理）

以太坊是一个开源的有智能合约功能的公共区块链平台，它希望搭建一个安全、易于使用的去中心化网络。只有在关键的底层架构完成的前提下才能实现以太坊大规模应用。我们称那些正在致力于构建以太坊的底层架构以及扩展其性能的项目为扩容解决方案。

1. 公链的扩容挑战

以太坊最常讨论的扩容挑战是交易吞吐量问题，目前每一个以太坊区块链上的操作都必须由网络中每个节点并行处理，我们需要找到不提高单个节点工作量的情况下，负担更多有用工作的方法，目前有两种方法可以解决这个问题：

第一种方法称为“分片”，区块链被分成很多分片，每个分片都可以独立地处理交易，分片通常被称为是“ layer 1” 扩容解决方案。

第二种选择是从反方向进行考虑的，即并不是提高以太坊区块链本身的能力，而是利用已有的能力做更多的事情，这些被称为“ layer 2” 解决方案。

2. layer 2 扩容解决方案是加密经济解决方案

支撑公有链最基础的动力源泉就是加密经济共识，通过仔细协调激励机制以及通过软件和加密算法对激励进行保障，我们就可以创建对于系统内部状态达成共识的可靠计算机网络。 layer 2 解决方案背后的见解，就是我们可以把核心内核的确定性作为锚定，来提升区块链性能。

I. State channels

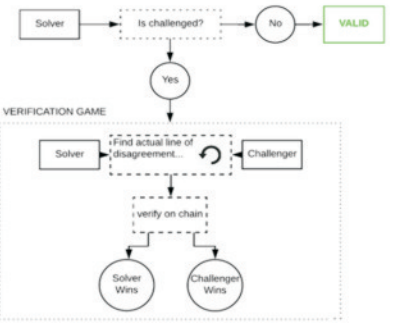
State Channel 是一种用于执行交易和其他状态更新的“ off-chain”技术。一个状态通道“内”发生的事务仍保持了很高的安全性和不可更改性，如果出现任何问题，我们仍然可以选择回溯到链上交易中确定的“稳定内核”。比如 Counterfactual 框架是一个可以使开发者可以在他们的应用程序中以模块化的方式使用状态通道；另一个针对以太坊的状态通道项目是 Raiden， 该项目目前正致力于构建一个支付通道网络。

II. Plasma

像状态通道一样， Plasma 是一种管理链下交易的技术，同时依靠底层的以太坊区块链来实现其安全性。 Plasma 采用了一种新思路，它是通过创建依附于以太坊“主”区块链的“子”区块链。其结果就是，我们可以在子链级别上执行许多复杂的操作，在与以太坊主链保持最低限度交互的情况下，运行拥有数千名用户的应用程序。除此之外 Plasma 还创建防止欺诈时不能退回到主链上的机制。

III. Truebit

Truebit 是一种帮助以太坊在链下进行繁重或复杂计算的技术。 Truebit 不会让我们完成更多的交易，但它能让应用程序去处理更复杂的事务，并且仍能被主链验证。我们能够对应用程序做一些有用的操作，这些操作的计算成本太高，无法在链上执行，例如验证来自其他区块链的简单支付验证证明。



3. 结论

以太坊使得我们能够建立第二层网络达到一个新的速度、完成性和运行成本的平衡点，这使得区块链能够完成更多的应用。因为不同的应用有不少的威胁模型，因此采用不同的第二层网络是十分自然的。对于高价值需要抵抗国家级别的侵袭的交易，我们可以在主网上进行；对于数字收集物交易，速度更为重要，因此我们采用 Plasma 。加密经济共识创造的确定性内核是区块链的优势，要完全利用这一优势的唯一方法，就是使用像以太坊这样的可编程区块链，从而进行大规模的应用开发。

黑客向热门 JavaScript 库注入恶意代码窃取 Copay 钱包的比特币

（方升泽 整理）

尽管上周已经发现了恶意代码的存在，但直到今天安全专家才理清这个严重混乱的恶意代码，了解它真正的意图是什么。黑客利用该恶意代码获得（合法）访问热门JavaScript库，通过注射恶意代码从BitPay的Copay钱包应用中窃取比特币和比特币现金。

这个可以加载恶意程序的JavaScrip库叫做Event-Stream，非常受者欢迎，在npm.org存储库上每周下载量超过200万。但在三个月前，由于缺乏时间和兴趣原作者将开发和维护工作交给另一位程序员Right9ctrl。Event-Stream，是一个用于处理Node.js流数据的JavaScript npm 包。

根据Twitter、GitHub和Hacker News上用户反馈，该恶意程序在默认情况下处于休眠状态，不过当Copay启动（由比特币支付平台BitPay开发的桌面端和移动端钱包应用）之后就会自动激活。它将会窃取包括私钥在内的用户钱包信息，并将其发送至copayapi.host的8080端口上。

目前已经确认9月至11月期间，所有版本的Copay钱包都被认为已被感染。今天早些时候，BitPay团队发布了Copay v5.2.2，已经删除Event-Stream和Flatmap-Stream依赖项。恶意的Event-Stream v3.3.6也已从npm.org中删除，但Event-Stream库仍然可用。这是因为Right9ctrl试图删除他的恶意代码，发布了不包含任何恶意代码的后续版本的Event-Stream。

建议使用这两个库的项目维护人员将其依赖树更新为可用的最新版本–Event-Stream版本4.0.1。此链接包含所有3,900+ JavaScript npm软件包的列表，其中Event-Stream作为直接或间接依赖项加载。

5G前夜的高通进击之路：赋能产业新生态

（林 晨 整理）

如果说，4G开启移动互联网的新时代，社交软件、网约车、直播平台、生活服务平台等移动应用兴起，人们通过手机也能解决吃、穿、住、行，打开了新的生活方式；那么5G进一步提升网络速度的同时，还将渗透手机之外的更多垂直行业。高通工程技术高级副总裁、4G/5G业务总经理马德嘉（Dr. Durga Prasad Malladi）曾撰文指出，5G将成为一个统一的连接架构，能够利用不同的频谱、满足不同的服务需求、采取不同的部署模式，从而实现万物互联。”

比如，5G和人工智能的结合，在5G部署、提高用户体验和增强不同应用场景上，具有很大潜力。这不仅能够赋予智能手机更多更新的体验，5G还将建立起一个统一的连接架构，连接无线边缘，为联网设备和服务提供更强的基础能力。因此，高通强调在边缘计算方面将5G与人工智能相互结合：通过人工智能的应用，我们可以将智能扩展到无线边缘，解决海量数据带来的挑战，从而让更多设备之间实现直接交流，实现传递知识和数据，为用户提供更好的体验和服务。

高通中国区董事长孟樸认为，5G与工业领域融合在中国的意义尤其重大，“因为中国成千上百万的工厂都面临着转型升级，而4G和Wi-

Fi的局限性，导致无线网络不能广泛应用于当前的工业生产里，很多工厂使用有线的方式接入互联网。5G到来之后，工厂可以利用5G无线方式操控机器人进行工业制造”。

“过去30年，大家一起努力协作，实现了人与人之间的连接，今后30年，大家有机会把世界所有的万物都智能地连接起来。高通的发展离不开中国的发展，希望产业各方能够继续合作，一起把5G做好，利用5G这个赋能技术，一起改变世界！”孟樑如是说。

网络表示学习的最新发展

（宋 宇 整理）

网络表示学习，即学习网络中节点的低维度潜在表示，所学习到的特征表示可以用作基于图的各种任务的特征，例如分类，聚类，链路预测和可视化。如今，越来越大，越来越复杂的网络被用于越来越多的应用中，针对于一些特定任务适用于大型网络各类网络表示学习方法不断被提出。现有的方法主要分为两个类别，基于随机游走进行求解的和通过矩阵分解进行求解的。

以社交网络为例，facebook的活跃用户高达二十亿，微信也有十多亿活跃用户，这些复杂的巨大社交网络要进行计算时间代价相当之高，因此一个适用于巨大型网络的高效快速的表示学习方法显得十分必要。许多研究人员在现有方法的基础上进行改进，使其能适用于如facebook、微信这样的大型网络，如合肥工业大学汪萌团队提出的approximated error reduction (AER) 模型，苏黎世联邦理工学院Olivier Bachem 等人的工作《Scalable k-Means Clustering via Lightweight Coresets》。

对网络的表示学习主要分为三个方向，一是网络结构的学习，二是网络属性的学习，三是网络动态化的学习。早期网络表示学习相关的重心集中在如何更加准确地保留网络的结构信息和属性信息，近年来网络自身动态变化的特性也更多的被大家所考虑，以满足实际应用的需求。针对网络结构信息的学习，清华大学朱文武团队提出了可以保存任意阶相似度的网络表示学习方法AROPE (arbitrary-order proximity preserved embedding)，同时在保证了其在应用于大型网络时的计算效率，此外该团队还针对网络的动态特性提出了TIMERS模型来解决动态网络表示学习中随时间不断地误差累积。

当知识图谱“遇见”深度学习

（徐 涛 整理）

大数据时代的到来，为人工智能的飞速发展带来前所未有的数据红利。人工智能技术获得了前所未有的长足进步，其进展突出体现在以知识图谱为代表的知识工程以及深度学习为代表的机器学习等相关领域。随着深度学习对于大数据的红利消耗殆尽，深度学习模型效果的天花板日益迫近。另一方面大量知识图谱不断涌现，这些蕴含人类大量先验知识的宝库却尚未被深度学习有效利用。融合知识图谱与深度学习，已然成为进一步提升深度学习模型效果的重要研究热点之一。

近几年在知识图谱技术的推动下，对于机器友好的各类在线知识图谱大量涌现。知识图谱本质上是一种语义网络，表达了各类实体、概念及其之间的语义关系。相对于传统知识表示形式（诸如本体、传统语义网络），知识图谱具有实体/概念覆盖率高、语义关系多样、结构友好（通常表示为RDF格式）以及质量较高等优势，从而使得知识图谱日益成为大数据时代和人工智能时代最为主要的知识表示方式。能否利用蕴含于知识图谱中的知识指导深度神经网络模型的学习从而提升模型的性能，成为了深度学习模型研究的重要问题之一。

现阶段将深度学习技术应用于知识图谱的方法较为直接。大量的深度学习模型可以有效完成端到端的实体识别、关系抽取和关系补全等任务，进而可以用来构建或丰富知识图谱。知识图谱在深度学习模型中的应用主要有两种方式：一是将知识图谱中的语义信息输入到深度学习模型中；将离散化知识图谱表达为连续

化的向量，从而使得知识图谱的先验知识能够成为深度学习的输入。二是利用知识作为优化目标的约束，指导深度学习模型的学习；通常是将知识图谱中知识表达为优化目标的后验正则项。前者的研究工作已有不少文献，并成为当前研究热点。知识图谱向量表示作为重要的特征在问答以及推荐等实际任务中得到有效应用。后者的研究才刚刚起步，主要体现在以一阶谓词逻辑作为约束的深度学习模型。

知识图谱作为深度学习的输入。知识图谱是人工智能符号主义近期进展的典型代表，知识图谱中的实体、概念以及关系均采用了离散的、显式的符号化表示。而这些离散的符号化表示难以直接应用于基于连续数值表示的神经网络。为了让神经网络有效利用知识图谱中的符号化知识，相关研究提出了大量的知识图谱的表示学习方法。知识图谱的表示学习旨在习得知识图谱的组成元素(节点与边)的实值向量化表示。这些连续的向量化表示可以作为神经网络的输入，从而使得神经网络模型能够充分利用知识图谱中大量存在的先验知识。这一趋势催生了对于知识图谱的表示学习的大量研究。

知识图谱作为深度学习的约束。现阶段工作提出将一阶谓词逻辑融合进深度神经网络的模型，并将其成功用于解决情感分类和命名实体识别等问题。逻辑规则是一种对高阶认知和结构化知识的灵活表示形式，也是一种典型的知识表示形式。将各类人们已积累的逻辑规则引入到深度神经网络中，利用人类意图和领域知识对神经网络模型进行引导具有十分重要的意义。其他一些研究工作则尝试将逻辑规则引入到概率图模型，这类工作的代表是马尔科夫逻辑网络，但是鲜有工作能将逻辑规则引入到深度神经网络中，是一个重要的研究趋势。

随着深度学习研究的进一步深入，如何有效利用大量存在的先验知识，进而降低模型对于大规模标注样本的依赖，逐渐成为主流的研究方向之一。知识图谱的表示学习为这一方向的探索奠定了必要的基础。近期出现的将知识融合进深度神经网络模型的一些开创性工作也颇具启发性。但总体而言，当前的深度学习模型使用先验知识的手段仍然十分有限，学术界在这一方向的探索上仍然面临巨大的研究挑战。

ASAP：近似计算用于图数据挖掘

（陈庆祥 整理）

图数据挖掘在大数据公司，尤其是 IT 公司是非常流行的一大类模型，因为是很多实际问题的最直接的解决方法。但是由于图的规模过大，以及图计算的效率低下，图计算性能往往得不到很好的效果。但是有学者发现，很多时候用户不需要非常精确的结果，一个足够准确的结果足以满足大量应用的需求。

伯克利Anand Padmanabha Iyer和约翰霍普金斯大学的Zaoxing Liu 和 Xin Jin在今年OSDI上发表了他们的工作ASAP: Fast, Approximate Graph Pattern Mining at Scale。

他们发现是对于很多graph mining的任务，其实并不需要得到一个完全准确的结果，一个足够准确的结果足以满足大量应用的需求包括：1）社交网络中挖掘相似的子图；2）欺诈检测系统中计算某个pattern的频率。并且对于一个pattern，并不需要把所有的embeddings都输出，所以作者提出使用近似的方法执行图挖掘算法。

作者提出的方法建立在图近似理论之上，通过把Neighborhood sampling这个针对于triangle counting 的近似挖掘算法泛化到更多的图中，并且扩展到分布式环境中，使得系统能够应用到更大的图。

作者开发的ASAP提供了一套API供程序员使用Neighborhood sampling来处理图数据挖掘。并且可以将图数据挖掘扩展到分布式环境中，通过map-reduce的方法提升图挖掘的算法。

实验结果表明ASAP相比state-of-the-art的分布式图挖掘系统，有更好的扩展性，能够扩展到更大的图数据集，执行时间缩短了几个数量级。

关于论文写作的一点经验分享

2017年6月5日，ICDCS 2017 (The 37th IEEE International Conference on Distributed Computing Systems) 在美国亚特兰大召开。CGCL/SCTS实验室博士生姜炜祥的论文“Virtual Machine Power Accounting with Shapley Value”被该会议录用，并在会议上做学术报告。该论文应用经济学中的博弈论来解决虚拟机资源竞争导致的能耗难以准确计量的问题。本刊对论文作者进行了有关科研心得的采访，以下以第一人称陈述他的经验总结。

序 言

刚来到刘方明老师小组的时候，对于发表高水平论文是向往的。可是当真正地开始时，却发现自己如无头苍蝇一般，不知道论文的idea该从何而来，对发表论文满怀焦虑。在此和大家分享一下自己在刘老师小组的学习经历和经验，与实验室的小伙伴们共勉。

论文选题–多交流、多讨论

借用刘老师的话说，论文在形式上来说，不外乎四种：“新问题、新方法”，“新问题、旧方法”、“旧问题、新方法”，“旧问题、旧方法”。比如能耗计量问题是数据中心里的老问题，在07年就有研究服务器能耗的论文出现在ISCA上了。之后便出现一系列研究数据中心虚拟机的能耗计量的论文相继出现在SoCC、SIGMETRICS、Eurosys等一系列高水平的会议中。之前的文章基本都是基于资源–能耗映射模型来进行能耗计量，但是在虚拟机之间存在资源竞争的现象，这种相互之间的影响难以在能耗层面进行量化，这就导致传统的方法失效。论文通过真实的实验测量验证了该现象，问题虽然是旧的，但是还是有新的东西可以挖掘，这就找到了论文的立足点。但是如果仅仅是想着改进传统资源–能耗映射方法，比如多增加一些参数来修正等等，论文就只能是一篇“旧问题、旧方法”，这种修修补补的工作贡献有限，很难发表到高水平的会议上。而博弈论在能耗计量中是一种全新的方法，而且虚拟机之间相互影响的现象本身就和博弈论的应用场景契合。这就赋予了论文工作一种全新的面貌，更容易获得评审的认可。因此论文的idea非常重要，不管什么形式的论文，必须要有鲜明独特的贡献。

尽管如此，真正寻找论文idea并不是一件容易的事情。最好的办法就是和别人多交流，任何简单的想法都应该及时交流和讨论，特别是和已经发表过高水平的论文的人（比如导师、学长）交流。一个不成熟、不经意的想法，经过讨论就可能会慢慢丰富，变成一篇高水平的论文。从最开始刘老师告诉我调研学习夏普利值方法，到寻找问题场景时在组会上的攻防讨论，甚至在去食堂的路上和“硬逼”饭友和自己讨论论文的motivation是否合适，这过程中的点点滴滴至今记忆尤深。从最初只知道一个夏普利值方法，经过反复、广泛和深刻的讨论，慢慢演变出一篇完整的论文idea，这中间让我深刻感受到交流的重要性。以至于到现在，每当我遇到问题，反复思考都没有很好答案的时候，就会找人讨论。目的并不是寻求对方给出答案，而在和对方的描述讨论中，自己也在梳理思路，往往最后是自己找到了想要的答案。这让我想起刘老师常教育我们的一句话：“if you cannot write well, it means you did not think well”，而我个人经验告诉我，要

"think well"就要和别人多讨论交流。我就是在这样的交流中成长的，这也主要得益于刘老师小组浓厚的学术氛围。

论文写作–换位思考，精益求精

如果说论文选题基本决定了论文能投什么档次的会议，那能不能成功发表，却又是另外一回事。论文的写作其实就是对论文工作的一种包装和销售，而审稿人就是顾客。因此，论文必须是非常通俗易懂，避免评审误解论文的内容而拒稿。从表达上来说，尽量使用简单的单词和句子，避免浮夸的单词等等，比如，使用"very novel/fast/high"等说法时就必须很小心，最好有解释或者数据支撑，避免评审觉得某些地方言过其实。同时也要避免写复杂的句式，故弄玄虚，一旦评审读起来不顺畅甚至读不懂的时候，就有可能会开始猜测句子的意思，而这往往是给论文造成不好的结果。从结构上来说，开篇的Introduction可以说是论文重中之重，必须准确明白地告诉评审论文解决了什么问题和问题的重要性，Introduction最重要的功能是说服评审，让评审认同论文将要解决这个问题是有意义的，这样评审才会有兴趣往后阅读。第一次写作的时候容易倾向去强调自己的解决方法是什么样的，怎么样解决了问题的，而忽略了问题本身重要性。可能对于论文作者来说，问题的重要性是显而易见的。但是审稿人的知识背景，对事物的认知都是不一样的，如果不准确地表达出来，就会导致评审出现一些意外的误判。论文评审工作本身就是带有很大主观性的，开篇印象、论文中一个不经意的说法都可能决定着论文的命运。因此论文写作过程中，必须时时刻刻站在评审的角度思考怎么写。

虽然我有切身的经历，但是在论文写作中，往往也很难避免犯错误。因此，尽早完成论文初稿，让导师、合作者甚至同组的小伙伴一起来帮忙阅读，指出论文写作中的问题所在，是一个很实用的方法。这也是在刘老师小组得到反复验证的经验。

执行力与计划

执行力应该是刘老师最看重的能力之一。当论文选定了idea，就应该立马寻找合适的会议，并依照该会议的截稿日期，安排好论文进程。赶上deadline是死命令，这是刘老师小组一贯的风格。如果一次错过截稿日期，就要等待下一次，这是对时间的极大浪费。投完稿的审稿期，可以为下一次投稿继续修改，也可以开始新的论文工作。论文投稿一击即中固然是好的，但是即便没中，也可以收获高水平审稿人的意见，及时地发现并补足论文的缺陷。犹记得第一次论文投稿后，虽然不知道结果如何，但是从此对论文的"害怕"心情彻底消失，因为从此知道了一篇论文是怎么来的，第二次、第三次写论文就变得越来越轻松与自信。

尾　声

论文写作是一门很大的学问，我也只是从自身的经历给大家分享一点点经验。别人说的经验总是别人的，自己悟得的才是自己的，只有自己经历过从写论文到被拒稿、再到被接收的过程，才会有属于自己的科研心得。科研是辛苦的，但也是充满收获的。收获不仅来自自身能力的提升，而且来自科研生活中的各种趣事，比如出国参会，作英文报告，享受当地的风景和美食。科研开启了我很多人生的第一次，也非常感谢刘老师和小组同学的一直以来支持与帮助，最后也希望实验室的小伙伴都能找到合适自己的科研道路与乐趣。

姜炜祥
2014级博士研究生
研究方向：数据中心能耗管理与优化
Email：wxjiang0905@gmail.com

2018 年实验室学术成果一览表

序号	第一作者	文　章
1	马晓静	Xiaojing Ma, Changming Liu, Sixing Cao, and Bin Zhu, "JPEG Decompression in the Homomorphic Encryption Domain", In *Proceedings of the ACM Multimedia Conference (MM)*, October 22-26, 2018, Seoul, Korea, pp. 905-913
2	黄　宏	Hong Huang, Bo Zhao, Hao Zhao, Zhou Zhuang, Zhenxuan Wang, Xiaoming Yao, XinggangWang, Hai Jin, and Xiaoming Fu. "A Cross-Platform Consumer Behavior Analysis of Large-Scale Mobile Shopping Data". In *Proceedings of The World Wide Web Conference (WWW)*, April 23-27, 2018, Lyon, France, pp. 1785-1794
3	于东晓	Dongxiao Yu, Yong Zhang, Yuyao Huang, Hai Jin, Jiguo Yu, and Qiang-Sheng Hua. "Exact Implementation of Abstract MAC Layer via Carrier Sensing". In *Proceedings of the 37th IEEE Conference on Computer Communications (INFOCOM)*, April 15-19, Honolulu, HI, USA, pp. 1196-1204
4	公绪辉	Xuhui Gong, Qiang-Sheng Hua, Lixiang Qian, Dongxiao Yu, and Hai Jin. "Communication-Efficient and Privacy-Preserving Data Aggregation without Trusted Authority". In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, April 15-19, Honolulu, HI, USA, pp. 1250-1258
5	林昌富	Changfu Lin, Jingjing Zhan, Hanhua Chen, Jie Tan, and Hai Jin. "Ares: A High Performance and Fault-Tolerant Distributed Stream Processing System". In *Proceedings of the 26th IEEE International Conference on Network Protocols (ICNP)*, September 25-27, 2018, Cambridge, UK, pp. 176-186
6	吴　松	Song Wu, Mi Liu, Shadi Ibrahim, Hai Jin, Lin Gu, Fei Chen, and Zhiyi Liu, “TurboStream: Towards Low-Latency Data Stream Processing”. In *Proceedings of the 38th IEEE International Conference on Distributed Computing Systems (ICDCS)*, July 2-5 2018, Vienna, Austria, pp. 983-993
7	金　海	Hai Jin, Xiaohai Dai, and Jiang Xiao. "Towards A Novel Architecture for Enabling Interoperability Amongst Multiple Blockchains". In *Proceedings of the 38th IEEE International Conference on Distributed Computing Systems (ICDCS)*, July 2-5, 2018, Vienna, Austria, pp. 1203-1211
8	牛轶佩	Yipei Niu, Fangming Liu, and Zongpeng Li. "Load Balancing across Microservices". In *Proceedings of the 37th IEEE International Conference on Computer Communications (INFOCOM)*, April 15-19, 2018, Honolulu, HI, USA, pp. 198-206
9	曾超冰	Chaobing Zeng, Fangming Liu, Shutong Chen, Weixiang Jiang, and Miao Li. "Demystifying the Performance Interference of Co-located Virtual Network Functions". In *Proceedings of the 37th IEEE International Conference on Computer Communications (INFOCOM)*, April 15-19, 2018, Honolulu, HI, USA, pp. 765-773
10	费新财	Xincai Fei, Fangming Liu, Hong Xu, and Hai Jin. "Adaptive VNF Scaling and Flow Routing with Proactive Demand Prediction". In *Proceedings of the 37th IEEE International Conference on Computer Communications (INFOCOM)*, April 15-19, 2018, Honolulu, HI, USA, pp. 486-494.
11	李肖瑶	Xiaoyao Li, Xiuxiu Wang, Fangming Liu, and Hong Xu. "DHL: Enabling Flexible Software Network Functions with FPGA Acceleration". In *Proceedings of the 38th IEEE International Conference on Distributed Computing Systems (ICDCS)*, July 2-5, 2018, Vienna, Austria, pp. 1-11

序号	第一作者	文　章
12	姜炜祥	Weixiang Jiang, Shaolei Ren, Fangming Liu, and Hai Jin. "Non-IT Energy Accounting in Virtualized Datacenter". In *Proceedings of the 38th IEEE International Conference on Distributed Computing Systems (ICDCS)*, July 2-5, 2018, Vienna, Austria, pp. 300-310
13	姚鹏程	Pengcheng Yao, Long Zheng, Xiaofei Liao, Hai Jin, and Bingsheng He. "An Efficient Graph Accelerator with Parallel Data Conflict Management". In *Proceedings of the 27th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, November 1-4, 2018, Limassol, Cyprus, pp. 8:1-8:12
14	李　珍	Zhen Li, Deqing Zou, Shouhuai Xu, Xinyu Ou, Hai Jin, Sujuan Wang, Zhijun Deng, Yuyi Zhong. "VulDeePecker: A Deep Learning-Based System for Vulnerability Detection". In *Proceedings of the 25th Annual Network and Distributed System Security Symposium (NDSS),* February 18-21, 2018, San Diego, California, USA
15	郑　龙	Long Zheng, Xiaofei Liao, Hai Jin, Jieshan Zhao, and Qinggang Wang. "Scalable Concurrency Debugging with Distributed Graph Processing". In *Proceedings of the 16th International Symposium on Code Generation and Optimization (CGO)*, February 24-28, 2018, Vienna, Austria, pp. 188-199
16	张　宇	Yu Zhang, Xiaofei Liao, Hai Jin, Lin Gu, Ligang He, Bingsheng He, and Haikun Liu. "CGraph: A Correlations-aware Approach for Efficient Concurrent Iterative Graph Processing". In *Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC)*, July 11-13, 2018, Boston, MA, USA, pp. 441-452
17	郑　龙	Long Zheng, Xiaofei Liao, Hai Jin, Bingsheng He, Jingling Xue, and Haikun Liu. "Towards Concurrency Race Debugging: An Integrated Approach for Constraint Solving and Dynamic Slicing". In Proceedings of *the 27th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, November 1-4, 2018, Limassol, Cyprus, pp. 26:1-26:23
18	刘长鸣	Changming Liu, Deqing Zou, Peng Luo, Bin Zhu, and Hai Jin. "A Heuristic Framework to Detect Concurrency Vulnerabilities". In *Proceedings of Annual Computer Security Applications Conference (ACSAC)*, December 3–7, 2018, San Juan, Puerto Rico, USA
19	袁　斌	Bin Yuan, Deqing Zou, Hai Jin, Laurence T. Yang, and Shui Yu. "A Practical Byzantine-based Approach for Faulty Switch Tolerance in Software-Defined Networks". *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 825-839, 2018
20	黄　宏	Hong Huang, Yuxiao Dong, Jie Tang, Nitesh Chawla, and Xiaoming Fu. "Will Triadic Closure Strengthen Ties in Social Networks?". *ACM Transactions on Knowledge Discovery from Data*, vol. 20, no. 30, pp: 30:1-30:25, 2018
21	石宣化	Xuanhua Shi, Zhigao Zheng, Yongluan Zhou, Hai Jin, Ligang He, Bo Liu, and Qiang-Sheng Hua. "Graph Processing on GPUs: A Survey". *ACM Computing Survey*, vol. 50, no. 6, pp. 81:1-81:35, 2018
22	于东晓	Dongxiao Yu, Yifei Zou, Jiguo Yu, Xiuzhen Cheng, Qiang-Sheng Hua, Hai Jin, and Francis C. M. Lau. "Stable Local Broadcast in Multihop Wireless Networks Under SINR". *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1278-1291, 2018
23	金　海	Hai Jin, Na Wang, Dongxiao Yu, Qiang-Sheng Hua, Xuanhua Shi, and Xia Xie. "Core Maintenance in Dynamic Graphs: A Parallel Approach Based on Matching". *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 11, pp. 2416-2428, 2018
24	金　海	Hai Jin, Bo Liu, Wenbin Jiang, Yang Ma, Xuanhua Shi, Bingsheng He, and Shaofeng Zhao. "Layer-centric Memory Reuse and Data Migration for Extreme-Scale Deep Learning on Many-Core Architectures". *ACM Transactions on Architecture and Code Optimization*, vol. 15, no. 3, pp. 37:1-37:26, 2018

序号	第一作者	文　章
25	徐　鹏	Peng Xu, Shuanghong He, Wei Wang, Willy Susilo, and Hai Jin. "Lightweight Searchable Public-Key Encryption for Cloud-Assisted Wireless Sensor Networks". *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3712-3723, 2018
26	陈汉华	Hanhua Chen, and Hai Jin. "Efficient Keyword Searching in Large-Scale Social Network Service". *IEEE Transactions on Services Computing*, vol.11, no. 5, pp. 810-820, 2018
27	王新猴	Xinhou Wang, Kezhi Wang, Song Wu, Sheng Di, Hai Jin, Kun Yang, and Shumao Ou. "Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network". *IEEE Transactions on Parallel and Distributed Systems*, vo. 29, no. 11, pp. 2429-2445, 2018
28	陈姝彤	Shutong Chen, Zhi Zhou, Fangming Liu, Zongpeng Li, and Shaolei Ren. "CloudHeat: An Efficient Online Market Mechanism for Datacenter Heat Harvesting". *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 3, no. 3, pp. 11:1-11:31, 2018
29	郑　龙	Long Zheng, Xiaofei Liao, and Hai Jin. "Efficient and Scalable Graph Parallel Processing With Symbolic Execution". *ACM Transactions on Architecture and Code Optimization*, vol 15, no 1, pp. 3:1-3:25, 2018
30	张　宇	Yu Zhang, Xiaofei Liao, Hai Jin, Lin Gu, and Bingbing Zhou. "FBSGraph: Accelerating Asynchronous Graph Processing via Forward and Backward Sweeping". *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 5, pp. 895-907, 2018
31	张　宇	Yu Zhang, Xiaofei Liao, Xiang Shi, Hai Jin, and Bingsheng He. "Efficient Disk-Based Directed Graph Processing: A Strongly Connected Component Approach". *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 4, pp. 830-842, 2018
32	廖小飞	Xiaofei Liao, Minhao Lin, Long Zheng, Hai Jin, and Zhiyuan Shao. "Scalable Data Race Detection for Lock-Intensive Programs with Pending Period Representation". *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 11, pp. 2599-2612, 2018
33	石宣化	Xuanhua Shi, Xuan Luo, Junling Liang, Peng Zhao, Sheng Di, Bingsheng He, and Hai Jin, "Frog: Asynchronous Graph Processing on GPU with Hybrid Coloring Model". *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 1, pp. 29-42, 2018
34	金　海	Hai Jin, Benxi Liu, Yajuan Du, and Deqing Zhou. "BoundShield: Comprehensive Mitigation for Memory Disclosure Attacks via Secret Region Isolation". *IEEE Access*, vol. 6, pp. 36341-26353, 2018
35	邹德清	Deqing Zou, Yu Lu, Bin Yuan, Haoyu Chen, and Hai Jin. "A Fine-Grained Multi-Tenant Permission Management Framework for SDN and NFV". *IEEE Access*, vol. 6, pp. 25562-25572, 2018
36	邹德清	Deqing Zou, Zirong Huang, Bin Yuan, Haoyu Chen, and Hai Jin. "Solving Anomalies in NFV-SDN Based Service Function Chaining Composition for IoT Network". *IEEE Access*, vol. 6, pp. 62286-62295, 2018
37	赵　峰	Feng Zhao, Zeliang Tian, and Hai Jin. "Entity-based Language Model Smoothing Approach for Smart Search". *IEEE Access*, vol. 6, pp. 9991-10002, 2018
38	羌卫中	Weizhong Qiang, Jiawei Yang, Hai Jin, and Xuanhua Shi. "PrivGuard: Protecting Sensitive Kernel Data From Privilege Escalation Attacks". *IEEE Access*, vol. 6, pp. 46584-46594, 2018
39	吴　松	Song Wu, Yusheng Yi, Jiang Xiao, Hai Jin, and Mao Ye. "A Large-Scale Study of I/O Workload's Impact on Disk Failure". *IEEE Access*, vol. 6, no. 1, pp. 47385-47396, 2018
40	代炜琦	Weiqi Dai, Jun Deng, Qinyuan Wang, Changze Cui, Deqing Zou, and Hai Jin. "SBLWT: A Secure Blockchain Lightweight Wallet Based on Trustzone". *IEEE Access*, vol. 6, pp. 40638-40648, 2018

序号	第一作者	文　章
41	段卓辉	Zhuohui Duan, Haikun Liu, Xiaofei Liao, and Hai Jin. "HME: A Lightweight Emulator for Hybrid Memory". In *Proceedings of The 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 19-23, 2018, Dresden, 2018, pp. 1375-1380
42	肖　江	Jiang Xiao, Zhuang Xiong, Song Wu, Yusheng Yi, Hai Jin, and Kan Hu. "Disk Failure Prediction in Data Centers via Online Learning". In *Proceedings of the 47th International Conference on Parallel Processing (ICPP)*, August 13-16, 2018, Eugene, Oregon, USA, pp. 35:1-35:10
43	吴　松	Song Wu, Zhiyi Liu, Shadi Ibrahim, Lin Gu, Hai Jin, Fei Chen: Dual-Paradigm Stream Processing. In *Proceedings of the 47th International Conference on Parallel Processing (ICPP)*, August 13-16, 2018, Eugene, Oregon, USA, pp. 83:1-83:10
44	丁晓峰	Xiaofeng Ding, Xiaodong Zhang, Zhifeng Bao, Hai Jin: Privacy-Preserving Triangle Counting in Large Graphs. In *Proceedings of The 27th ACM International Conference on Information and Knowledge Management (CIKM)*, October 22-26, 2018, Lingotto, Turin, Italy, pp. 1283-1292
45	金　海	Hai Jin, Saqib Qamar, Ran Zheng, and Parvez Ahmad. "Single Binding of Data and Model Parallelisms to Parallelize Convolutional Neural Networks through Multiple Machines". *Journal of Intelligent and Fuzzy Systems*, vol. 35, no. 5, pp. 5449-5466, 2018
46	代炜琦	Weiqi Dai, Pengfei Wan, Weizhong Qiang, Laurence T. Yang, Deqing Zou, Hai Jin, Shouhuai Xu, and Zirong Huang. "TNGuard: Securing IoT Oriented Tenant Networks Based on SDN". *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1411-1423, 2018
47	吴　松	Song Wu, Chao Mei, Hai Jin, and Duoqiang Wang, "Android Unikernel: Gearing Mobile Code Offloading Towards Edge Computing". *Future Generation Computer Systems*, vol. 86, pp. 694-703, 2018
48	胡　琼	Qiong Hu, Hanhua Chen, Hai Jin, Chen Tian, Aobing Sun, and Tongkai Ji. "IBCube: An Economical and Incremental Datacenter Network". *International Journal of Web Service Research*, vol.15, no.1, pp. 27-46, 2018
49	代炜琦	Weiqi Dai, Yukun Du, Hai Jin, Weizhong Qiang, Deqing Zou, Shouhuai Xu, and Zhongze Liu. "RollSec: Automatically Secure Software States Against General Rollback". *International Journal of Parallel Programming*, vol. 46, no. 4, pp. 788-805, 2018
50	羌卫中	Weizhong Qiang, Shizhen Wang, and Hai Jin. "Fine-Grained Control-Flow Integrity Based on Points-to Analysis for CPS". *Security and Communication Networks*, vol. 2018, pp. 3130652:1-3130652:11, 2018
51	姚琼杰	Qiongjie Yao, Xiaofei Liao, and Hai Jin. "Training Deep Neural Network on Multiple GPUs with A Model Averaging Method". *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp 1012–1021, 2018
52	余　辰	Chen Yu, Qinmin Hong, Dezhong Yao, and Hai Jin. "Tensor-based User Trajectory Mining". *Computer Systems: Science & Engineering*, vol. 33, no. 2, pp. 87-94, 2018
53	罗　鹏	Peng Luo, Deqing Zou, Hai Jin, Yajuan Du, Long Zheng, and Jinan Shen. "DigHR: Precise Dynamic Detection of Hidden Races with Weak Causal Relation Analysis". *The Journal of Supercomputing*, vol. 74, no. 6, pp. 2684-2704, 2018
54	陈　勇	Yong Chen, Hai Jin, Ran Zheng, Yuandong Liu, Wei Wang. "A Hybrid CPU-GPU Multifrontal Optimizing Method in Sparse Cholesky Factorization". *Signal Processing Systems*, vol. 90, no. 1, pp. 53-67, 2018

说明：文章按Top80、Transaction、CCF B类会议和SCI期刊排序。

同态加密域下的JPEG解压缩

马晓静，刘长鸣

文章发表在ACM Multimedia ‘18上。该会议2018年接受到的有效投稿数为757篇，其中64篇被接收为长文口头报告，144篇被接收为长文海报展示。该会议为CCF A类会议，是多媒体领域最重要的会议之一，专注各种多媒体相关理论和技术，从高效压缩到多媒体内容理解和计算等。

内容概述

以图片为代表的多媒体信息占到了当前互联网信息传输和存储信息总量的70%左右，多媒体信息所带来的隐私问题一直亟待解决。传统的对称加密虽能保证信息不被泄露，但也使得被加密的信息完全丧失被再次利用的价值。另一方面，同态加密使得信息被加密之后还能被计算，加之目前对于多媒体信息的各种处理方式层出不穷，故将同态加密引入多媒体信息是一个理想的即保护了多媒体信息，又使得其能被进一步处理的方法。但同态加密本身具有密文膨胀效应，加之多媒体信息不经压缩其大小也过于巨大，不便于网络上传输，为了解决这个问题，本文提出了一种同态域下的JPEG解压缩算法，使得网络上传输压缩后的图片，传输之后再在同态域进行解压缩称为可能。大大降低了需要传输的信息量。

系统总体框架见图1。图片先在同态加密之前，进行SIMD打包，这样使得一张图片的ECS能同时处理。打包完成过后的同态域中，本文将JPEG解压缩。原始的JPEG在熵解码阶段，除了输出直流系数是独立解码，其余的交流系数都依赖于前一轮的解码结果，但同态域下内容都是加密的，故没法查看上一轮的解码结果，为了解决这个问题，本文将这个上下文相关的算法转化为一个上下文无关的算法，即每次遍历码字所有可能的情况，输出唯一匹配的码字。

在反DCT变化和反量化的过程中，本文引入了HEVC的反DCT算法，加快了计算的效率，最终通过移位操作保证像素值在有效范围内。

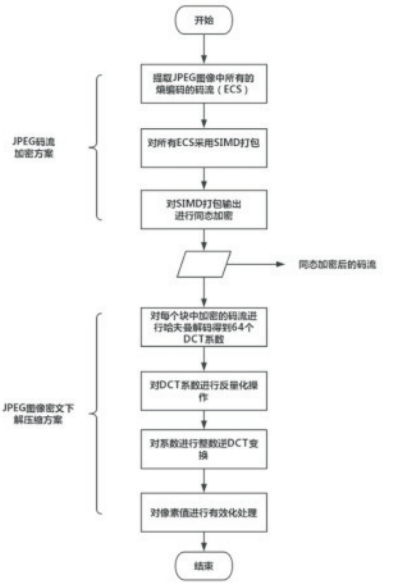


Figure 1 系统框架

通过测试，我们发现解出的系数个数到32 左右时，图片的psnr值已经足够高，满足一般的需求，在这个情况下，同态域的解密对于一个一般大小的图片，需要2000秒左右的时间。

详细内容参见：

Xiaojing Ma, Changming Liu, Sixing Cao, and Bin B. Zhu. 2018. JPEG Decompression in the Homomorphic Encryption Domain. In Proceedings of the 26th ACM international conference on Multimedia (MM '18). ACM, New York, NY, USA, 905-913.
DOI: https://doi.org/10.1145/3240508.3240672

刘长鸣
硕 士
研究方向：多媒体同态加密
Email：liuchangming@hust.edu.cn

大规模用户的跨平台购物行为研究

黄　宏

文章发表在Proceedings of the 2018 World Wide Web Conference上。该会议是中国计算机学会推荐国际学术会议交叉研究领域A类会议。2018年会议于4月23日至27号在法国里昂召开，本届会议共收到1172篇投稿，论文录用率为15%。会议主要关注万维网各方面的研究。

内容概述：

电子商务时代，用户的网上购物行为一直是企业、研究者等广泛关注的研究方向。然而，由于数据的限制，很多研究仅仅只限于单一平台上用户购物行为的研究，难以捕获用户在电子平台上真正的浏览与购买行为。在本文中，我们与中国电信合作，分析了140万用户的浏览日志，从四个维度对用户的跨平台购物行为进行了研究，即用户的地理位置属性（用户所处的功能区），用户的社会经济地位，用户的喜好和用户在购物上花费的精力。我们发现：1）尽管购物平台多样化，但是用户会习惯于他们常去的那些平台浏览及购物；2）大多数用户会在短时间内做出购买的决定，这个时间长度常常不到半小时；3）用户更倾向于在下班时间在住宅区进行购物；4）中产阶级喜欢网上购物，但是低产阶级和高产阶级一旦选择了网上浏览，他们成交的概率更高。

多平台用户购物行为见图1。本文在对用户跨平台购物行为分析的基础上，提出了用户下一次购买预测的通用性框架。本文的研究对电商对用户的研究，产品的推广等提供了一个很好的指导。

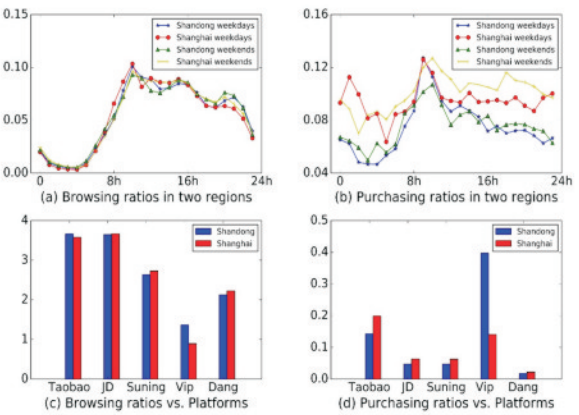


图1　多平台用户购物行为概览

详细内容参见：

Hong Huang, Bo Zhao, Hao Zhao, Zhou Zhuang, ZhenxuanWang, Xiaoming Yao, XinggangWang, Hai Jin, and Xiaoming Fu. 2018. A Cross-Platform Consumer Behavior Analysis of Large-Scale Mobile Shopping Data. Proceedings of the 2018 World Wide Web Conference, Pages 1785-1794. April 23-27, 2018, Lyon, France. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186169

黄　宏
博士
研究方向：社会计算，大数据分析与挖掘
Email：honghuang@hust.edu.cn

基于载波监听精确实现抽象媒体访问控制层

黄钰瑶

文章发表在第37届IEEE国际计算机通信会议（INFOCOM）上。该会议是IEEE在通信网络领域的旗舰性会议，由IEEE Communications Society举办。本届INFOCOM共收到来自世界各地的1606篇论文投稿，有309篇被接受，接受率仅为19.2%。

内容概述：

抽象媒体访问控制层（absMAC）层由Kuhn等人提出。absMAC层提供了可靠的本地广播通信，具有根据抽象延迟函数的时间保证，使得可以根据这些函数设计高级算法，而与特定的信道行为无关。absMAC层的实现是通过定义具体信道行为的特定通信模型为本地广播通信原语设计分布式算法，目标是最小化抽象延迟函数的时间界限。 Halldórsson等相关论文已经表明，在标准物理干扰（SINR）模型中，不能有效的精确实现absMAC层。本文中我们采用无线设备常见的载波监听，在SINR模型中设计出了高效的精确实现算法。

我们的具体工作如下：（1）我们提出了一种用于GLB（一般局部广播）的随机分布式算法，该算法是渐近最优。我们的算法表明，确定本地广播消息的复杂性是不同消息的数量，而不是像先前本地广播结果中所说的是由于节点周围的竞争；（2）使用GLB算法作为子程序，我们提出了第一个精确实现absMAC层的算法。该算法为absMAC层中定义的Acknowledgement和Progress延迟函数提供渐近最优的边界；（3）基于精确实现absMAC层的算法，给出了解决几个基本问题的更快算法，包括邻居发现，一致性，多消息广播（MMB）和单消息广播（SMB）。

除了实现absMAC层之外，GLB算法可以处理更多的通信场景。我们的算法可以用于许多新的更快的算法来解决SINR模型中的其它问题。仿真结果表明我们提出的算法在现实中也表现良好。

详细内容参见：

Dongxiao Yu, Yong Zhang, Yuyao Huang, Hai Jin, Jiguo Yu, Qiang-Sheng Hua. Exact Implementation of Abstract MAC Layer via Carrier Sensing. In Proc. the 37th International Conference on Computer Communications (INFOCOM 2018), April 15-19, 2018, 1250-1258, Honolulu, Hawaii, USA.

黄钰瑶
硕士研究生
研究方向：无线网络分布式计算
Email：m201672844@hust.edu.cn

没有可信任第三方的具有高效通信和隐私保护的数据聚合算法

公绪辉

文章发表在第37届IEEE国际计算机通信会议（INFOCOM）上。该会议是IEEE在通信网络领域的旗舰性会议，由IEEE Communications Society举办。本届INFOCOM共收到来自世界各地的1606篇论文投稿，有309篇被接受，接受率仅为19.2%。

内容概述：

关于隐私保护数据聚合问题已经广泛被相关科研人员进行研究。这类问题简单表述为：将参与者的隐私数据安全的聚合到一个不可信的协调节点，同时没有泄露参与者的私密数据。对于这类问题，大部分已有相关工作都需要可信任的第三方。通过安全信道，可信任的第三方分配秘钥以及其它信息给每个参与者。根据获得秘钥，每个参与者加密数据，从而保护数据隐私。正如一些相关文献所提及的，通过可信任的第三方解决隐私保护数据聚合问题不切实际。因为在很多实际场景中，找到一个这样的可信任的第三方非常困难。有一些工作是结合公钥加密方案或者安全多方计算设计相关解决算法。这导致了这些算法通信复杂度和计算复杂度非常大。基于上述原因，我们提出一种不需要可信任第三方参与的协议，可以计算任意函数。为了降低通信复杂度和计算复杂度，每个参与者产生唯一序列号，并根据唯一序列号发送相关密文。因此我们问题的难点在于：在没有可信任第三方的情况下，每个参与者需要产生秘钥和唯一序列号。对于秘钥问题，我们采用DH(Diffie-Hellman)秘钥交换协议。通过运行此协议，每个参与者获得两个秘钥集合，再进一步产生加密函数。对于唯一序列号问题：我们设计了随机抽样与划分技术。具体来说，每个参与者在给定区间随机选取一个随机数，然后结合划分技术降低通信复杂度。最后每个参与者秘密获得各自随机数在全部参与者的随机数中按照升序排列所在的位次而不知道其余参与者的位次。通过严格的安全性分析，我们的协议满足（n-k）源匿名（假设系统中有n个参与者，其中有k个参与者互相同谋）。通过严格的理论推导，我们协议通信复杂度是$O((n(\frac{\log n}{\log\log n}))^2)$。对比已有的相关工作（需要可信任的第三方参与），我们协议的通信复杂度仅仅高$(\frac{\log n}{\log\log n})^2$ 2倍。

详细内容参见：

Xuhui Gong, Qiang-Sheng Hua, Lixiang Qian, Dongxiao Yu, Hai Jin. Communication-Efficient and Privacy-Preserving Data Aggregation without Trusted Authority. In Proc. the 37th International Conference on Computer Communications (INFOCOM 2018), April 15-19, 2018, 1250-1258, Honolulu, Hawaii, USA.

公绪辉
博士研究生
研究方向：低通信复杂度分布式隐私保护算法
Email：d201577742@hust.edu.cn

分布式流处理容错系统Ares

林昌富

文章发表在IEEE International Conference on Network Protocols (ICNP) 上。IEEE ICNP是计算机网络领域享有盛誉的顶级国际学术会议之一。今年ICNP共收到投稿196篇，录用35篇，录用率仅约17.9%。

内容概述：

为了满足实时流处理应用的低延迟和高容错的需求，分布式流处理系统需要保障系统能够以低延迟的方式处理海量的实时数据流，同时当系统中的节点发生故障时，系统能够在较短的时间内恢复正常。现有的分布式流处理系统通常通过优化任务调度来实现海量实时数据流的低延迟处理。典型的分布式流处理系统任务调度策略通过将流处理应用拓扑中的上下流任务进行绑定并分配到同一节点的方式，降低上下流任务之间的数据传输时间，从而保障低延迟处理性能。然而，该任务调度策略忽视了上下流任务之间的依赖关系对系统容错性能的影响。即当系统中的节点发生故障时，由于失效节点上存在大量的上下流任务，这些任务之间的依赖关系会导致在任务恢复期间发生任务级联等待的现象，从而使得系统无法在较短的时间内恢复正常。

本研究指出实现一个低延迟和高容错的分布式流处理系统的关键在于能否在任务调度过程中充分挖掘流处理应用拓扑中上下流任务之间的依赖关系。为此，研发了分布式流处理容错系统Ares。Ares在系统任务调度过程中兼顾上下流任务之间的依赖关系对处理性能和容错性能两者的影响，并提出了一种容错调度问题来同时优化处理延迟和恢复时间。为了最大化系统效用，Ares利用单一任务的最佳放置策略往往取决于该任务的上下流任务的放置策略的特性，设计了一种基于最佳应对动态算法Nirvana来自动优化任务放置策略。相对于目前最新流处理系统，Ares将系统总体吞吐率提高了3.6倍，将平均处理延迟降低了50.2%，将平均恢复时间降低了52.5%。

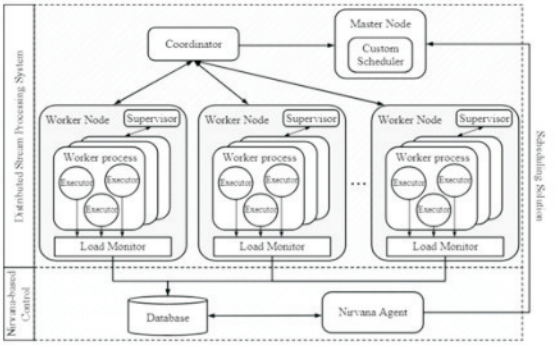


图1 Ares系统框架

图1显示了Ares的系统架构。Ares的架构在传统分布式流处理系统架构的基础上，增加了一个负载监视器和一个Nirvana控制器。负载监视器负责实时监控所有计算节点上的任务的资源使用情况。Nirvana控制器根据当前系统的资源使用情况，通过Nirvana算法计算得到满足低延迟和高容错需求的最佳任务调度方案。

详细内容参见：

Changfu Lin, Jingjing Zhan, Hanhua Chen, Jie Tan, Hai Jin: Ares: A High Performance and Fault-Tolerant Distributed Stream Processing System. IEEE ICNP 2018, pp. 176-186.

林昌富
2015级博士
研究方向：分布式流处理系统
Email：lcf@hust.edu.cn

TurboStream: Towards Low-Latency Data Stream Processing

柳　密

文章发表在38th IEEE International Conference on Distributed Computing Systems(ICDCS'18)。该会议主要关注分布式系统与软件、分布式算法以及大数据系统与分析等方面的研究。

内容概述：

数据流处理（DSP，Data Stream Processing）应用通常是建模为有向无环图：操作符与它们之间的数据流。尽管现在的数据中心的节点都是通过Gigabit以太网或者InfiniBand 连接，但是跨越进程的操作符间的通信开销依然远远高于一个进程内部的操作符间的通信开销。在实践中发现，操作符间的延时占到DSP 应用的总处理延时的86。以上。因此，操作符间的通信对DSP应用的总处理延时有重大的影响。改进操作符间的通信是降低DSP应用处理延时的一个重要因素。

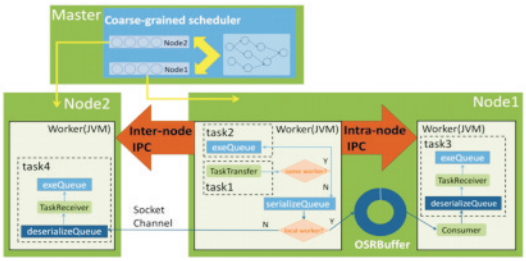


图1.1 Turbo Stream架构图

通过对当前流计算系统的深入分析，提出了TurboStream的设计和实现，系统的设计图如图1.2所示。TurboStream是专为解决操作符间通信所带来的高延时的问题而设计的新型DSP系统。为了降低操作符间的延时，引入了两个功能上互补的组件：（1）改进的IPC框架。它内部集成了一个OSRBuffer。OSRBuffer是一个专门设计的面向DSP的堆外环形缓冲区。OSRBuffer 主要用于减少在一个节点内部进程间通信的开销。传统的节点内进程间通信需要借助套接字通过虚拟设备回环（loopback device）传送给下游任务，这一过程需要多次的内存数据拷贝以及不必要的排队，增加了通信延迟，OSRBuffer通过在两个进程间共享一个环形缓冲区，并以内存映射的方式传输消息，可以减少不必要的内存拷贝和排队，大大提升节点内进程间通信的性能。（2）粗粒度调度器。该调度器可以在调度之前，根据操作符间的数据依赖关系或者通信量来对操作符进行整合，能够将通信密集型的上下游操作符整合在一起作为一个整体进行调度。这样就可以避免通信密集型的上下游操作符频繁的跨越节点进行通信，从而降低系统延迟。

鉴于JStorm在工业界的广泛应用以及其在低延时方面的优异表现，TurboStream的原型实现是基于JStorm。通过实验证明，改进后的IPC框架将节点内部的IPC的端到端的延时降低超过45.94%。此外，与JStorm相比，TurboStream 将DSP的平均处理延时降低了83.23%。

详细内容参见：

Wu Song, Mi Liu, Shadi Ibrahim, Hai Jin, Lin Gu, Fei Chen and Zhiyi Liu. "TurboStream: Towards Low-Latency Data Stream Processing." 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) (2018): 983-993.

柳　密
硕士研究生
研究方向：云计算与分布式系统
Email：liumihuster@gmail.com

面向跨链交互的区块链新型架构设计

戴小海

文章发表在IEEE 38th International Conference on Distributed Computing Systems上。该会议每年举办一次，本次会议全球378篇投稿中录用78篇，主要关注云计算和数据中心，分布式大数据系统和分析，分布式操作系统和中间件等方向的研究。

内容概述：

近年来，区块链技术经历了高速发展，越来越多的区块链系统得以部署和应用。但现有区块链系统上的数据流通性差，不同区块链上的数据形成一个个数据孤岛，限制了区块链技术的进一步发展。我们在本篇文章中旨在研究区块链的链间交互技术，为不同区块链系统上的数据建立高效可信的连接。

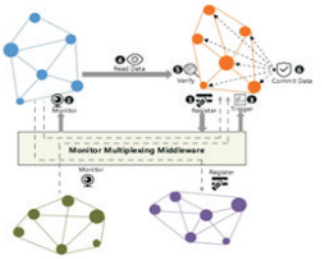


图1 “多路复用”的被动交互方法图

首先，我们对“区块链链间交互”进行了严格定义，在此基础上提出了区块链链间交互的一系列关键挑战。类比传统的网络分层协议，我们对区块链的系统结构进行了分层设计，并探讨从不同层次解决这些挑战的可能思路。1）介于当前尚没有关于“区块链链间交互”的严格定义，我们综合考虑安全性、高效性等多方面的因素，尝试给出“区块链链间交互”的严格定义。在此基础上，我们提出了五个关键挑战，并对每个挑战进行了详细描述。2）在研究现状的调研方面，我们对现有的主流区块链链间交互技术进行了深入研究和总结，将其分为了“主动模式”和“被动模式”两大类，并分析了各自的优缺点。3）受传统网络分层协议的启发，我们对区块链的系统架构进行了分层设计，包括数据层、网络层、共识层、合约层和应用层。从不同层次，我们尝试探讨了解决以上关键挑战的可能思路。4）为解决“兼容性”的挑战，我们尝试从数据层着手，统一底层的交易格式，使一条链上的交易可以轻易地发往另一条链。对于已部署的区块链系统，我们尝试提供一套“交易翻译”工具，将统一的交易格式翻译成所需的交易格式。5）为解决“高效性”的挑战，我们尝试从网络层着手。针对“被动模式”的链间交互所产生出的轮询开销，我们设计了一种“多路复用”的被动交互方法（如图1所示），并通过实验验证了该方法的高效性。6）为解决“安全性”的挑战，我们尝试从共识层着手。考虑到单链上的共识层用于维护不同节点上状态的一致性，而不同链上保存的数据状态必然是不同的，我们将链间的“共识协议”重新定义为一种特殊的“验证协议”，以验证不同链上是否存在矛盾数据，从而保证链间交互的安全性。

详细内容参见：

Jin, Hai, Xiaohai Dai, and Jiang Xiao. "Towards a Novel Architecture for Enabling Interoperability amongst Multiple Blockchains.", in Proccedings of 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018, pp. 1203-1211

戴小海
博士研究生
研究方向：区块链的可扩展性研究
Email：seafooler@hust.edu.cn

微服务的负载均衡策略

牛轶佩，刘方明

文章“Load Balancing across Microservices”被IEEE INFOCOM 2018收录。IEEE INFOCOM是计算机网络与通信领域的旗舰级会议，在国际上享有盛誉和重要影响力。同时也是中国计算机学会（CCF）推荐的计算机网络领域最高级别的三大A类国际学术会议之一。

内容概述：

传统单体应用设计架构具有模块耦合度高、运维复杂等缺点，于是催生出新兴的微服务设计架构，考虑到应用功能模块的分化，一个微服务应用的请求往往需要被多个特定的微服务实例处理，从而构成一条微服务链，而不同的微服务链则有可能需要相同的微服务实例为其提供服务。因此如何面向微服务设计负载均衡策略成为一个急需解决的问题。

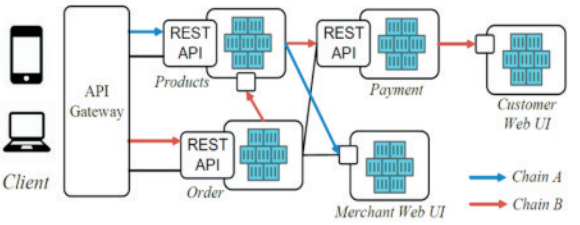


图1 微服务应用架构图

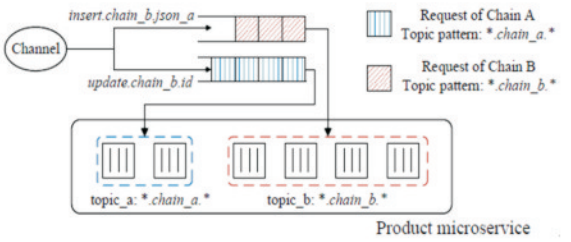


图 2 基于消息队列的负载均衡策略实现图

为了解决以上问题，本文提出了一种面向微服务链的负载均衡策略。该策略利用消息队列技术极大的简化了应用运维，在降低请求响应时延的同时满足不同微服务链的QoS。首先将微服务负载均衡问题建模成为一个非合作博弈问题，进而运用纳什均衡为服务链制定微服务实例分配策略，在满足微服务链QoS的同时最小化请求响应时延。

大量实际数据驱动的测试结果表明，相较已有的负载均衡策略，本文提出的策略能够将请求响应时延降低至少13%的同时满足不同微服务链的QoS，有效缓解微服务链竞争带来的性能。

详细内容参见：

Yipei Niu, Fangming Liu*, and Zongpeng Li, "Load Balancing across Microservices", in Proc. of IEEE INFOCOM, 15-19 April 2018, Honolulu, HI, USA.

牛轶佩

2016级博士研究生

导　　师：刘方明 教授

研究方向：混合云

Email：newypei@hust.edu.cn

刘方明

华中科技大学教授、博导

研究方向：云计算与数据中心、软件定义的网络SDN与虚拟化技术、绿色计算与通移动互联网。国家优秀青年科学基金获得者，国家高层次人才特殊支持计划（中组部“万人计划”）青年拔尖人才。

Email：fmliu@hust.edu.cn

虚拟网络功能性能干扰的测量研究

曾超冰，刘方明，陈姝彤，姜炜祥，李　苗

文章“Demystifying the Performance Interference of Co-located Virtual Network Functions”被IEEE INFOCOM 2018收录。IEEE INFOCOM是计算机网络与通信领域的旗舰级会议，在国际上享有盛誉和重要影响力。同时也是中国计算机学会（CCF）推荐的计算机网络领域最高级别的三大A类国际学术会议之一。

内容概述：

网络功能虚拟化将网络功能从专用硬件解耦到通用硬件上，使得虚拟网络功能可以运行在网络任意位置的虚拟机或容器上。然而，这种使用模式会给网络运营商带来新的挑战。为了提高资源使用率及减少电力消耗，网络运营商们通常会尽可能把虚拟网络功能集中部署在尽量少的物理服务器上。虽然虚拟化技术可以提供一定程度的性能隔离，但是对同一物理设施的共享仍会给虚拟网络功能带来严重的性能干扰。在这篇文章，我们对主流的虚拟网络功能应用进行详细分类，从资源角度分析性能干扰现象的成因，并提出优化建议。

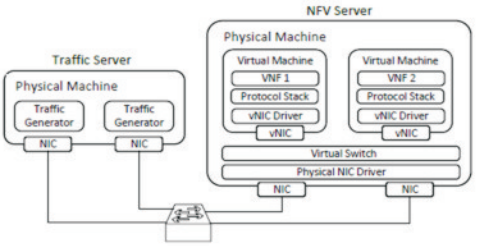


图1 测量平台框架

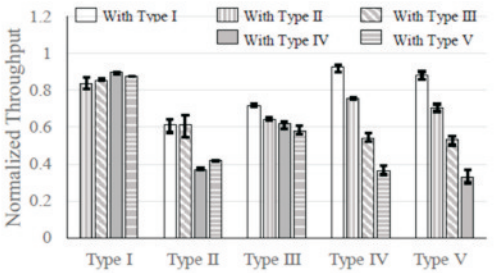


图2 各类虚拟网络功能归一化后的吞吐量

测量平台见图1，包括通用服务器（作为流量产生器）和通用服务器（作为NFV服务器）。该测量平台通过流量产生器去产生不同类型的流，发送给NFV服务器上的虚拟网络功能进行处理。在这个过程中，我们利用监控程序去监控不同场景下虚拟网络功能的性能（吞吐量、延迟等）及资源使用情况（CPU，Cache，内存等）。图2为各类虚拟网络功能与其他类别虚拟网络共同运行时以单独运行无干扰情况下的吞吐量为标准归一化后的吞吐量情况。测量表明，虚拟网络功能的性能干扰现象是普遍存在的，造成12.36%到50.3%的吞吐量损失，对网络I/O带宽的竞争是性能干扰现象的重要原因。

详细内容参见：

Chaobing Zeng, Fangming Liu, Shutong Chen, Weixiang Jiang, and Miao Li. “Demystifying the Performance Interference of Co-located Virtual Network Functions”. In proceedings of the 37 th IEEE International Conference on Computer Communications (INFOCOM), April 15-19, 2018, Honolulu, HI, USA, pp. 765-773.

曾超冰

2016级硕士研究生

导　　师：刘方明　教授

研究方向：网络功能虚拟化

Email：bingzizeng@gmail.com

刘方明

华中科技大学教授、博导

研究方向：云计算与数据中心、软件定义的网络SDN与虚拟化技术、绿色计算与通移动互联网。国家优秀青年科学基金获得者，国家高层次人才特殊支持计划（中组部“万人计划”）青年拔尖人才。

Email：fmliu@hust.edu.cn

基于主动预测的自适应虚拟网络功能扩展和流量调度

费新财

文章“Adaptive VNF Scaling and Flow Routing with Proactive Demand Prediction”被IEEE INFOCOM 2018收录。IEEE INFOCOM是计算机网络与通信领域的旗舰级会议，在国际上享有盛誉和重要影响力。同时也是中国计算机学会（CCF）推荐的计算机网络领域最高级别的三大A类国际学术会议之一。

内容概述：

随着网络功能虚拟化（NFV）的不断发展，越来越多的企业用户将他们所依赖的网络功能外包到云中处理。然而，由于请求服务的流量是动态波动的，NFV服务提供商往往无法根据用户的实际需求智能地扩展虚拟网络功能的（VNF）的处理能力，因此如何在云环境中合理利用资源来部署虚拟网络功能并保证服务质量，成为了极具挑战性的问题。

针对该问题，提出了一种联合在线算法：（1）首先利用在线学习的思想，以预测值与实际值之间的差距作为损失函数。在原损失函数难以求解的情况下，采用了合理的代理损失函数；随着训练样本的增多，代理损失函数与原损失函数求出解的损失差距越来越小，即保证了预测服务链请求的有效性；根据预测结果，算法得出所需要的VNF实例并配置不同的处理能力。（2）然后，在线算法再调用另外两个算法，分别解决新VNF实例的分配和流量在新实例之间的调度两个子问题。通过理论分析和基于实际数据的仿真实验，证明所提出的联合在线算法具有良好的性能保证。

在基于实际数据的仿真实验验证了所提预测方法的准确性（如图1）。通过与现有的其他方法（比如online gradient decent，follow the leader）比较，我们进一步证明了所提在线算法的有效性（如图2）。

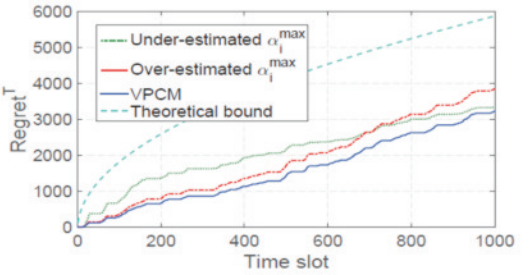


图1

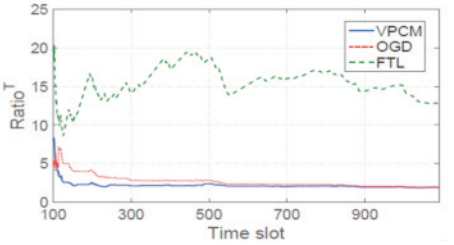


图2

详细内容参见：

Xincai Fei, Fangming Liu, Hong Xu, and Hai Jin, “Adaptive vnf scaling and flow routing with proactive demand prediction,” in IEEE INFOCOM, 2018

费新财
博　士
研究方向：网络功能虚拟化中的资源分配和性能优化
Email：god14fei@hust.edu.cn

基于FPGA的高效能网络平台研究

李肖瑶，王秀秀，刘方明

文章发表在The 38th IEEE International Conference on Distributed Computing Systems (ICDCS 2018)。ICDCS是分布式计算与系统领域享有盛誉和重要影响力的顶级国际学术会议，本节ICDCS在全球378篇投稿中录用78篇论文，录用率仅约20%。

内容概述：

网络功能虚拟化旨在将多种多样的网络功能从昂贵固化的专用网元设备解耦到通用服务器上，以软件方式灵活部署与运行。然而，当前软件网络功能在进行深度包处理时，需要消耗大量的CPU核心和资源才能达到线速度。虽然支持高并发度和可编程性的FPGA具备加速深度包处理的可行性和潜力，但是FPGA中的可编程逻辑十分有限而且成本昂贵，因此若将整个网络功能部署到FPGA上会造成不切实际的资源浪费。此外，当网络功能需要更改时，还需要耗费数小时生成新的FPGA加速程序，阻碍网络功能的快速部署。针对上述挑战，我们提出和实现了基于动态硬件库（Dynamic Hardware Library, DHL）的FPGA-CPU协同设计框架，使得网络功能的浅包处理在CPU中执行，深包处理卸载到FPGA中执行。

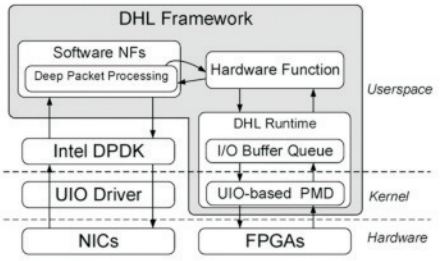


图1 DHL平台总体架构

系统平台架构见图1，该平台架构可分为三个层次，依次为硬件层、内核层、以及用户层。在DHL平台中，网络功能为运行在传统服务器中的软件程序，网络功能程序能够调用硬件函数以代替软件函数进行网络报文的部分处理。其中硬件函数具体形式为运行在FPGA板卡中的硬件加速模块，其用于计算密集的深包处理类操作。该平台通过采用一系列方法提供了一个高吞吐率、低延迟的数据传输层，主要包括用户态I/O轮询驱动、NUMA感知的资源分配、批处理传输、共享的输入缓存队列和私有的输出缓存队列。通过多组实验和多方面的评估，我们证明了DHL平台可以提供与仅FPGA方案相同的吞吐率（40 Gbps），并且只增加了不超过10 μs的延迟。

详细内容参见：

X. Li, X. Wang, F. Liu, and H. Xu, “Dhl: Enabling flexible software network functions with fpga acceleration,” in Proc. of ICDCS, July 2018.

李肖瑶
硕士研究生
研究方向：FPGA网络功能加速
Email：calmisi975@gmail.com

刘方明
华中科技大学教授、博导
研究方向：云计算与数据中心、软件定义的网络SDN与虚拟化技术、绿色计算与通移动互联网。国家优秀青年科学基金获得者，国家高层次人才特殊支持计划（中组部“万人计划”）青年拔尖人才。
Email：fmliu@hust.edu.cn

面向云数据中心非IT设施的能耗计量方法

姜炜祥

文章发表在The 38th IEEE International Conference on Distributed Computing Systems (ICDCS 2018)。ICDCS是分布式计算与系统领域享有盛誉和重要影响力的顶级国际学术会议，本届ICDCS在全球378篇投稿中录用78篇论文，录用率仅约20%。

内容概述：

能耗管理是数据中心实现节能减排和降低成本的重要技术基石。在整个数据中心中，制冷、电力系统等非IT设施能耗占比高达30~50%。然而，由于这些非IT设施由大量承载各种应用负载的IT设备共享和使用，并且IT能耗和非IT能耗之间存在非线性增长的关系，因此难以进行细粒度的非IT能耗测量及有效的能耗管理。论文运用博弈论，将数据中心中非IT设施能耗在虚拟机层面进行细粒度划分的工程问题提炼转化为成本分配问题，从而运用经济学中著名的夏普利值（Shapley Value）方法为数据中心中大量的动态虚拟机的非IT能耗实现公平高效的计量。论文通过对IT能耗和非IT能耗之间的行为模式进行实测与分析发现，非IT能耗和IT能耗之间的关系可以用二次函数进行描述，如图1所示。

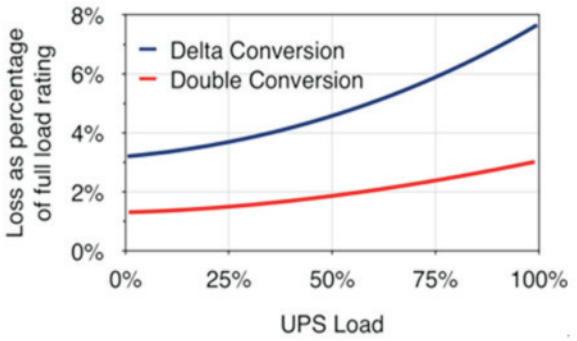


图1 UPS能耗损和IT负载的关系

据此，论文利用非IT设施的能耗特征，设计了高效的降维方法，将计算复杂度大幅降低至O（N），得到与原始夏普利值等价的计算方法，如下所示。

$$\Phi_{ij} = p_i[a_j \sum_{k \in N} p_k + b_j] + \frac{c_j}{|N|}$$

基于真实数据中心能耗数据驱动的实验验证，在引入真实测量误差的情况下，论文所提出的降维方法与理论最优的夏普利值相比，最大误差只有6.97%，能够有效应用于实际数据中心系统。

详细内容参见：

Weixiang Jiang, Shaolei Ren, Fangming Liu, and Hai Jin. Non-IT Energy Accounting in Virtualized Datacenter. In Proc. of IEEE ICDCS, 2-5 July 2018, Vienna, Austria

姜炜祥
2014级博士生
研究方向：数据中心能耗管理与优化
Email：wxjiang0905@gmail.com

基于并行归并的高性能图计算加速器

姚鹏程

文章 “An Efficient Graph Accelerator with Parallel Data Conflict Management”在PACT 2018发表。PACT是ACM SIGARCH，IEEE CS和IFIP共同承办的国际学术会议，主要关注并行体系架构与编译技术的研究，在国内外学术界有着很高的影响。

内容概述：

近年来，图结构因其灵活的表达力被广泛地利用在社交网络分析等重要的现实应用中。然而由于图结构自身的复杂性，图应用的性能难以得到充分的挖掘。具体而言，在图结构的遍历过程中，每个节点往往需要同时接收和处理多个邻居节点的读写请求，从而引发了严重的数据冲突问题。为了保证计算结果的正确性，现有的图计算加速器通常在架构中引入相应的原子结构，通过停滞流水线的方式保证冲突操作间的顺序性。这类原子结构在处理图计算中的原子操作时会产生大量的同步开销，降低加速器约45%的整体性能。

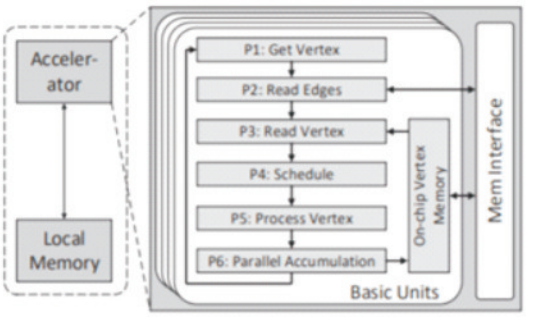


图1 系统框架

论文通过对图应用的计算特性进行观察和分析，发现经典图应用的原子操作往往遵循递增性和单一性，即以任意顺序归并这些原子操作的结果并不会改变最终的计算结果。基于上述发现，论文提出了一个如图1所示的加速器架构，其主要创新点包括：1）设计并行累加器。通过并行累加多个冲突操作的计算结果，优化冲突操作间的顺序性，避免节点数据在流水线间频繁同步；2）节点度数感知机制。通过运行时感知节点度数的动态维护调度与执行机制，保证高、低度数节点的高效执行；3）高吞吐片上缓存。通过图数据预处理与访存乱序化，保证累加器的高效执行。对BFS、WCC和PR等经典图应用和SNAP数据集的测试结果表明，提出的加速器架构的平均吞吐量可以达到2.36 GTEPS。相比于同类图计算加速器，加速比最高可以达到3.14x。

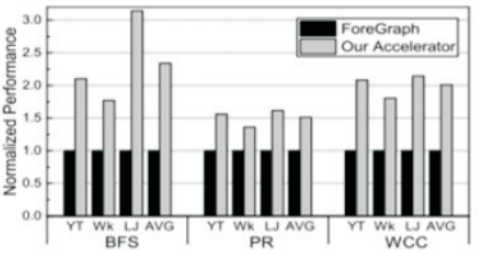


图2 性能提升

详细内容参见：

Pengcheng Yao, Long Zheng, Xiaofei Liao, Hai Jin, Bingsheng He. An Efficient Graph Accelerator with Parallel Data Conflict Management. In Proceedings of International Conference on Parallel Architectures and Compilation Techniques (PACT 2018). 8:1-8:12.

姚鹏程
博士生
研究方向：主要从事图计算、FPGA等领域的研究
Email：pcyao@hust.edu.cn

基于深度学习的漏洞检测系统

李　珍

文章发表在Proceedings of the 25th Annual Network and Distributed System Security Symposium（NDSS 2018）上。NDSS是国际公认的计算机系统安全领域四大顶级学术会议（BIG4）之一，主要关注网络与系统安全。2018年收到331篇投稿，录用71篇，录用率约为21%。

内容概述：

软件漏洞的自动检测是一个重要的研究问题。现有的漏洞静态分析方法存在两个问题：第一，依赖人类专家定义漏洞特征。由于漏洞特征复杂，即使对专家而言也是一个冗长乏味、主观性强、且易出错的工作。第二，现有的漏洞检测方法漏报较高。一个具有高误报的漏洞检测系统是不可用的，而具有高漏报的漏洞检测系统是无用的。理想的漏洞检测系统是同时满足低误报和低漏报，但通常二者很难同时满足，这时更好的处理方法是强调降低漏报，只要误报在可接受的范围内。

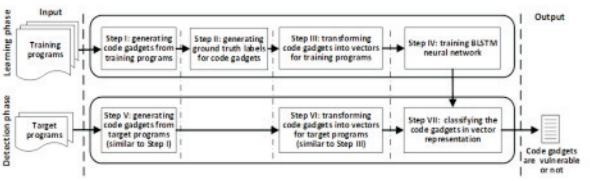


图1　系统结构

为解决上述问题，首次开展基于深度学习的漏洞检测研究，提出了基于深度学习的漏洞检测系统VulDeePecker，结构如图1所示。以候选代码段code gadget为粒度，基于双向长短期记忆网络模型（BLSTM）自动学习生成漏洞模式，在不需人类专家定义特征的前提下，自动检测目标程序是否含有漏洞，并给出漏洞代码的位置。针对能否同时处理多类漏洞、人类经验能否改进有效性、与其他静态检测方法的有效性比较3个方面对VulDeePecker的有效性进行了评价。

实验结果表明，VulDeePecker可以应用到多类漏洞，其有效性与安全相关库/API函数的个数有关；人类经验可用于选择和安全有关的库/API函数，能够改进VulDeePecker的有效性；VulDeePecker比人工定义规则的静态分析工具更有效，比基于代码相似性的漏洞检测方法具有更低的漏报，其有效性受数据量的影响。VulDeePecker在Xen，Seamonkey和Libav 3个软件中检测到4个在美国国家漏洞库NVD中未公布的漏洞，这些漏洞在相应软件的后续版本中进行了默默修补。

详细内容参见：

Zhen Li, Deqing Zou, Shouhuai Xu, Xinyu Ou, Hai Jin, Sujuan Wang, Zhijun Deng, Yuyi Zhong: VulDeePecker: A Deep Learning-Based System for Vulnerability Detection. In Proceedings of the 25th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, February 2018.

李　珍
2014级博士生
研究方向：漏洞检测
Email：lizhen_hust@hust.edu.cn

一种基于分布式图计算的并发调试框架

郑　龙

文章发表在 International Symposium on Code Generation and Optimization(CGO)上，是由IEEE CS和ACM SIGMICRO共同主办的重要国际学术会议，主要关注代码生成和优化等方面的研究，在国内外有很高的影响。

内容概述：

多核处理器催生多线程编程，由于多线程的不确定性，并发程序易于出现并发错误，然而并发错误的重现十分困难。对于较长执行时间的程序轨迹，现有基于约束求解的技术水平与实际可扩展性调试需求之间仍存在差距。因此，我们调查了大规模多线程程序调试的可伸缩性问题，发现，许多并发错误的并发调试可以转化为图遍历问题。因此，我们提出了一种新颖的并行调试框架（GraphDebugger）。在分布式环境中，GraphDebugger通过图并行分析在程序图上实现可扩展的并发分析。首先，我们将动态程序执行转换为程序图，该程序图可用于表示所有可能的线程内和线程间依赖关系。其次，基于生成的程序图，采用提出的check-update-push编程模型，进一步设计了一种调度指导的图并行遍历算法，它可以找到一组最大可行的并发错误触发路径。

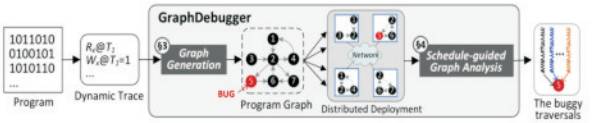


图1 GraphDebugger的框架

我们调试框架的技术基础是分布式图计算系统，它能够将大型复杂任务分解为许多小而简单的任务，以实现更高的调试效率。图1示出GraphDebugger的框架，主要包含如下技术亮点：（1）从动态跟踪中提取信息以构建程序图，该程序图可以遵循基本的图并行抽象，以便与现有的图处理系统可以无缝集成以进行并发调试；（2）基于Check-Update-Push编程模型在一次运行中遍历程序图上的最大可行调度集。

结果表明，GraphDebugger在重现并发错误方面比基于约束求解的并发调试系统效率更高， 特别是对于许多长时间运行的程序，GraphDebugger能够在几分钟内重现并发错误，而CLAP则需要几小时或甚至几天的时间完成，甚至在许多情况下没有返回结果。

详细内容参见：

Long Zheng, Xiaofei Liao, Hai Jin, Jieshan Zhao, and Qinggang Wang. "Scalable Concurrency Debugging with Distributed Graph Processing". In Proceedings of the 16th International Symposium on Code Generation and Optimization (CGO), February 24-28, 2018, Vienna, Austria, pp. 188-199

郑　龙
博　士
研究方向：可重构计算机体系结构及其运行时环境
Email：longzh@hust.edu.cn

关联性感知的并发图处理系统

张　宇

该论文被USENIX Annual Technical Conference 2018收录为Full paper。USENIX是计算机系统领域颇具影响力的组织之一。该组织主办的ODSI, NSDI, FAST, USENIX Security等会议在学术界及业界均享有盛誉。其中USENIX Annual Technical Conference是计算机系统领域的重要国际学术会议，创办于1992年，关注开源系统软件最新进展和实验分析。

内容概述：

随着图计算的快速发展，图计算平台通常需要高效处理大量并发图分析任务。虽然现有解决方案已广泛研究和优化单图分析任务的执行，然而它们由于缓存干扰和内存墙等问题对于并发图分析任务的执行面临高额的数据访问开销，导致系统吞吐量低。我们观察到并发图分析任务的数据访问中存在大量时空关联性。基于此分析观察结果，本文提出了一种关联性感知的并发图处理系统，以充分利用这些数据访问关联性，使并发图分析任务能够高效地处理缓存/内存中共享图结构数据，通过有效降低平均数据访问开销提供更高的系统吞吐量。

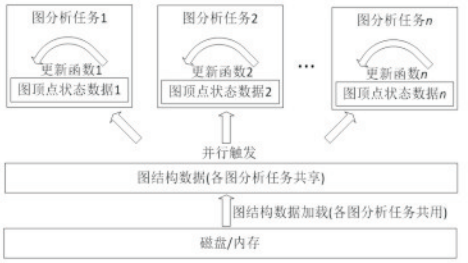


图1. 基于LTP的多任务执行模型解释

具体来说，如图1所示，本系统采用基于LTP（Load，Trigger，Push）的多任务执行模型支持多图分析任务的有效执行。LTP模型将图数据分解为图结构数据和图顶点状态数据。各图分析任务都会拥有一个私有图顶点状态数据集，而图结构数据为各图分析任务所共享。然后，LTP模型将根据各图分析任务的数据访问和处理需要，反复载入各任务共享的待处理的图结构数据，触发各图分析任务根据此图结构数据和各自的更新函数（用于完成各算法目的），更新各自私有的图顶点状态直到收敛。这样，它允许各图分析任务利用它们之间的数据访问空间/时间关联性，充分共享图结构数据及其访问，提高系统吞吐率。在LTP模型中，各图分析任务分别拥有私有空间存储各算法自身图顶点状态数据，而图结构数据作为全局数据被它们所共享。因此，它保证了执行过程的正确性。实验表明，与现有解决方案相比，本方法能显著提高并发图分析任务的吞吐量。

详细内容参见：

Yu Zhang, Xiaofei Liao, Hai Jin, Lin Gu, Ligang He, Bingsheng He, Haikun Liu. CGraph: A Correlations-aware Approach for Efficient Concurrent Iterative Graph Processing. USENIX Annual Technical Conference 441-452, 2018.

张　宇
博士后
研究方向：大数据系统软件、运行时优化
Email：zhang_yu9068@163.com

一种面向数据竞争的并发调试方法

郑　龙

文章发表在Parallel Architectures and Compilation Techniques (PACT)上。该会议每年举办一次，是计算机体系结构方面的重要国际学术会议，主要关注于并行体系结构、编译技术等方面的研究。

内容概述：

若多个线程同时访问同一内存区域，且至少存在一个线程进行更新操作，这时便发生数据竞争。现有关于于数据竞争的工作大都主要集中在数据竞争的检测，即识别出其在代码中发生时的位置，但是上述信息并不足以修复问题，数据竞争的诊断仍是一个开放性问题。

现有的动态竞争检测器大多依赖于程序的实际运行来触发一个竞争调度，但是这通常需要较长时间和次数方可触发一个竞争，甚至可能在有限次数内无法触发。尽管基于约束求解SMT (Satisfiability Modulo Theory)的检测技术无需反复运行程序，通过推导计算的方式获得生成竞争调度，但这些竞争调度也不能保证竞争故障行为的发生，更为糟糕的是，当涉及的变量和线程数较多时，调试效率低下。

我们提出了一种面向数据竞争的并发调试框架RaceDebugger，其集成了约束求解和动态切片技术，以帮助理解数据竞争的根本原因，即触发数据竞争并发错误的最小程序事件集。

图1 RaceDebugger系统框架

RaceDebugger系统总体框架见图1，其工作流程如下：（1）运行程序生成程序动态事件轨迹，变量采用符号化方式进行表示；（2）利用约束求解器求解数据竞争调度，以此作为RaceDebugger的输入；（3）设计一种数据竞争约束求解条件，对上述海量调度集重新进行调度，生成产生并发错误的数据竞争调度，简化调度集；（4）为定位数据竞争根本原因，利用程序动态切片技术，对数据竞争行为无关的事件进行精简，将简化后的子调度作为该数据竞争的根本原因。

对若干大型并发软件和基准程序进行评估，结果表明，RaceDebugger可以诊断376个（总共382个）数据竞争，诊断率高达98.4%，同时，相比于原始调度，根本原因调度可以减少99.7%的程序事件，线程间切换降低99.3%，显著提升程序员调试效率。

详细内容参见：

Long Zheng, Xiaofei Liao, Hai Jin, Bingsheng He, Jingling Xue, and Haikun Liu. "Towards Concurrency Race Debugging: An Integrated Approach for Constraint Solving and Dynamic Slicing". In Proceedings of the 27th International Conference on Parallel Architectures and Compilation Techniques (PACT), November 1-4, 2018, Limassol, Cyprus, pp. 26:1-26:23

郑　龙
博　士
研究方向：可重构计算机体系结构及其运行时环境
Email：longzh@hust.edu.cn

检测软件并发漏洞的启发式框架

刘长鸣

文章发表在Annual Computer Security Application Conference '18上。该会议2018年接受到的有效投稿数为299篇，正式接受了60篇。该会议为安全类CCF B类会议，专注计算机安全方面各项技术的应用等。

内容概述：

现今，我们越来越依赖多核和并发带来的算力的提升，但同时由并发引起的软件错误甚至漏洞也越来越多地暴露出来。本文针对软件并发漏洞的特点，结合经典的模糊测试工具AFL，提出了一个轻量级针对并发的模糊测试框架。

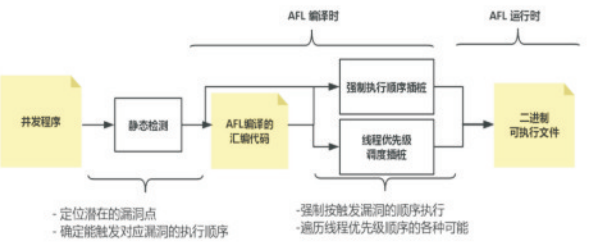


图1 系统框架

系统总体框架见图1。对于一个并发程序，众所周知，一个并发漏洞的触发除了需要特定的程序输入，往往还需要执行到特定的敏感操作，以及这些操作按照特定顺序去执行，在实际执行的过程中，由于并发带来的不确定性，这些条件不一定会被满足。故本文先对其进行静态检测，得到一系列能触发并发漏洞的敏感操作，以及这些敏感操作之间的执行顺序，再通过插桩使得程序每次执行都会按照预先设定的顺序去执行，大大提高了漏洞被触发的概率。

其次AFL由于缺少对线程的感知，不会对线程有特殊的操作，但是由经典的并发错误检测领域的工作可以得出，遍历各个线程之间的调度有助于提高并发错误被触发的概率。故本文在AFL的基础上加入了线程级别的优先级主动调度，使得AFL在模糊测试过程中尽可能多地遍历到各种线程之间的调度可能性，从而提高了并发错误被触发的可能性。

本文将这个框架用Ocaml和C语言实现，并在六个现实程序中测试。经过测试得到了之前没有发现三个之前没有发现过的并发错误，和两个没有发现的并发漏洞。

详细内容参见：

Changming Liu, Deqing Zou, Peng Luo, Bin B. Zhu, and Hai Jin. 2018. A Heuristic Framework to Detect Concurrency Vulnerabilities. In Proceedings of the 34th Annual Computer Security Applications Conference (ACSAC '18). ACM, New York, NY, USA, 529-541. DOI:https://doi.org/10.1145/3274694.3274718

刘长鸣
硕 士
研究方向：软件漏洞检测
Email：liuchangming@hust.edu.cn

SDN中基于拜占庭协议的错误交换机容忍机制

袁 斌

文章发表在IEEE Transactions on Network and Service Management上。该期刊每季度出版一期，每季度录用论文25篇左右，影响因子是3.286，主要关注网络架构、管控模型、服务可靠性与服务质量保障、使能技术、信息与通讯模型以及新兴技术与标准等方面的研究。

内容概述：

在软件定义网络数据层中，错误的内部交换机会导致控制器无法得到正确的网络状态信息（称之为控制器的输入信息），从而使得控制器无法做出正确的网络决策，进而影响网络行为的正确性。在控制器主要的输入信息中，status_reply可以为控制器提供数据层的统计信息，如每条规则所处理的数据包个数等。控制器上的多种应用都需要用到该信息，如路由决策、负载均衡等应用。然而，错误的交换机在收到控制器的获取统计信息请求时，可能表现出无回复、延迟回复、错误回复、伪装回复以及丢弃和篡改回复等多类恶意行为，导致控制器无法得到统计消息或者得到错误的统计消息。

为了解决上述问题，提出基于拜占庭协议的错误交换机容忍机制。拜占庭容错机制的主要思想是，在系统内部某些组件出现错误行为的情况下，仍保证系统能正确运行。如果把软件定义网络看成系统，交换机看成系统内的组件，交换机的恶意行为看成系统组件出现的错误行为，那么在拜占庭容错模型下，保证系统的正确运行可以看成保证控制器能得到正确的输入信息，而拜占庭模型中的具有冗余特性的服务可以通过存有同一信息的多个交换机来实现。基于上述思路，提出基于逻辑代理的错误交换机容忍框架。

系统框架如图1所示。为了把交换机从繁重的拜占庭容错机制中解放出来，在软件定义网络架构中引入一个代理层。代理层是位于控制层和数据层之间的一个逻辑层。当控制器发出网络流量统计信息查询请求时，代理层从相应的交换机上获取统计结果，执行拜占庭容错协议，最后通过独立的专用网络把正确结果返回给控制器。

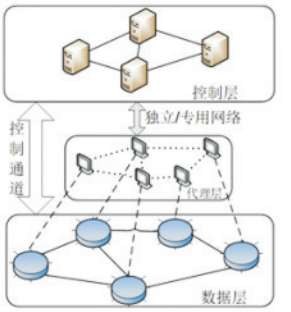


图1 基于逻辑代理的错误交换机容忍框架

通过将拜占庭方案与随机选择方案以及投票方案进行对比测试，发现拜占庭方案要明显优于其他方案（拜占庭方案能够容忍最多的错误行为类型），且错误交换机的个数、网络拓扑结构等因素对拜占庭方案的性能开销影响也十分有限。

详细内容参见：

Bin Yuan, Deqing Zou, Hai Jin, Laurence T. Yang, and Shui Yu. "A Practical Byzantine-based Approach for Faulty Switch Tolerance in Software-Defined Networks". IEEE Transactions on Network and Service Management, vol. 15, no. 2, pp. 825-839, 2018

袁 斌
博 士
研究方向：软件定义网络安全
Email：yuanbin@hust.edu.cn

社交网络中三角关系的演化研究

黄　宏

文章发表在ACM Transactions on Knowledge Discovery from Data上。该期刊2017年影响因子是1.489，为中国计算机学会推荐国际学术期刊数据挖掘领域B类期刊。主要关注知识发现，数据挖掘和多维数据分析等方面的研究。

内容概述：

三角形结构是三个人组成的最小的群体单元，是网络中的一个最基本的结构单元，被广泛用来研究网络的特性和网络的演化。然而，研究中对三角形本身的演化特征却很少涉及，三角形的形成对原网络的影响也并不清楚。本文研究了社交网络中三角形结构对网络的影响，探究了闭环三角形的形成是否会影响原有三角形的交互强度。

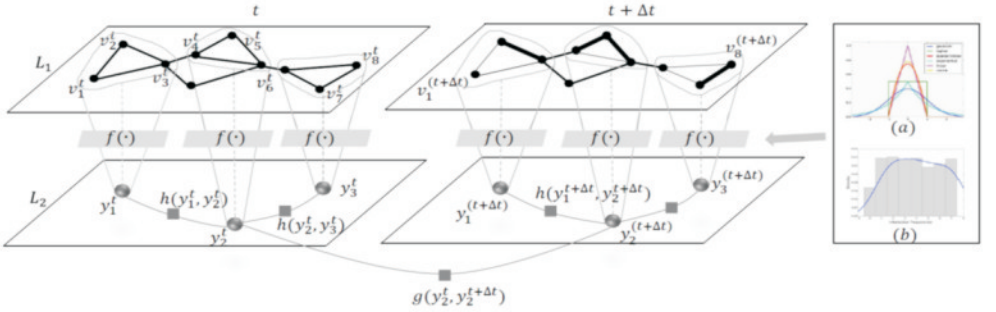


图1 算法框架

本文以微博数据为基础，分析了近180万用户，3亿条关系中三角形闭合对网络的影响。文章首先从三个维度对三角形中边的强弱关系进行了分析，即边的强弱动态变化，边的社会属性，时间的影响。我们发现：1）当三角形中最后闭合的边交互强度越强时，三角形越稳定；2）当三角形中原有边的交互强度越弱时，三角形越稳定；3）虽然男性比女性更容易形成朋友关系，但是男性间的关系不如女性间的关系稳定。

基于以上的观测，本文提出了基于三角形结构的动态因子图模型对三角形的边强弱关系进行预测。算法框架见图1。该模型不仅考虑了用户的基本属性信息，时间影响，还考虑了网络中的结构信息。除此之外，该模型还对提取的特征进行了核密度估计以提高预测的精度。我们在微博数据集和电话数据集上进行了大量的实验，实验结果表明，我们提出的模型的预测精度比当前的最优方法提高近30%。该模型亦可用于朋友推荐等应用。

详细内容参见：

Hong Huang, Yuxiao Dong, Jie Tang, Nitesh Chawla, Xiaoming Fu. Will Triadic Closure Strengthen Ties in Social Networks? ACM Transactions on Knowledge Discovery from Data (TKDD) , Vol. 20 (3), no. 30, Apr. 2018.

黄　宏
博　士
研究方向：社会计算，大数据分析与挖掘
Email：honghuang@hust.edu.cn

GPU图计算综述

郑志高

文章发表在ACM Computing Surveys上。该期刊每个季度出版一期，每期录用论文15篇左右，2017年影响因子是5.55，该期刊收录计算机所有子方向的综述文章，此前中国学者已经在该期刊上发表论文九十余篇，大陆学者在此期刊上发表论文七十余篇，以华中科技大学为第一单位发表的论文仅一篇。

内容概述：

随着大数据时代的到来，图的规模越来越大，有的甚至有数十亿的顶点和数千亿的边，快速处理如此大规模图是一个巨大的挑战。近年来，随着GPGPU计算技术的发展，尤其是其大规模并行性和高存储器访问带宽，吸引了大量研究人员研究如何应用GPGPU来加速图计算等各种应用中的计算任务。论文综述了GPU图计算的关键问题，包括数据放置、内存访问模式、任务负载以及GPU编程等。全面总结了当前的研究现状和进展，详细分析了GPU图计算存在的挑战性问题，并深入探讨了未来的研究方向。

在图处理算法方面，当前针对遍历型算法的研究主要集中于内存访问模式和任务负载两个方面，然而针对迭代型算法的研究主要集中于数据放置、任务负载等方面。在图处理系统方面，当前的研究工作主要集中于数据放置、内存访问、任务负载以及GPU编程等方面。GPU的体系结构设计是针对于计算密度高的数据并行应用，并且要求应用具备连续访存的特点。然而图数据由于其独特结构使得计算过程中的访存轨迹呈现一种跳跃式的特点。如何利用图数据组织以及预取、数据流化等手段实现GPU图数据处理的连续访存是GPU图处理最重要也最基本的问题。另一方面，图数据高度稀疏性使得在采用GPU进行图数据处理的过程中的任务负载倾斜严重，如果利用任务划分等手段保证GPU图处理任务负载的平衡是保证GPU图处理性能的一个重要方面。在系统设计中如何抽象GPU图处理的特征，设计适合GPU体系结构的编程模型；如何充分利用GPU的内存层次模型、访存模型、分支预测等技术对GPU图处理进行优化，是GPU图处理的重要补充。

本文设计了一系列的实验，对已有优化技术进行评估。同时本文还展望了未来GPU图处理的研究方向，例如基于GPU以及一些新型硬件的图处理、新型GPU体系结构上的图处理、基于GPU的动态处理以及基于机器学习的图处理应用，都是未来可能的一些研究方向。

详细内容参见：

Xuanhua Shi, Zhigao Zheng, Yongluan Zhou, Hai Jin, Ligang He, Bo Liu, and Qiang-Sheng Hua. 2018. Graph Processing on GPUs: A Survey. ACM Comput. Surv. 50, 6, Article 81 (January 2018), 35 pages.
https://doi.org/ 10.1145/ 3128571

郑志高
博士研究生
研究方向：并行与分布式计算以及图计算
Email：zhengzhigao@hust.edu.cn

基于SINR模型的多跳无线网络中稳定局部广播协议

华强胜

文章发表在IEEE/ACM Transactions on Networking上。该期刊每两个月出版一期，每期录用论文30-40篇左右，2018年影响因子是3.11。该期刊涵盖以下主题：网络架构和设计、通信协议、网络软件、网络技术、网络服务和应用程序以及网络运营管理。

内容概述：

无线网络中节点之间的信息交换通常通过共享信道上的局部广播（每个节点仅广播消息到所有相邻节点）来完成。静态局部广播（SLB）问题（即所有数据包在传播之前已经存储在节点上）已经在各种干扰模型下被广泛研究。目前连续包局部广播（CLB）问题（即节点上不断有新的数据包到达）由于更加贴近现实从而引起了更多的专注。当前连续包局部广播研究都是基于单跳网络拓扑结构和基于局部干扰模型。如何在多跳无线网络以及更加实际的物理干扰模型（一种考虑了累加干扰的信噪比SINR模型）中考虑连续局部广播问题是无线网络中信息交换的一个重要问题。

本文针对CLB问题在多跳无线网络中提出了一种用于局部广播的分布式稳定协议，其中数据包被连续注入节点，并且每个节点需要在给定的通信范围内快速地将注入的数据包传播给其在该通信范围内的所有邻居。我们给出了一个分布式稳定协议可以实现最大数据包注入速率和最小数据包局部广播延迟时间。本文基于物理干扰SINR模型，它比传统的局部干扰模型（如基于图的模型）更准确地反映无线干扰的物理特性，如衰落和干扰累积。更具体地说，我们提出了一种可以处理随机和对抗数据包注入模式的分布式稳定协议。

该协议在注入速率和数据包局部广播延迟方面都达到渐近最优。本文是第一篇研究基于SINR模型的多跳无线网络中局部广播基本操作稳定协议性质的文章。我们提出的协议使用静态局部广播算法作为子程序协议，它本身也是渐进最优的。仿真结果表明，我们提出的算法在现实环境中表现良好。

详细内容参见：

Dongxiao Yu, Yifei Zou, Jiguo Yu, Xiuzhen Cheng, Qiang-Sheng Hua, Hai Jin, Francis C. M. Lau. Stable Local Broadcast in Multihop Wireless Networks Under SINR. IEEE/ACM Trans. Netw. 26(3): 1278-1291,2018.

华强胜

副教授，博士生导师

研究方向：网络及分布式计算理论与算法

Email：qshua@hust.edu.cn

动态图下一种基于匹配的并行核值维护方法

王　娜

文章发表在IEEE Transactions on Parallel and Distributed Systems上。该期刊每月出版一期，每期录用论文20篇左右，2018年影响因子是3.971，主要关注并行与分布式架构、并行与分布式算法、并行与分布式计算应用以及并行与分布式系统软件等方面的研究。

内容概述：

图中顶点的核值是描述图的内聚性的基本指标，在大规模图分析中得到广泛地使用。在之前的动态图核值维护算法中，当加入或删除多条边时，转化为加入或删除一条边的问题，通过多次迭代直到所有边被处理完成。整个执行过程是串行的，当图的规模或加入或删除的边的规模很大时，算法的执行效率很低。我们发现当加入或删除的边构成一个匹配（matching）时，由这些边引起的核值变化能通过并行的算法进行更新。基于这一关键发现，根据一定约束条件将插入/删除的边分为多个组，使得同一个组内的边插入/删除时造成的顶点核值变化量可确定，将多边更新时的核值维护问题分解为边分组和寻找核值变化的顶点两个子问题，从而高效解决动态核值更新问题。更具体地，通过证明满足“匹配”和“优边集”性质的边集在被同时处理时，能够保证顶点的核值变化量为确定值，给出基于“匹配”和“优边集”两种不同的边分组策略，得到高效核值维护算法。其中，基于匹配的算法具有更少的预处理时间，而基于优边集的算法可同时处理更多的边，减少处理轮数。

相比于传统基于单边处理算法的多边顺序处理方式，基于分组策略的处理算法不仅极大降低核值更新的时间，提高核值更新效率，而且可以有效减少单边处理算法顺序执行过程中产生的冗余计算，降低计算成本和存储空间。此外，基于分组策略的多边处理算法允许并行执行，可通过在并行系统运行进一步提高核值更新的效率。

在真实图数据、时序图数据和生成图上的大量实验表明，基于分组策略的多边处理算法在实际环境中可高效更新核值，极大降低核值维护所需的时间，并在并行系统运行环境下具有良好的可扩展性。

详细内容参见：

Hai Jin, Na Wang, Dongxiao Yu, Qiang-Sheng Hua, Xuanhua Shi, Xia Xie. Core Maintenance in Dynamic Graphs: A Parallel Approach based on Matching. IEEE Transactions on Parallel and Distributed Systems, 29(11): 2416-2428, 2018.

王　娜
硕　士
研究方向：动态图中核值维护算法研究
Email：Ice_lemon@hust.edu.cn

面向深度学习系统的GPU内存复用及数据迁移机制

刘　博

文章发表在ACM Transactions on Architecture and Code Optimization上。该期刊每季度出版一期，每期录用论文10篇左右，2017年影响因子是1.313，主要关注面向嵌入式和通用系统的计算机架构、编程模型、编译器和操作系统等方面的研究。

内容概述：

深度学习是以大规模的训练数据以及复杂模型作为支撑，其模型训练过程需要大容量的GPU内存才能有效执行。但是，目前GPU上DRAM的容量扩展无法满足训练过程中内存消耗日益增长的要求。为了充分理解内存消耗的主要因素，我们观察训练过程的访存行为特征。首先，传统做法中框架会在训练初始过程默认分配整个模型所需的内存空间，这会导致巨大的内存消耗，因此我们考虑在训练初始时不再分配全部内存。第二，模型激活值及其梯度值的内存占用量完全一致，因此训练前向过程中为激活值所分配的内存空间可以被对应的梯度值所使用。第三，不同层之间的内存占用情况是相互独立的，因此我们考虑以层为中心按需分配内存空间。

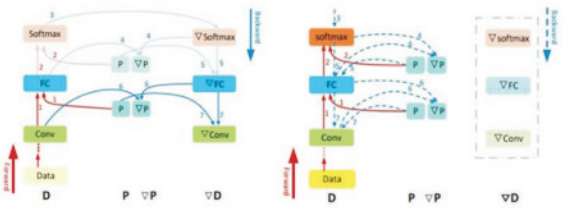


图1：（1）层内中间数据复用；（2）跨层中间数据复用

根据上述的观察及结论，我们提出一种基于模型层敏感的内存复用方法Layrub，使得训练过程的激活值梯度数据能够复用其对应激活值的内存空间，如图1（1）所示。其次，提出针对模型跨层数据复用的策略，以层为中心按需分配内存空间，并且利用远端CPU的内存作为数据迁移复用的缓冲区域，如图1（2）所示。该策略能够显著的节约内存开销，在内存复用后，内存分配最多只保留常数级的空间，例如网络中最多三层的空间。此外，由于该策略主要集中在跨层的数据复用，因此该策略更适用于深度神经网络。最后，由于两种策略分别从层内及跨层两种维度进行数据复用，我们可知，根据上文提到的数据访问序列重排规则，并保证正确计算序列的情况下，两种策略可以混合使用，能够达到更好地内存节约效果，兼顾多种形态的神经网络。

详细内容参见：

Hai Jin, Bo Liu, Wenbin Jiang, Yang Ma, Xuanhua Shi, Bingsheng He, Shaofeng Zhao: Layer-centric Memory Reuse and Data Migration for Extreme-Scale Deep Learning on Many-Core Architectures. ACM Transactions on Architecture and Code Optimization, 2018, 15(3), pp. 37:1-37:26

刘　博

2015级博士生

研究方向：深度学习系统

Email：80937591@qq.com

面向云协同无线传感器网络的轻量级可搜索公钥加密

贺双洪

文章发表在IEEE Transactions on Industrial Informatic上。该期刊于2018年起每月出版一期，每期录用论文40篇左右，2017年影响因子是5.430，主要关注工业与自动化控制系统、工程与制作系统、机器人技术、工业通信以及无线传感器网络等方面的研究。

内容概述：

云协同无线传感器网络广泛应用于各类领域，包括农业、军事防御、环境监测以及智慧交通等等，但是由传感器设备采集的海量数据面临着严峻的安全性问题。基于此，我们提出了一种适用于云协同无线传感器网络的轻量级可搜索公钥加密方案LSPE（Lightweight Searchable Public-Key Encryption），用于保护云中的传感器数据，以及实现对这些数据的安全、高效访问。

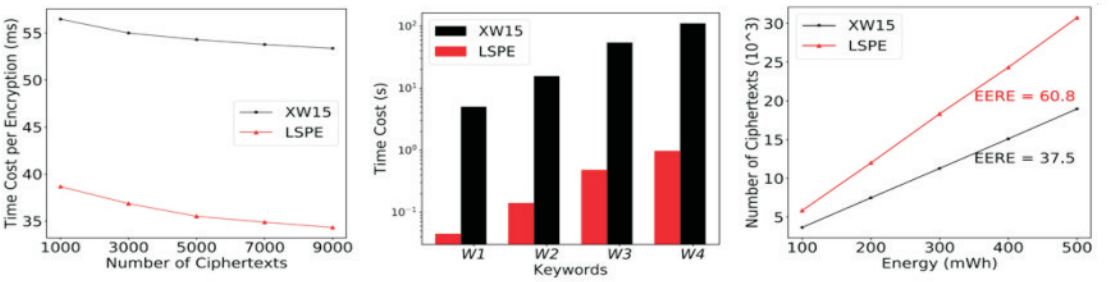


图1 试验结果图

LSPE沿用了徐鹏等人15年提出的结构化可搜索公钥加密（XW15）思想，同一个关键字的可搜索密文具有隐藏的链式结构，这样可以达到亚线性级的检索复杂度。我们基于双线性映射Paring去构造LSPE方案，并在适应性选择关键字和结构攻击的模型SS-CKSA下，基于计算性双线性迪菲赫尔曼假设CBDH完成了方案的安全性证明。为了降低加密开销并提高检索效率，采用混合加密的思想，将轻量级的乘法和除法操作取代计算昂贵的复杂密码操作例如Paring。在LSPE方案中，完成同一个关键字的N次加密只需执行一次Paring和N+1次群元素的乘法，完成一次检索只需执行一次Paring和N次群元素的除法，具有很高的效率。测试结果如图1所示。通过和XW15对比，生成相同的密文，LSPE节约了35%的时间；在检索效率方面，LSPE的检索速度大约是XW15的113倍；在能耗方面，LSPE的加密能效比EERE大约比XW15提高了62%。

详细内容参见：

Peng Xu, Shuanghong He, Wei Wang, Willy Susilo, and Hai Jin. Lightweight Searchable Public-Key Encryption for Cloud-Assisted Wireless Sensor Networks. IEEE Trans. on Industrial Informatics, vol. 14, no. 8, pp. 3712-3723, 2018

贺双洪
硕　士
研究方向：实用型可搜索公钥加密
Email：heshuanghong@hust.edu.cn

基于摘要索引的在线社交网络搜索机制

陈汉华

文章发表在IEEE Transactions on Services Computing上。该期刊是服务计算领域的权威期刊，每两个月一期，每期发表论文12篇左右，影响因子为4.418。主要关注软件工程、服务计算领域的研究。

内容概述：

随着在线社交网络的飞速发展，数以亿计的用户开始使用社交网络上进行交流与获取信息。与传统的网页搜索不同，在线社交网络常常出于对用户隐私保护的需求，限制用户可浏览的信息范围。如Facebook限制用户所发布的信息默认情况下仅可被其好友所浏览。如果使用传统的全局索引机制，则必须对全网匹配结果进行严格过滤，这对在线社交网络带来不可接受的额外开销。本研究提出一种基于用户摘要索引的在线社交网络关键词搜索机制，即为每一个用户设计一个轻量的好友内容索引。每个用户缓存其每一个好友的关键词布隆滤波作为内容摘要索引（如图1所示）。搜索时，用户首先通过其索引进行匹配，预测可能返回搜索结果的好友视图，并将查询请求发送给这些好友所对应的服务器，以此避免大量不必要的通信开销。

为了解决用户摘要索引的表示和动态更新问题，本研究同时设计了一种增量可扩展布隆过滤器结构。同时，为了进行高效的索引维护，本研究探究了一种Piggyback通讯机制，将通信优化问题转化为社交网络图的子图匹配问题，有效降低了索引维护产生的通信开销。通过使用大规模真实系统数据进行全面的实验测试，结果表明，本方法相对于Cassandra系统将社交网络关键字搜索系统的网络通信流量降低了98%。

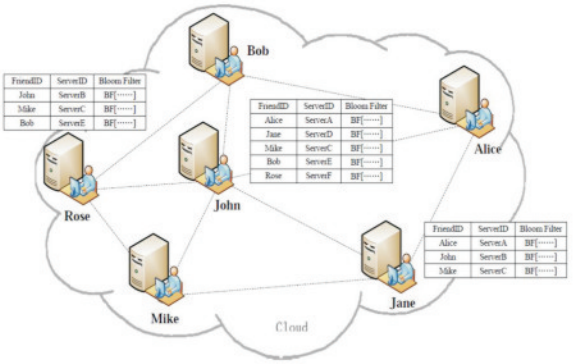


图1 用户摘要索引结构

详细内容参见：

Hanhua Chen, Hai Jin: Efficient Keyword Searching in Large-Scale Social Network Service. IEEE Trans. Services Computing 11(5): 810-820, 2018.

陈汉华
教　授
研究方向：分布式计算与系统、大数据处理系统等
Email：chen@hust.edu.cn

混合云无线网和边缘计算下的动态资源调度

王新猴

文章发表在IEEE Transactions on Parallel and Distributed Systems上。该期刊每月出版一期，每期录用论文20篇左右，2018年影响因子是3.971，主要关注并行与分布式架构、并行与分布式算法、并行与分布式计算应用以及并行与分布式系统软件等方面的研究。

内容概述：

随着移动设备的迅猛发展，云服务运营商同时采用云无线网和边缘云计算两种技术来满足日益增长的用户需求。但是相应能耗的显著增加，限制了运营商的利润的增长。已有研究工作中常常单独对云无线网或者边缘云计算的能耗进行分析。但是运营商的利润同时受云无线网和边缘云设备的影响。单纯考虑其中一项技术无法使利润最大化。基于此，我们提出一个统一的框架来均衡运营商的能耗和性能，通过联合调度云无线网中的网络资源和边缘云计算中的计算资源。为达此目的，我们将联合资源调度问题转变为一个随机优化问题，并通过拓展标准李雅普诺夫技术来设计新的优化框架。标准李雅普诺夫技术通常假设任务的长度固定或者小于一个时间片，但是移动云任务是随机变化的。为解决这个问题，我们拓展了标准李雅普诺夫技术，设计了VariedLen算法，该算法能够在多个时间片之间进行在线决策。

系统总体框架见图1。移动云用户的任务请求被远程信号接收站接收后，通过前端光传输网将任务传输到边缘云中。在边缘云中，调度器将不同的移动云任务调度到不同服务器上的容器里面。最后容器通过资源调度对个任务进行资源分配，完成任务。该调度算法主要包括如下资源调度：（1）在远程信号接收器和边缘云之间的前传光网络上，设计了基于阈值的调度算法来尽可能多的处理任务但同时防止拥塞；（2）在边缘云资源调度器中，设计了选择最小队列长度算法对任务进行分配，这是一种负载均衡的决策；（3）在每一个边缘云服务器中，设计了贪心算法来对容器的资源进行调度。

通过测试我们发现VariedLen算法能够获得接近于最优的利益，并同时能够保证整个系统的稳定性。

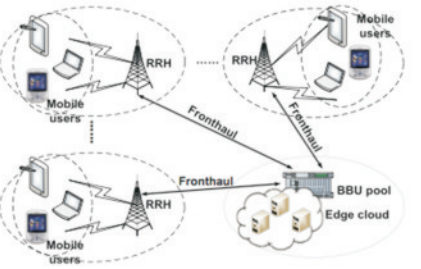


图1 系统框架

详细内容参见：

Xinhou Wang, Kezhi Wang, Song Wu, Sheng Di, Hai Jin, Kun Yang, Shumao Ou: Dynamic Resource Scheduling in Mobile Edge Cloud With Cloud Radio Access Network. IEEE Trans. Parallel Distrib. Syst., 2018, 29(11): 2429-2445

王新猴
博　士
研究方向：人工智能，AIOps，云计算资源调度
Email：wangxinhou@huawei.com

面向数据中心废热利用的高效在线交易与双赢激励机制

陈姝彤，周　知，刘方明

文章发表在ACM Transactions on Modeling and Performance Evaluation of Computing Systems上。
该期刊主要关注计算与通信系统的建模、分析及性能评估等方面的研究。

内容概述：

当前数据中心电能消耗日益增长，电能流经服务器等IT设备后产生的大量热量被直接排放出去，加剧了大气的温室效应，同时也造成了能量的浪费。论文将数据中心运维过程中的“散热”这一副产品，以新颖的“环保”和“绿色”视角看待为一种“资源”，探索数据中心是否可以将热量变废为宝。

位于温带的大城市大多拥有区域供暖系统，在寒冷天气为建筑区域供热；而主机托管数据中心常位于人口密集且经济发达的都市，这些主机托管数据中心可以方便地利用区域供热系统向用户提供环保且便宜的热量。为了激励主机托管数据中心租户参与废热再利用，论文提出了一种基于博弈论和拍卖方法的高效在线交易与激励机制，如图1所示，将废热回收并提供给区域供暖系统（如城市里的酒店、楼宇、家庭等）以提高能源利用效率并获得双赢的经济回报。

通过由实际城市供热需求数据集与托管数据中心租户计算负载数据集驱动的仿真实验，验证了所提出在线交易与激励机制的高效性，为促进数据中心废热回收再利用这一超前方案的普及提供了基础原理和可行性量化指导。实验发现，相比于传统的使用锅炉等采暖装置供热方式，本文所提出的方案能显著降低区域供暖系统采暖成本，并提高能源利用率。

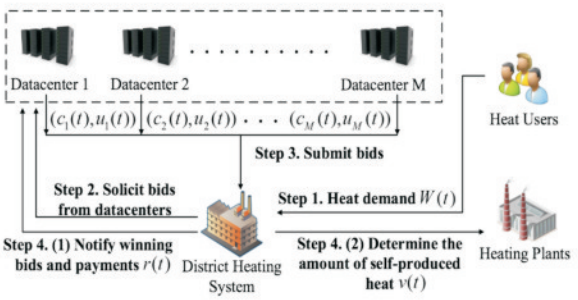


图1 主机托管数据中心废热在线交易与激励机制

陈姝彤
2015级博士研究生
导　　师：刘方明 教授
研究方向：数据中心与绿色计算、智能电网、边缘计算
Email：shutongchen@hust.edu.cn

刘方明
华中科技大学教授、博导
研究方向：云计算与数据中心、软件定义的网络SDN与虚拟化技术、绿色计算与通移动互联网。国家优秀青年科学基金获得者，国家高层次人才特殊支持计划（中组部“万人计划”）青年拔尖人才。
Email：fmliu@hust.edu.cn

一种基于符号执行的图计算系统

郑　龙

文章发表在ACM Transactions on Architecture and Code Optimization上。该期刊每季度出版一期，每期录用论文10篇左右，2017年影响因子是1.313，主要关注面向嵌入式和通用系统的计算机架构、编程模型、编译器和操作系统等方面的研究。

内容概述：

现有图处理引擎本质上是基于同步模式或异步模式进行图迭代。然而，由于图数据的复杂依赖特性，它们通常存在着显著的同步和通信开销。我们提出了一种基于符号执行的新型图计算引擎SymGraph，其可以打破数据依赖限制，实现数据依赖并行的图迭代模式。SymGraph允许使用符号抽象替代准确值参与运算。如图1所示，为充分发挥符号迭代的并行性潜力，提出了一系列精细的处理技术，包括激活诱导的图划分技术，组关联符号赋值和异步式符号聚合机制。这些方法共同克服了符号迭代引擎设计中并行开发、符号扩展、以及聚合开销的难题。

论文包含如下技术亮点：（1）对图数据复杂依赖的特性进行了广泛的研究，重新审视现有图计算系统可伸缩性问题；（2）提出了符号式图计算迭代模式，它可以实现图计算的高度并行，提高图计算的扩展效率，同时保证图计算语义的正确性和图算法的收敛性；（3）提出新型图形引擎SymGraph以充分发挥符号迭代的并行性潜力。

在12个节点的测试平台下，通过对大量真实图数据和算法进行测试，结果表明，相比同步引擎、异步引擎以及同步–异步混合处理引擎，符号迭代引擎SymGraph可提升平均1.93倍（相比同步模式）、1.98倍（相比异步模式）、1.57倍（混合模式）的性能提升。特别地，在32个节点的平台环境下，对于PageRank算法，符号迭代可显著提升算法性能高达16.5倍（相比同步模式），23.3倍（相比异步模式）和12.1倍（相比混合模式）。

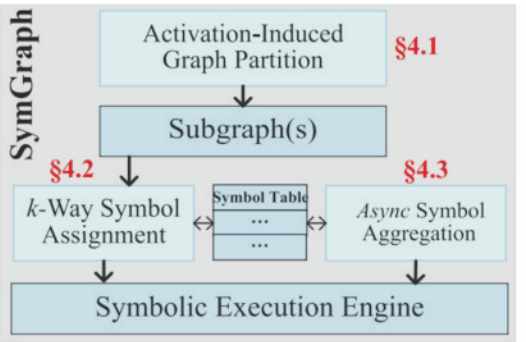


图1 SymGraph工作框架

详细内容参见：

Zheng, Long, Xiaofei Liao, and Hai Jin. "Efficient and Scalable Graph Parallel Processing With Symbolic Execution." ACM Transactions on Architecture and Code Optimization (TACO)15.1 (2018): 3.

郑　龙
博　士
研究方向：可重构计算机体系结构及其运行时环境
Email：longzh@hust.edu.cn

一种基于交替式数据处理的图计算优化方法

张　宇

该论文被期刊IEEE Transactions on Knowledge and Data Engineering收录为Regular paper。该期刊是数据分析挖掘领域的国际学术期刊。该期刊为月刊，每年出版12期，每期通常刊登10篇以上论文。

内容概述：

图普遍存在于现实世界中。对其进行快速分析广泛应用于各种真实应用中。本论文使用一种交替式数据处理策略，根据图顶点在各路径上的排列顺序，以顺序–逆序交替的方式处理它们，让图顶点状态可以更快地沿着路径扩散到其余图顶点，并对它们状态产生影响，加快各划分块内局部收敛速度。具体来说，在交替式数据处理策略中，各图划分块中的图顶点首先会按路径进行排列和存储。然后，此策略开始对各路径上的图顶点进行前涌轮以及后退轮的交替式处理直到各图顶点的状态收敛。在前涌轮中，它按路径上图顶点的存储顺序，从头顶点（排在第一的图顶点）开始对图顶点状态进行更新直到尾顶点（排在最末的图顶点）。这样，路径上的各图顶点状态能够利用级联效应快速传递给排列在它们后面的图顶点。在更新尾顶点状态之后，后退轮将会开始。后退轮将按图顶点的存储顺序从尾顶点开始对图顶点状态进行更新直到头顶点，使得路径上各图顶点状态能够快速传递给排列在它们前面的图顶点。这样，一条路径上的各图顶点状态在一个前涌轮和后退轮之后就能推送给此路径上的其余图顶点。这意味着，为了将任意一个图顶点状态推送给其余所有图顶点，整个图需要处理的轮数等于各状态推送经历的路径数量最大值Pmax。然而，循环数据处理策略需要的轮数等于图直径大小，远大于Pmax。例如，对于直径为32的图数据Com-Friendster，此策略只需要14轮图处理就能将任何图顶点状态推送给其余图顶点，获得更快的收敛速度。实验表明此方法能显著提高图处理性能。

详细内容参见：

Yu Zhang, Xiaofei Liao, Hai Jin, Lin Gu, Bing Bing Zhou. FBSGraph: Accelerating Asynchronous Graph Processing via Forward and Backward Sweeping. IEEE Trans. Knowl. Data Eng. 30(5): 895-907, 2018.

张　宇
博士后
研究方向：大数据系统软件、运行时优化
Email：zhang_yu9068@163.com

一种有效的基于磁盘的有向图处理方法

张　宇

该论文被期刊IEEE Transactions on Parallel and Distributed Systems收录为Regular paper。该期刊是并行与分布式系统领域的国际学术期刊。该期刊为月刊，每年出版12期，每期通常刊登10篇左右论文。该期刊是SCI期刊。

内容概述：

有向图普遍存在于现实世界中。对其进行快速分析广泛应用于各种真实应用中。本论文根据图顶点之间依赖关联性将图的每个强连通分量缩放成抽象的图顶点，以将图转化为一个抽象的有向无环图，然后将此有向无环图切分成多层，使得每层的强连通分量之间不存在相互依赖关联性，然后根据层次号并行处理各强连通分量。具体来说，不依赖于任何强连通分量的强连通分量们作为第一层，然后剩余的强连通分量中不依赖于任何强连通分量的强连通分量们作为第二层，以此类推直到给所有强连通分量分配一个层次号。在此之后，它根据层次序号的大小，从小到大依次将各层的强连通分量分配给空闲的计算单元进行并行处理。此外，对于真实世界的图，大量强连通分量会依赖于少部分核心强连通分量，并且部分强连通分量包含整个图的大部分图顶点，使得这些强连通分量的处理成为瓶颈。因此，在每层中，核心强连通分量被置于每层的最前面以优先处理，以在有空闲计算单元时能够处理其后续层次中的强连通分量。

其关键点包括基于强连通分量的迭代模式算法，基于强连通分量迭代的并行处理实现方案，基于强连通分量迭代的核外计算实现方案，以及用户自定义编程接口。基于强连通分量的迭代模式，将图数据以强连通分量的形式进行抽象，迭代收敛的计算按照强连通分量的拓扑顺序进行。迭代操作以强连通分量为单位，在强连通分量内部独立进行计算直至收敛。强连通分量间计算的先后顺序按照强连通分量组成的有向无环图拓扑顺序进行。强连通分量按照拓扑顺序依次被计算直至收敛，当所有强连通分量都收敛时，整个图即达到收敛状态。基于强连通分量的迭代模式分为两个阶段，预处理阶段和计算阶段。预处理阶段通过分析原始的图数据集，得到有向无环图。计算阶段依照强连通分量进行迭代计算，得到计算结果。每个数据集仅需预处理一次，之后计算阶段就可以根据不同的算法多次运行。实验表明此方法能提高图处理性能达到2倍以上。

详细内容参见：

Yu Zhang, Xiaofei Liao, Xiang Shi, Hai Jin, Bingsheng He. Efficient Disk-Based Directed Graph Processing: A Strongly Connected Component Approach. IEEE Trans. Parallel Distrib. Syst. 29(4): 830-842, 2018.

张　宇
博士后
研究方向：大数据系统软件、运行时优化
Email：zhang_yu9068@163.com

基于Pending Period的针对锁密集型程序的数据竞争检测范式

林敏豪

文章发表在IEEE Transactions on Parallel and Distributed Systems上。该期刊每月出版一期，每期录用论文20篇左右，2016年影响因子是4.181，主要关注并行与分布式架构、并行与分布式算法、并行与分布式计算应用以及并行与分布式系统软件等方面的研究。

内容概述：

现有的数据竞争检测相关工作都是基于Happen-Before关系实现数据竞争的精确检测，但是这会带来巨大的检测开销。虽然之后的工作成功地通过减少写操作的记录，减少了分析的内存开销，但是针对同步操作带来的分析开销却仍然存在，使得检测工具的可扩展性不足，而这一问题在针对锁密集型程序进行检测时，表现的更加明显。通过重新分析现有的数据竞争检测的实现原理，发现为了精确的竞争检测，使得所有的同步操作都需要被记录用于分析。但是实际上，这其中一部分的同步操作可以通过“全局时钟”的方式进行记录。因此我们设计了一种基于Pending Period的数据竞争检测机制，可以有效地减少需要记录和监控的同步操作的数量，从而减少整个数据竞争检测的开销。与此同时，我们的方法在采样数据竞争检测上同样取得了明显的效果。

我们通过对现有的数据竞争检测工具进行分析，发现现有的数据竞争检测工具在同步操作分析上存在较大的开销，可能存在可扩展性不足的问题。首先我们通过实验印证了猜想，现有的数据竞争检测工具FastTrack在线程数量增加时，检测开销会有增加，而这个现象在锁密集型程序上的表现更加明显。

通过进一步分析发现，开销的大量产生是由于为了保证数据竞争检测的精确性，保留了所有Happen-before关系，而实际上其中一部分的Happen-before关系是可以通过全局时钟进行推算的。因此我们提出一种基于全局时钟的Pending Period检测PPA，减少了同步操作的监控和分析，使得分析开销从原有的O(n)降低到O(1)，从而有效的增加了程序的可扩展性。

通过测试发现，PPA通过全局时钟有效地降低监控和分析的同步操作数量，并且对每一次同步操作的分析开销也有较大减少，从而在能够有效地降低在锁密集型程序上分析开销，同时随着线程数量的增加，检测开销也没有很明显的增加，呈现了良好的可扩展性。针对采样数据竞争，PPA同样表现出明显的性能提升。

详细内容参见：

Xiaofei Liao, Minhao Lin, Long Zheng, Hai Jin, Zhiyuan Shao. Scalable Data Race Detection for Lock-intensivePrograms with Pending Period Representation. IEEE Trans. Parallel. Syst.2018

林敏豪
硕　士
研究方向：动态程序分析与并行调试
Email：linminhao@hust.edu.cn

基于GPU的异步图处理框架

石宣化

文章发表在IEEE Transactions on Knowledge and Data Engineering上。该期刊每月出版一期，每期录用论文16篇左右，2017-2018年影响因子是2.775，主要关注数据挖掘方向的计算机科学、人工智能、电子工程、计算机应用技术等方面的研究。

内容概述：

图是一种常用的抽象数据结构，具有数据表达能力强、结构复杂等特点。随着GPGPU的技术成熟，利用GPU 加速图处理过程，成为了图数据处理的一个重要研究方向。然而，当前的GPU图计算系统仍然采取BSP 同步处理模型加速处理过程，同步处理模型存在通信开销大、无法保证任务并发度等问题。我们通过优化数据划分策略，基于异步编程模型，实现一个无锁的GPU 图计算系统。

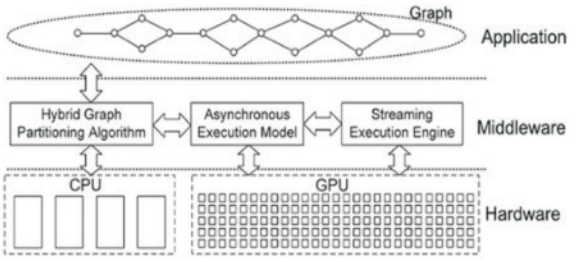


图1 系统框架

系统总体框架见图1。为提升GPU 图计算系统的性能，我们采用一种新的混合着色算法，解决传统数据划分策略中任务均衡性差的问题，使得划分结果在GPU 上被处理时能提高GPU 的资源利用率。在此基础上，我们还基于异步编程模型，实现了一种基于GPU 的无锁、并能保证异步处理过程中数据一致性的图计算系统Frog，减少由于同步通信而带来的额外开销，通过加速图算法的收敛速度，提升图计算系统的性能。此外，对现有计算系统普遍无法处理的大规模图数据，Frog 系统采用Stream 处理模式，通过迭代处理、CPU 和GPU 及时交互的方法，解决数据量超出GPU 内存大小的图数据处理问题

在基于Kepler架构的GPU（NVIDIA Telsa K20m）上测试BFS和PR的结果表明，与Medusa，Totem，Cusha、MapGraph、Gunrock以及Frog-Native等框架进行比较，Frog在所有测试数据集上均能大幅提升图处理效率。

详细内容参见：

X. Shi, X. Luo, J. Liang, P. Zhao, S. Di, B. He, and H. Jin. Frog: Asynchronous graph processing on gpu with hybrid coloring model. *IEEE Transactions on Knowledge and Data Engineering*, 30(1): 29-42, Jan 2018.

石宣化
华中科技大学计算机学院教授
研究方向：云计算与大数据处理、异构并行计算等
Email：xhshi@hust.edu.cn

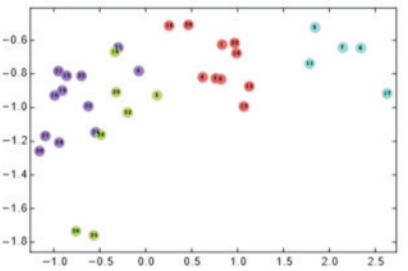
network embedding学习心得

（方子玄 https://www.cnblogs.com/venom-e/p/10036456.html）

如今，网络这种形式被广泛应用在各个领域，例如社交网络，生物网络和信息网络。随着如今大数据的火热，信息变得越来越繁多，网络也变得越来越庞大复杂，因此对于网络的分析和处理也变得越来越有挑战性。其中一项挑战就是怎么样找到一种有效的网络表示，通过这种表示使得网络相关的任务在时间和空间上更有效率。

Network embedding 是一种网络表示的方法，它通过学习来将网络中的节点映射到低纬度的向量空间中，节点之间的关系（如边）可以通过计算它们在向量空间中的距离来捕获，即网络中节点的拓扑信息和结构信息也要嵌入到向量空间中去。

(a) Input: karate network



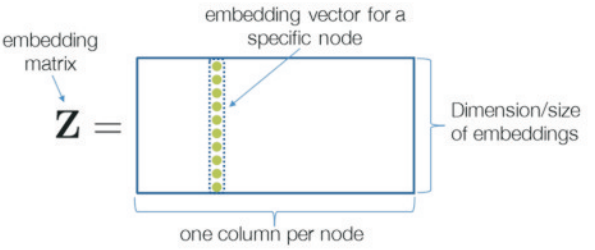
(b) Output: representations

这种网络表示方法相较传统的方法优点是十分明显的，传统的方法通常直接使用网络的邻接矩阵去表示网络，这样就很有可能包含噪声或冗余信息，并且当网络较大时，这种表示方法占用的空间也是巨大的。而基于嵌入的方法是通过学习来获得网络中节点的表示，所以可以减少噪声或冗余信息，并且可以保留内部的结构信息。同时，由于映射的方法是由研究者来定义，即研究者可以根据实际的问题来嵌入他感兴趣的信息到向量空间中，所以这种方法具有更高的灵活性和可扩展性。由于学习后的节点之间不再耦合，所以可以将主流并行计算的解决方案应用到大规模的网络分析中去。此外，network embedding与如今比较火热的机器学习方向关系紧密，许多现成的机器学习方法（如深度学习模型）可以直接应用与解决网络问题。

Network embedding通常有两个目标：首先，可以从学习的向量空间中重建原始网络，即如果两个节点之间存在边的关系，则在向量空间中这两个节点之间的距离应该相对较小，通过这种方式可以很好地保持网络关系。其次，学习的向量表示还可以有效地支持网络推理，例如边的预测、识别重要的节点和推断节点标签等。

总的来说，network embedidng有三个步骤：

1. 定义一个编码器（即网络中的节点到低纬度向量空间的映射）。



2. 定义一个节点相似度函数（即如何计算原始网络中节点之间的相似度）。

3. 通过学习来对编码器进行更新，使得原始图中节点之间的相似度可以通过向量空间中两个向量的点积来计算，即

$$\text{similarity}(u,v) \approx \mathbf{z}_v^\top \mathbf{z}_u$$

Similarity of u and v in the original network

dot product between node embeddings

可以看出，network embedding的一个关键点在于如何定义节点之间的相似度，即可以认为相连接的两个节点是相似的，也可以认为有共同邻居的两个节点是相似的，或者可以认为具有相似的结构环境的节点是相似的……以此可以设计不同的embedding方法。

这里举几个例子简单了解一下network embedding 的方法和改进思路：

1. 基于邻接矩阵的相似度

这种方法通过边的权值来定义节点之间的相似度，即两个节点之间存在边且边的权值越大，两个节点的相似度也就越大，通过此定义我们可以得到相应的损失函数：

$$\mathcal{L} = \sum_{(u,v)\in V\times V} \|\mathbf{z}_u^\top \mathbf{z}_v - \mathbf{A}_{u,v}\|^2$$

loss (what we want to minimize)

sum over all node pairs

embedding similarity

(weighted) adjacency matrix for the graph

然后可以用梯度下降（SGD）等方法去最小化损失函数，并通过此过程来不断更新Z来学习节点的向量表示。

这种方法的缺点很明显：

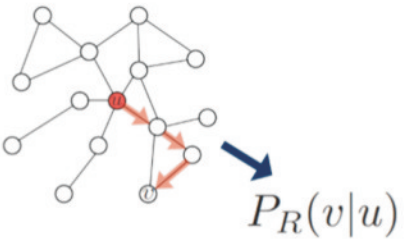
一是时间复杂度较高（O（$|V|^2$）），因为要考虑每一对节点之间的相似度。

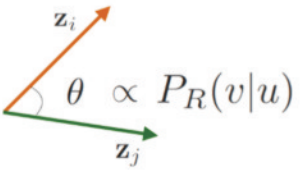
二是参数较多（O（$|V|$）），即每个节点的向量表示中的每一个元素都需要当作参数进行更新。

三是只考虑了局部直接相连的关系，对于网络的整体信息没有得到保留。

2. Random Walk Embedding

Random walk embedding 是目前比较成功且被广泛利用的一种embedding方法，目前许多最新的方法是在它的基础之上进行改进得到的，它的思想是认为在一次随机游走的过程中所遍历的节点是具有相似性的，若两个节点同时出现在一次随机游走的次数越多，则它们之间的相似度也就越高。在此方法中，节点的向量表示的点积对应的是两个节点同时出现在一次随机游走的概率。





这种方法有效地解决了上一方法中存在的问题，一是它通过随机游走整合了局部和高阶邻居的信息，二是在训练过程中不需要考虑所有的节点对（只需要考虑在随机游走过程中同时出现的节点）。

Random walk embedding的步骤如下：

A. 定义随机游走策略R和游走长度，对网络中的每个节点进行一定次数的随机游走。

B. 对每次游走的开始节点u，收集它所对应的在随机游走过程中出现的节点集合N_R（u）。

C. 根据如下的损失函数对节点的向量表示进行更新。

$$\mathcal{L} = \sum_{u\in V} \sum_{v\in N_R(u)} -\log(P(v|\mathbf{z}_u))$$

其中，P（$v|z_u$）可以通过softmax来表示：

$$P(v|\mathbf{z}_u) = \frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n\in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)}$$

DMA的一些基础知识

（李逍遥 https://blog.calmisi.cn/post/2017/3/DMA的一些基础知识.html）

总线地址

DMA的每次数据传送(至少)需要一个内存缓冲区，它包含硬件设备要读出或写入的数据。一般而言，启动一次数据传送前，设备驱动程序必须确保DMA电路可以直接访问RAM内存单元。

现已区分三类存储器地址：逻辑地址、线性地址以及物理地址，前两个在CPU内部使用，最后一个是CPU从物理上驱动数据总线所用的存储器地址。但还有第四种存储器地址，称为总线地址(bus address)，它是除CPU之外的硬件设备驱动数据总线时所用的存储器地址。

从根本上说，内核为什么应该关心总线地址呢？这是因为在DMA操作中，数据传送不需要CPU的参与；I/O设备和DMA电路直接驱动数据总线。因此，当内核开始DMA操作时，必须把所涉及的内存缓冲区总线地址或写入DMA适当的I/O端口，或写入I/O设备适当的I/O端口。

在80x86体系结构中，总线地址与物理地址一致。然而，其他体系结构如Sun SPARC和HP Alpha都包括一个I/O存储器管理单元(IO-MMU)硬件电路，它类似微处理器分页单元，将物理地址映射为总线地址。使用DMA的所有I/O驱动程序在启动一次数据传送前必须设置好IO-MMU。

不同的总线具有不同的总线地址大小，ISA的总线地址是24位长，因此在80x86体系结构中，可在物理内存的低16MB中完成DMA传送——这就是为什么DMA使用的内存缓冲区分配在ZONE_DMA内存区中（设置了GFP_DMA标志）。原来的PCI标准定义了32位总线地址；但是，一些PCI硬件设备最初是为ISA总线设计的，因此它们仍然访问不了物理地址0x00ffffff以上的RAM内存单元。新的PCI-X标准采用64位的总线地址并允许DMA电路可以直接寻址更高的内存。

在Linux中，数据类型dma_addr_t代表一个通用的总线地址。在80x86体系结构中，dma_addr_t对应一个32位长的整数，除非内核支持PAE，在这种情况下，dma_addr_t代表一个64位整数。

pci_set_dma_mask()和 dma_set_mask()辅助函数用于检查总线是否可以接收给定大小的总线地址(mask)，如果可以，则通知总线层给定的外围设备将使用该大小的总线地址。

高速缓存的一致性

系统体系结构没有必要在硬件级为硬件高速缓存与DMA电路之间提供一个一致性协议，因此，执行DMA映射操作时，DMA辅助函数必须考虑硬件高速缓存。为什么呢？假设设备驱动把一些数据填充到内存缓冲区中，然后立刻命令硬件设备利用DMA传送方式读取该数据。如果DMA访问这些物理RAM内存单元，而相应的硬件高速缓存行（CPU与RAM之间）的内容还没有写入RAM，则硬件设备读取的就是内存缓冲区中的旧值。

设备驱动开发人员可采用2种方法来处理DMA缓冲区，即两种DMA映射类型中进行选择：

1. 一致性DMA映射

CPU在RAM内存单元上所执行的每个写操作对硬件设备而言都是立即可见的。反之也一样。

2. 流式DMA映射

这种映射方式，设备驱动程序必须注意小心高速缓存一致性问题，这可以使用适当的同步辅助函数来解决，也称为“异步的”。在80x86体系结构中使用DMA，不存在高速缓存一致性问题，因为设备驱动程序本身会“窥探”所访问的硬件高速缓存。因此80x86体系结构中为硬件设备所设计的驱动程序会从前述的两种DMA映射方式中选择一个：它们二者在本质上是等价的。而在MIPS、SPARC以及PowerPC的一些模型体系中，硬件设备通常不窥探硬件高速缓存，因而就会产生高速缓存一致性问题。总的来讲，为与体系结构无关的驱动程序选择一个合适的DMA映射方式是很重要的。

一般来说，如果CPU和DMA处理器以不可预知的方式去访问一个缓冲区，那么必须强制使用一致性DMA映射方式（如，SCSI适配器的command数据结构的缓冲区）。其他情形下，流式DMA映射方式更可取，因为在一些体系结构中处理一致性DMA映射是很麻烦的，并可能导致更低的系统性能。

1. 一致性DMA映射的辅助函数

通常，设备驱动程序在初始化阶段会分配内存缓冲区并建立一致性DMA映射；在卸载时释放映射和缓冲区。为分配内存缓冲区和建立一致性DMA映射，内核提供了依赖体系结构的pci_alloc_consistent()和dma_alloc_coherent()两个函数。它们均返回新缓冲区的线性地址和总线地址。在80x86体系结构中，它们返回新缓冲区的线性地址和物理地址。为了释放映射和缓冲区，内核提供了pci_free_consistent()和dma_free_coherent()两个函数。

2. 流式DMA映射的辅助函数

流式DMA映射的内存缓冲区通常在数据传送之前被映射，在传送之后被取消映射。也有可能在几次DMA传送过程中保持相同的映射，但是在这种情况下，设备驱动开发人员必须知道位于内存和外围设备之间的硬件高速缓存。

为了启动一次流式DMA数据传送，驱动程序必须首先利用分区页框分配器或通用内存分配器来动态地分配内存缓冲区。然后驱动程序调用pci_map_single()或者dma_map_single()建立流式DMA映射，这两个函数接收缓冲区的线性地址作为其参数并返回相应的总线地址。为了释放该映射，驱动程序调用相应的pci_unmap_single()或dma_unmap_single()函数。

为避免高速缓存一致性问题，驱动程序在开始从RAM到设备的DMA数据传送之前，如果有必要，应该调用pci_dma_sync_single_for_device()或dma_sync_single_for_device()刷新与DMA缓冲区对应的高速缓存行。同样的，从设备到RAM的一次DMA数据传送完成之前设备驱动程序是不可以访问内存缓冲区的：相反，如果有必要，在读缓冲区之前，驱动程序应该调用pci_dma_sync_single_for_cpu()或dma_sync_ single_for_cpu()使相应的硬件高速缓存行无效。在80x86体系结构中，上述函数几乎不做任何事情，因为硬件高速缓存和DMA之间的一致性是由硬件来维护的。

即使是高端内存的缓冲区也可以用于DMA传送；开发人员使用pci_map_page()或dma_map_page()函数，给其传递的参数为缓冲区所在页的描述符地址和页中缓冲区的偏移地址。相应地，为了释放高端内存缓冲区的映射，开发人员使用pci_unmap_page()或dma_unmap_page()函数。

DMA设备与Linux内核内存的I/O

需要说明的是DMA的硬件使用总线地址而非物理地址，总线地址是从设备角度上看到的内存地址，物理地址是从CPU角度上看到的未经转换的内存地址(经过转换的那叫虚拟地址)。

在PC上，对于ISA和PCI而言，总线即为物理地址，但并非每个平台都是如此。由于有时候接口总线是通过桥接电路被连接，桥接电路会将IO地址映射为不同的物理地址。例如，在PRep (PowerPC Reference Platform)系统中，物理地址0在设备端看起来是0X80000000，而0通常又被映射为虚拟地址0xC0000000，所以同一地址就具备了三重身份：物理地址0，总线地址0x80000000及虚拟地址0xC0000000，还有一些系统提供了页面映射机制，它能将任意的页面映射为连续的外设总线地址。内核提供了如下函数用于进行简单的虚拟地址/总线地址转换：

unsigned long virt_to_bus(volatile void *address)

void *bus_to_virt(unsigned long address)

在使用IOMMU或反弹缓冲区的情况下，上述函数一般不会正常工作。而且，这两个函数并不建议使用。需要说明的是设备不一定能在所有的内存地址上执行DMA操作，在这种情况下应该通过下列函数执行DMA地址掩码：int dma_set_mask(struct device *dev, u64 mask);

比如，对于只能在24位地址上执行DMA操作的设备而言，就应该调用dma_set_mask(dev, 0xffffffff).DMA映射包括两个方面的工作：分配一片DMA缓冲区；为这片缓冲区产生设备可访问的地址。结合前面所讲的，DMA映射必须考虑Cache一致性问题。

内核中提供了一下函数用于分配一个DMA一致性的内存区域：

void *dma_alloc_coherent(struct device *dev, size_t size, dma_addr_t *handle, gfp_t gfp);

这个函数的返回值为申请到的DMA缓冲区的虚拟地址。此外，该函数还通过参数handle返回DMA缓冲区的总线地址。与之对应的释放函数为：

void dma_free_coherent(struct device *dev, size_t size, void *cpu_addr, dma_addr_t handle);

以下函数用于分配一个写合并(writecombinbing)的DMA缓冲区：

void *dma_alloc_writecombine(struct device *dev, size_t size, dma_addr_t *handle, gfp_t gfp);

与之对应的是释放函数：dma_free_writecombine(), 它其实就是dma_free_conherent, 只不过是用了#define重命名而已。

此外，Linux内核还提供了PCI设备申请DMA缓冲区的函数pci_alloc_consistent(),原型为：void *pci_alloc_consistent(struct pci_dev *dev, size_t size, dma_addr_t *dma_addrp),对应的释放函数为：void pci_free_consistent(struct pci_dev *pdev, size_t size, void *cpu_addr, dma_addr_t dma_addr); 相对于一致性DMA映射而言，流式DMA映射的接口较为复杂。对于单个已经分配的缓冲区而言，使用dma_map_single()可实现流式DMA映射：

dma_addr_t dma_map_single(struct device *dev, void *buffer, size_t size, enum dma_data_direction direction)，如果映射成功，返回的是总线地址，否则返回NULL.最后一个参数DMA的方向，可能取DMA_TO_DEVICE, DMA_FORM_DEVICE, DMA_BIDIRECTIONAL 和 DMA_NONE;

与之对应的反函数是：

void dma_unmap_single(struct device *dev, dma_addr_t *dma_addrp,size_t size,enum dma_data_direction direction);

通常情况下，设备驱动不应该访问unmap()的流式DMA缓冲区，如果你说我就愿意这么做,我又说写什么呢，选择了权利，就选择了责任，对吧。这时可先使用如下函数获得DMA缓冲区的拥有权：

void dma_sync_single_for_cpu(struct device *dev,dma_handle_t bus_addr, size_t size, enum dma_data_direction direction)

在驱动访问完DMA缓冲区后，应该将其所有权还给设备，通过下面的函数:

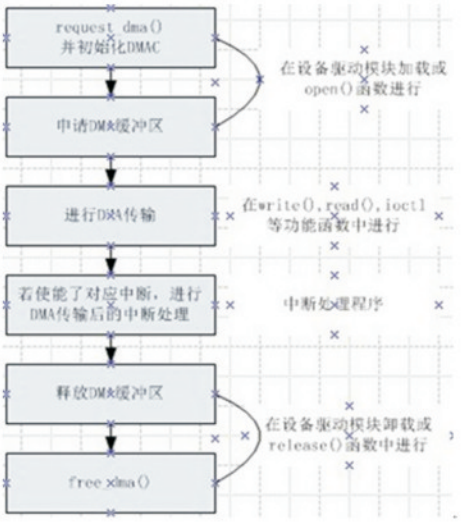
void dma_sync_single_for_device(struct device *dev,dma_handle_t bus_addr, size_t size, enum dma_data_direction direction)

int dma_map_device(struct device *dev,struct scatterlist *sg, int nents,enum dma_data_direction direction);Linux系统中可以有一个相对简单的方法预先分配缓冲区，那就是同步“mem=”参数预留内存。例如，对于内存为64MB的系统，通过给其传递mem=62MB命令行参数可以使得顶部的2MB内存被预留出来作为IO内存使用，这2MB内存可以被静态映射，也可以执行ioremap()。

在linux设备驱动中如何操作呢：像使用中断一样，在使用DMA之前，设备驱动程序需要首先向系统申请DMA通道，申请DMA通道的函数如下：

int request_dma(unsigned int dmanr, const char * device_id)同样的，设备结构体指针可作为传入device_id的最佳参数。使用完DMA通道后，应该使用如下函数释放该通道：void free_dma(unsinged int dmanr);

作为本篇的最后，总结一下在Linux设备驱动中DMA相关代码的流程。如下所示：



言 论（转载）

无论有多困难，都坚强地抬头挺胸，人生是一场醒悟，不要昨天，不要明天，只要今天。活在当下，放眼未来。人生是一种态度，心静自然天地宽。不一样的你我，不一样的心态，不一样的人生。

（林 晨 https://weibo.com/6872475828/profile? topnav=1&wvr=6&is_all=1）

生命不息，奋斗不止！只要相信，只要坚持，只要你真的是用生命在热爱，那一定是天赋使命使然，那就是一个人该坚持和努力的东西，无论梦想是什么，无论路有多曲折多遥远，只要是灵魂深处的热爱，就会一直坚持到走上属于自己的舞台！

（方升泽 https://m.weibo.cn/3960350638/4312872253809735）

我们学习中虽然苦，有许许多多的挫折，困难，等待着我们，所以我们要勇敢的面对困难，挑战困难，永不言败，那么成功离我们就不远了，成功是要付出努力的，付出汗水，没有能随随便便成功的，所以我们应该付出不懈努力去学习。

（王 璐）

人生来就注定要改变世界，哪怕是一点点。

（石 嘉）

不必太纠结于当下，也不必太忧虑未来，当你经历过一些事情的时候，眼前的风景已经和从前不一样了。

（宋 宇）

当你不想思考的时候，就不妨写点东西逼迫自己思考。特别是在写论文的时候，之前写的东西会成为你宝贵的素材库，看到笔记就会知道自己当时是如何思考的。

（费新财）

科技部中欧项目“移动网络环境下云端融合的关键技术合作研究”通过验收

郑志高

2018年10月19日，受科技部国际合作司委托，湖北省科技厅组织验收专家组在华中科技大学计算机学院召开了实验室承担的国家国际科技合作专项“移动网络环境下云端融合的关键技术合作研究”项目验收会。专家组由中南大学陈志刚教授、南京大学茅兵教授、中科院计算所孙毓忠研究员、中科院软件所王宏安研究员、杭州电子科技大学周晓慧教授、武汉工程大学陈绪兵教授、中国地质大学谢忠教授等专家组成。

该项目由我校计算机学院牵头，英国艾塞克斯大学与匈牙利科学院计算机与自动化研究所共同参与，主要围绕移动网络环境下云端融合异构资源动态适配、复杂任务透明迁移和融合服务能效优化三大挑战开展一系列研究工作。验收会上，项目负责人石宣化教授向专家组汇报了项目完成情况，介绍了合作三方在欧盟第七框架下合作项目的历史和合作研究的动机，项目组在云端异构资源动态适配、切片编程透明任务迁移、带宽波动感知的能耗优化和云端融合游戏示范等方面的技术进展，以及双方合作研究过程与成果。专家组经质询和充分讨论，一致认为，该项目结合中方云计算技术优势、英方通信领域技术优势以及匈方分布式系统技术优势，有效提升了中方云端融合技术的研发实力，完成了项目任务合同书规定的各项研究任务，通过验收。

湖北省科技厅对外合作处王锦举处长感谢各位专家的莅临，对项目给予肯定，并提出项目后期工作的建议。计算机学院党委书记吴涛和科发院相关人员也参加了验收会。

通讯员：郑志高

实验室国家重点研发计划项目启动暨方案论证会召开

郑　然

11月13日，由我实验室牵头、廖小飞教授为项目负责人的2018年度国家重点研发计划“云计算和大数据”重点专项“面向图计算的通用计算机技术与系统”项目启动会暨实施方案论证会在武汉召开。科技部高技术中心相关负责人，专项专家、顾问专家、牵头单位相关负责人和研究骨干等70余人，我校科发院和计算机学院相关负责人参加会议。

科发院副院长高亮在致辞中表示，作为牵头单位管理部门，将为项目进展提供大力支持和完善服务。

科技部高技术中心信息处处长傅耀威系统

实验室主任金海教授当选2019 IEEE会士

羌卫中，郑　然

11月21日，美国电气和电子工程师协会（IEEE）发布了2019年度新当选会士（Fellow）名单，实验室主任金海教授当选。

金海教授此次当选IEEE Fellow是基于他在对等计算和云计算系统领域的贡献。他是我校在IEEE Computer Society中当选的第一位Fellow。

电气和电子工程师协会（IEEE，全称是Institute of Electrical and Electronics Engineers）是一个国际性的电子技术与信息科学工程师的协会，是当今世界电子、电气、计算机、通信、自动化工程技术研究领域最著名、规模最大的非营利性跨国学术组织，在160多个国家和地区中拥有40多万会员和39个专业分会，是信息技术领域最重要的创新驱动源之一。IEEE Fellow即IEEE会士/院士，为协会最高等级会员，是IEEE授予的最高荣誉，在学术科技界被认定为权威的荣誉和重要的职业成就，每年由同行专家在做出突出贡献的会员中评选出，当选人数不超过IEEE会员总人数的0.1%。由于每年当选的IEEE Fellow数量较少，决定了当选科学家基本都是在科学与工程技术领域内取得重要成就的杰出科学家。

通讯员：羌卫中、郑　然

接上页

介绍了“云计算和大数据”专项的组织实施情况、专项管理流程和重点研发计划项目过程管理规范，并在项目一体化实施、规范化管理、法人责任制落实等方面提出相关建议，希望项目团队不负重托，努力完成项目任务目标。

廖小飞代表项目组介绍了立项背景、技术路线、成果形态、指标体系、进度安排以及项目管理等方面的具体情况。

通过质询，与会专家认为，项目立项必要性强，创新性突出，研发意义重大。专家们希望本项目进一步关注专利池建设，争取形成有国际影响的创新成果，加强关键技术对应用示范的支撑。

13日下午，项目组举行了项目进展讨论会。针对专家们提出的具体意见，项目组就实施方案、项目进展和进度计划展开深度交流和讨论，对实施方案具体内容进行细化，明确了后续研发思路并优化了研究方案。

据悉，项目计划构建面向图计算的新型计算机技术与系统，旨在突破基于数据流的高性能图并行执行模型和结构、图计算编程环境和运行时系统、图数据管理以及分布式环境下图计算高效执行引擎等关键难题。本项目由我实验室牵头，并联合7所知名高校、3个科研院所及3个重量级应用示范单位共同参与。项目期望通过三年研究，形成面向图计算的高效处理体系，为金融反欺诈分析、电力数据分析、社交网络数据分析等诸多重要领域的关联数据分析与决策提供重要的技术支撑，推动图计算全面推广，助力相关产业实现跨越式发展。

通讯员：郑　然

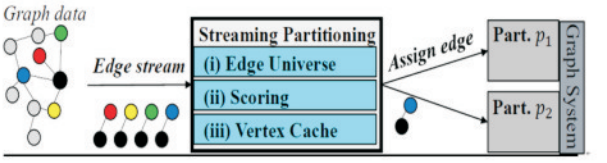
ADWISE: Adaptive Window-based Streaming Edge Partitioning for High-Speed Graph Processing

武斯杰 推荐

"*ADWISE: Adaptive Window-based Streaming Edge Partitioning for High-Speed Graph Processing*"是被计算机国际顶级会议ICDCS 2018录用的一篇文章。本文作者观察到现有分布式图划分和图计算系统在衡量系统性能时往往只关注单一的图划分时间或图计算时间，然而系统真正的目标应该是降低端到端的延迟，即降低图划分和处理的整体时间。针对这一问题作者设计了一种新的基于窗口的流式图划分系统ADWISE，既可以降低图划分时的延迟，又保证了图划分的质量，减少了图计算的时间。实验表明，与传统的图计算系统相比，ADWISE降低了最多47%的端到端处理时间。

现阶段，图数据越来越大，并且有越来越复杂的图分析要在大图上实现。如社交网络图上的社区发现，网络图上的PageRank，和生物网络中的子图同构等。传统的图计算系统有的为了降低图划分的延迟，不考虑全局信息，对每一条边单独划分，导致划分效果不好，图计算时间的增加；有的为了保证图划分的质量，降低图计算的时间，划分时考虑边与边之间的联系，却又增加了图划分的延迟。一个优秀的图计算系统应该降低端到端的时延：图划分和图计算时间之和。

现有的工作已经证明点划分比边划分的效果更好，所以本文采用的是点划分的策略。点划分有两个目标：1. 降低点的平均副本数量。2. 平衡各个机器上的负载。点的副本越少，副本之间的同步开销就会越小，图计算的时间也就越短。为了保证图划分的质量，对一条边划分时需要考虑它与其他边之间的联系。为了不增加图划分的延迟，本文提出了一个基于窗口的流式图划分系统。图一展示了ADWISE的系统结构图。左边是原始的图数据，ADWISE逐条边地读取图数据，然后通过一定规则把每条边分到不同的图分区中。

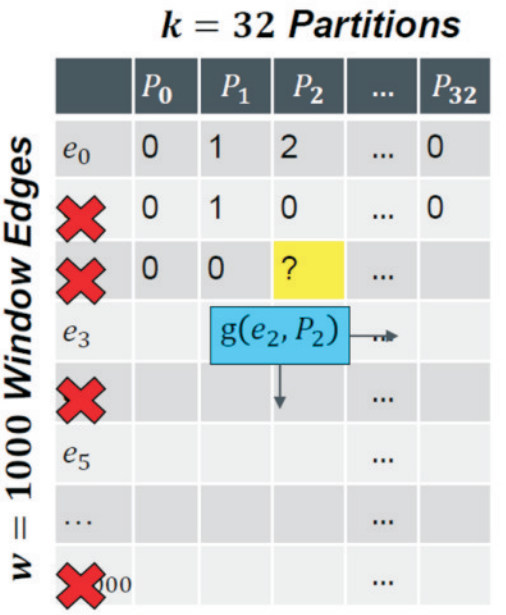


图一：ADWISE系统图

如果每读取一条边就马上划分，会导致划分效果不好，而如果读取了许多边再划分，则会增加图划分的延迟。ADWISE通过权衡划分时延和划分效果设定窗口大小，动态维护一个Edge Universe，存放现在读入系统但还未划分的边。

为了把每条边都分配到最适合它的分区内，ADWISE设置了一个Scoring函数。这个打分函数会考虑各个分区的负载均衡，点的复制比，和聚类系数。如果把一条边分到一个分区能使分区的负载更均衡，点的复制比

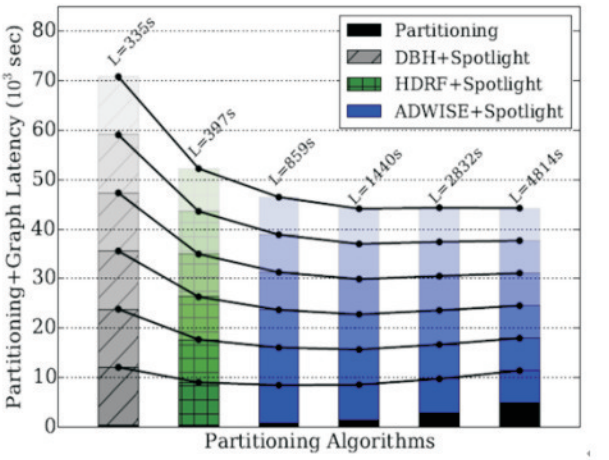
较小，聚类系数较高，那么这条边相对这个分区的得分就越高。图二展示了ADWISE为每个分区对每条边打分的过程。窗口w包含1000条边，一共有k=32个图分区。如图对边e_2为分区P_2打分，然后为分区P_3…P_{32}打分，其他边也一样。最终在窗口中选取一个得分最高的边进行划分，然后再读入一条边。以一种基于窗口的流式划分方法对整个图进行划分。



图二：边打分过程

ADWISE还会保存每个分区的Vertex Cache，记录每个分区都包含哪些点。在给每条边打分的时候，需要访问Vertex Cache以进行精准的打分。ADWISE还引入了并行划分策略，以进一步提高图划分速度。

图三展示了运行PageRank时ADWISE与现有的单边流式图划分工作的对比，分别从图划分时间和整体时间两个方面进行了比较。可以看出，尽管ADWISE的图划分时间相对较长，但其划分质量更好，图计算时间更短，因而整体时间更短。与DBH和HDRF相比，整体时间分别降低了39%和18%。



图三：实验结果

总结：分布式图计算系统依赖于快速和高效的图划分算法。本文为了降低图计算系统中的端到端延迟，设计了一种新的流式图划分策略，并实现了一个基于窗口的图划分系统ADWISE。通过权衡图划分延迟和划分质量，投入更多图划分时间以提高划分质量，降低图计算时间，实现了降低整体处理时间的目标。与现有工作相比，处理时间最高降低了47%。

武斯杰
2016级博士研究生
研究方向：大数据处理系统
Email：wsj@hust.edu.cn

Communication-Optimal Parallel Recursive Rectangular Matrix Multiplication

朱 琳 推荐

“*Communication-Optimal Parallel Recursive Rectangular Matrix Multiplication*”是计算机体系结构/并行与分布计算/存储系统IPDPS 2013录用的一篇文章。本届会议于2013年五月20日在美国马萨诸塞州波士顿举行。本文由伯克利大学数学与计算机科学教授James Demmel 及其团队完成，该团队致力于软件性能调优以及硬件设计的交互及影响。

本篇论文主要解决矩形矩阵相乘C=AB问题。提出了一种并行算法——CARMA，该算法在所有的内存范围内是通讯最优算法。并且证明了矩形矩阵乘法通讯开销的紧下界。

矩阵计算是高性能计算、科学计算以及分布式计算的核心问题。它本质上是可并行化的任务，利用并行架构的算法比串行算法有更高的性能。算法有算术运算（浮点计算）开销以及通讯开销。要设计高效可行的矩阵计算并行算法，我们要尽可能减少这两个开销。不仅要保证算术运算在各个处理器上的负载均衡，而且要减少处理器之间以及处理器内部各级之间的数据传输。目前算术运算的增长速度要远远大于带宽的增长速度，通讯开销已经慢慢成为算法的主要开销。并且随着时间增长，通讯开销所占的比重将越来越大，寻找通讯最优算法具有重要的现实意义。

我们将分布式内存机看作有P个处理器，每个处理器有内存大小M。访问另一个处理器内存数据的唯一方法是接收消息。我们计算接收和发送消息的字节数以及接收和发送消息的消息数。假设每个时间每个处理器只能发送或者接收一条消息，我们计算算法在关键路径上的带宽及延迟成本。

目前使用最广泛的并行矩阵计算算法是SUMMA算法，该算法是2D算法，对任意的矩阵维数都能保证算术运算的负载均衡，但只在确定的矩阵维数以及没有额外内存的假设下，才能够保证是通讯最优算法。James Demmel的学生Grey Ballard关于方阵相乘的通讯下界证明说明了2D以及3D算法只在确定的内存范围内是通讯最优的[1]。因此人们基于此提出了2.5D算法以及基于BFS/DFS方法的快速矩阵相乘算法[2] (Strassen’s-like matrix multiplication algorithm)。该算法对于任何内存大小都是通讯最优的，即缓存无关算法，该算法在实际应用中有更好的性能。

已知有两种并行化线性代数方法。一种需要调优，一种不需要。第一种方法基于迭代以及处理器的排列。将处理器排列为二维或者三维的形式，该类方法包括SUMMA，2.5D算法以及最近提出的3D-SUMMA，1.5D算法等等。3D-SUMMA算法对于多维矩阵相乘问题是通讯最优的，但并不针对于所有维数矩阵相乘问题。该类方法具有很高的性能，特别是当与许多现代超级计算机的网格或基于结构的拓扑相

匹配时。然而，他们在更一般的拓扑结构中就没有比较好的性能。第二种方法为BFS/DFS，被用于Strassen快速矩阵相乘算法的并行化，以通讯最优的方式设计出在实验中性能最好的密集矩阵相乘算法。BFS/DFS算法基于顺序递归算法，将处理器结构看作层次结构而不是网格结构。通过BFS和DFS步骤交替解决子问题。在BFS中，所有子问题在独立的处理器子集上并行解决，而DFS步骤利用所有的处理器解决一个子问题。BFS步骤可以降低通信开销，但相对于DFS步骤来说需要额外的内存，通过BFS以及DFS的相互交替运行可以保证算法在可用的内存中顺利运行。由于BFS/DFS的递归结构，该类算法是cache-oblivious，processor-oblivious，network-oblivious算法。除此之外，它通常非常适合分层计算机。

Demmel将BFS/DFS方法应用于递归算法得到一种通讯最优的递归矩阵相乘算法——CARMA。该算法对于任意维数的矩阵相乘都是渐进最优的。CARMA是一个简单的算法，但由于其在所有输入范围内都是最优的，它优于目前已知的所有调优算法。在每个递归步骤中，将三个维数中最大的维数进行减半，由原来的一个大问题化为两个小的子问题。然后根据可用内存来决定这些子问题需要通过BFS还是DFS来解决。

关于矩形矩阵乘法的下界，Irony[3]在2004年给出了分布式内存矩阵乘法的下界，证明主要通过将通讯开销进行分段，根据每段中标量乘法的数量，来确定该算法可能的最大段数。从而得到通讯下界。本篇论文借鉴Irony论文的证明方法，更细致的将矩阵相乘分为三种情况，利用Loomis-Whitney inequality来判断每种情况下标量乘法的数量，得到与CARMA算法相对应的下界。

参考文献：

[1] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. Minimizing communication in numerical linear algebra. SIAM J. Matrix Analysis Applications, 32(3): 866–901, 2011

[2] G. Ballard, J. Demmel, O. Holtz, B. Lipshitz, and O. Schwartz. Communication-optimal parallel algorithm for Strassen's matrix multiplication. In Proceedings of the 24th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '12, pages 193-204, New York, NY, USA, 2012. ACM

[3] D. Irony, S. Toledo, and A. Tiskin. Communication lower bounds for distributed-memory matrix multiplication. J. Parallel Distrib. Comput., 64(9): 2004

朱　琳

2017级博士研究生

研究方向：分布式计算

Email：zl@hust.edu.cn

Osiris: Hunting for Integer Bugs in Ethereum Smart Contracts

王泽丽　推荐

智能合约因与巨额数字资产挂钩而频繁遭受攻击，16年The DAO事件导致价值50百万美元的以太币（ETH）被盗，17年Patity钱包由于智能合约漏洞先后导致30百万美元和150美元的ETH被盗。本篇文章是CCF B类安全会议ACSAC 2018年录用的一篇文章，主要介绍了一种漏洞检测工具Osiris，结合符号执行和污点分析技术基于字节码检测以太坊智能合约中存在的整数漏洞。基于字节码而非源码是因为所有智能合约的字节码必须公开而源码并没有要求，目前仅大约1%合约有源码。并且整数溢出攻击是截止目前导致经济损失最高的攻击手段（见图1.），Dao攻击就是由于整数漏洞导致的。文章亮点主要集中在三个方面，一是将整数漏洞细化为算数漏洞（arithmetic bugs）、截断漏洞（trcuncation bugs）以及符号漏洞（signedness bugs），并对三种漏洞产生的原因进行了详细的分析；二是介绍了Osiris基于字节码检测整数漏洞的原理；三是基于上万条合约从定量分析和定性分析两个角度评估了Osiris的性能。

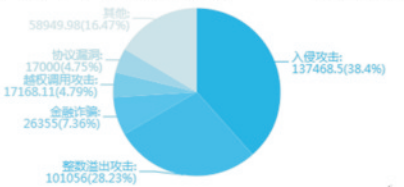


图1. 攻击手段造成的经济损失分析（来自BCSEC）

算数漏洞主要是指在加减乘除四则运算以及求模过程中而导致的整数上下溢。这主要是由于solidity编程语言与合约执行环境EVM不一致性而导致的。比如solidity支持各种长度类型的整数，uint8，uint16，uint32等等，但是EVM中只有256这一种存储长度，编程人员以为uint8就是在EVM中占8位，但实际中它可能占256位的存储。此外整数在计算机中以补码格式计算和存储，也常常会与常规整数计算模式存在一些差异，比如256位能表示的最小有符号整数，原码是2^256-1，但是在计算机实际计算中是用补码，可表示的最小有符号整数是2^256。图2即是一段存在算数漏洞的solidity代码，a、b均为32位整数，但当和大于2^32-1时，就会产生溢出。截断漏洞是指将一个整型值转换为一个较短的整型时产生的漏洞。如图3，msg.value是uint256位的整型，但是却被转换为32位的无符号整型，这个操作很可能导致发送者余额少于其实际设置的值。符号漏洞是指将有符号整型转换为同宽的无符号类型整数时导致的漏洞。图4是一段从账户余额中取钱的合约代码片段，若amout是一个很大的正整数或很小的负整数，最终可能因为在第五行代码转换为无符号类型时，而变成一个很小的负整数或很大正整数，这个漏洞可能导致攻击盗取巨额资产。

```
1 function add(uint32 a, uint32 b) public returns(uint) {
2     return a + b;
3 }
```

图2 算数漏洞代码示例

```
1 mapping(address => uint32) balance;
2
3 function() public payable {
4   balance[msg.sender] = uint32(msg.value);
5 }
```

图3 截断漏洞代码实例

```
1 function withdrawOnce(int amount) public {
2   if (amount > 1 ether || transferred[msg.sender]) {
3     revert();
4   }
5   msg.sender.transfer(uint(amount));
6   transferred[msg.sender] = true;
7 }
```

图4 符号漏洞代码例

Osiris通过污点分析模块标记可能导致不可靠数据源引入的操作（Source）和漏洞被利用的操作（Slink），然后用符号执行模块遍历执行合约所有可行的路径（通过查询Z3对约束条件求解判断路径是否可行），直至路径被遍历完或者超时。并在执行过程中将满足路径约束条件且至少有一个指令被标记的已执行指令传送给整数漏洞检测模块。该模块通过查询Z3判断在当前路径条件下，漏洞触发条件是否满足，从而判断该路径是否确实存在漏洞，并向符号执行模块返回结果。完整的Osiris体系结构见图5。在Osris漏洞检测中，最为关键的两个技术是基于字节码推断整数类型，以及如何形式化漏洞约束条件。

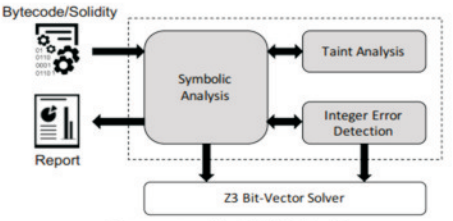


图5 Osiris体系结构概览

基于源码很容易得到类型信息（如大小、有无符号等），但是从编译之后的字节码推断得到类型是一个很大的挑战。文中提出一种基于solidity编译器在编译时期引入的一些代码优化方法即操作数执行的相关的指令推断类型。比如对于无符号整数，编译器通过AND去掉超出整数大小的一些位，类似网络中的子网掩码一样.‘0’可以去掉一些位数，而‘1’可以保留。比如，对于uint32会采用AND和0xffffffff作为其‘位掩码’。因此从AND指令可以推断出该整数是无符号的，同时由掩码位数可以得知该操作数的大小。

漏洞约束条件是决定检测工具正确性的关键。Osiris针对三种漏洞制定了详细的公式。如在算数漏洞中，对于两位无符号整数的加法操作a+b，n是这两个数中最大的位数，若a+b>2^n-1就会导致整数溢出。截断漏洞主要是通过AND和SIGNEXTEND指令来检测，因为这两种指令分别是用于截断有符号和无符号整数的。此外需要排除两类由solidity因优化需求而故意引入的良性漏洞，分别是将二进制表示截断为160位的地址类型和为了将多个变量压缩到同一个存储槽而导致的截断漏洞。符号漏洞检测通过重新构建所有整数值的符号信息，然后找到同时出现在有符号和无符号操作中出现的操作数。

最后通过大量的合约对Osiris的准确性和效率进行了评估。在定性分析中，通过与18年Security一篇文章中提出的工具ZEUS比较，发现了ZEUS未检测出的漏洞。但Osiris也存在漏报的情况，这主要是因为Osiris选取的作为Source和Slink的指令并不全。在定量分析中，通过对1207335条智能合约分析，发现其中42108条存在整数漏洞，并且大多数是算数漏洞。漏洞检测状况见图6。其中，平均漏洞分析时间为75秒，平均路径数为71个。

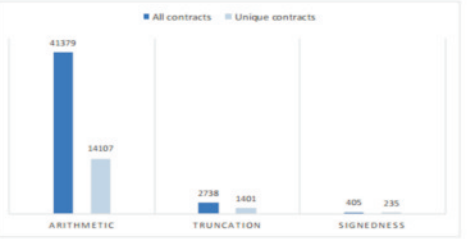


图6 Osiris检测的漏洞合约数量

总体来说，Osiris检测整数漏洞效率和准确度较高，但是仍然存在很多局限性，比如未完全模拟EVM中的指令、未考虑跨合约整数漏洞等等。目前为止，大多数合约漏洞检测都是基于符号执行技术，但是仍有少量的基于其他技术的工具取得了良好的检测效果，比如模糊测试和机器学习方法。但尚未有一种完全令人满意的漏洞检测工具，都有各自的缺陷，更好的合约漏洞检测工具将值得期待。

王泽丽
2017级博士研究生
研究方向：区块链安全
Email：zeliwang@hust.edu.com

HiKV: A Hybrid Index Key-Value Store for DRAM-NVM Memory Systems

段卓辉 推荐

本次我推荐的是中国科学院计算技术研究所的夏飞等人在ATC2017上发表的一篇文章，文章主要论述的是使用混合索引技术来提升KV存储系统在异构内存环境下的使用性能。

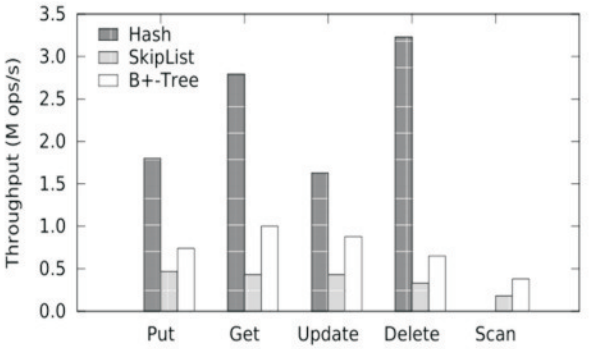


图1

随着非易失存储材料技术的成熟，非易失内存（NVM）和传统易失内存（DRAM）组成的混合内存系统有希望提供快速的持久化数据访问。但是传统的KV存储并不适用于异构内存系统，因为它们是专门针对硬盘或SSD的性能特点而设计的，并不能充分利用异构内存中不同内存介质的特性。例如，许多现有的研究，都采用Log-Structured Merge Tree作为索引结构，它避免了对硬盘或SSD的随机写问题。但与硬盘和固态硬盘不同的是，异构内存系统的字节寻址，顺序访问和随机访问具有相同的性能，因此这种索引结构就不再适用了。也有大量研究工作对NVM内存系统提出了各种关于B+树索引的优化。但是，这些优化主要着眼于降低NVM中使用B+树索引时的一致性成本。另一方面，KV存储的可扩展性受索引结构的限制。例如，为了增加KV store在多线程时的性能，可以将整个哈希索引拆分成较小的片段，从而提高哈希索引结构的并发能力。但是若将B+树索引进行拆分会带来巨大的数据拷贝和移动开销。如图1所示，作者通过对现有的3种常用KV存储数据结构的操作性能进行对比发现，单一结构的KV存储并不能对所有KV操作提供高效支持。

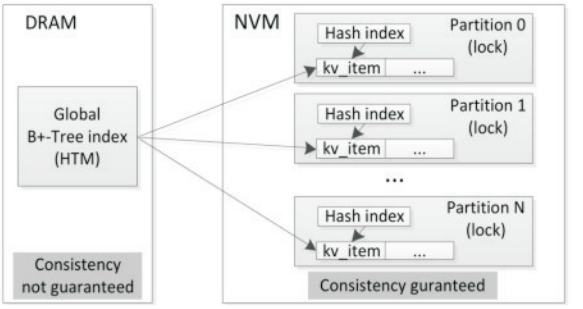


图2

KV存储的基本键值操作包括Put，Get，Update，Delete和Scan。要找到请求的键值项，单键操作（Put / Get / Update / Delete）首先只需要一个键来搜索索引，在找到KV项后，Get直接返回数据，而写操作（Put / Update / Delete）需要保留更新的索引条目和新的KV项。因此，索引搜索和数据持久的效率成为此类KV操作性能瓶颈，而哈希索引能为此类操作提供高效的搜索。同时，NVM读性能与

DRAM相同，所以在NVM中放置哈希索引是一种合理的设计选择。这种设计不仅保留了哈希索引的快速搜索，而且还允许直接在NVM中持久化索引而无需DRAM到NVM的额外数据复制。另一方面，Scan采用开始键和计数（或开始键和结束键）作为输入，而这种操作可以从排序索引获得性能收益。为了有效地支持Scan，主存储系统中采用广泛使用的B+树索引。但因为引入了不同的索引结构管理KV存储系统，保持混合索引的数据一致性成为一大问题，从根本上来说就是需要更新执行KV写入操作的哈希索引和B+树索引。由于B+树索引存在排序以及叶节点的拆分/合并，所以B+树的更新涉及许多写入操作。因此，作者将B+树索引置于快速DRAM中以避免混合内存中的慢速NVM写入问题，同时作者将哈希索引放置在NVM中，以提供更好的单键操作。其构建的混合索引架构如图二所示。

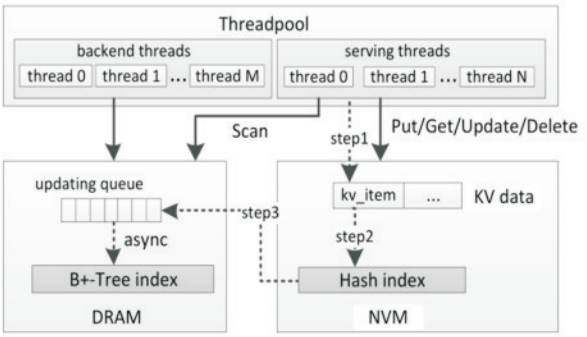


图3

当KV写入（放置/更新/删除）时，HiKV需要更新哈希索引和B+树索引以使它们保持一致状态。一个直观的解决方案是同步更新两个索引，但B+树索引的同步更新会增加KV写入的额外延迟（如搜索，排序，拆分和合并）。因此，HiKV对混合索引采用异步更新。换句话说，HiKV保留了对NVM中KV项和哈希索引的同步更新。对于DRAM中的B+树索引，HiKV在后台异步更新它以隐藏额外的延迟。

图3显示了HiKV处理不同KV操作的过程。以Put为例，HiKV首先使用服务线程来提供传入请求。服务线程负责将KV项写入NVM（步骤1），然后将新添加的索引条目写入哈希索引（步骤2）。最后，服务线程将Put请求插入更新队列（步骤3），然后返回。异步线程（称为后端线程）从更新队列获取请求，并在后台运行B+树索引。如果由于系统崩溃而无法更新B+Tree索引，HiKV可以从哈希索引中恢复B+Tree索引。

但是，只要在更新队列存在请求，Scan操作就会面临B+树索引的不一致状态。直接进行Scan操作将检索旧的或无效的数据。HiKV一旦接收到Scan操作就通过暂时阻止后续写入请求添加到更新队列来解决这个问题。Scan和后续写入一直等待到更新队列处理完所有现有请求为止。一旦更新队列变空，它就开始接收进一步的请求，同时进行Scan操作。然后，硬件事务存储器（HTM）提供对B+树索引的Scan和后续写入之间的并发控制。

在实验部分，文章在该系统上对单线程性能、延时情况、吞吐提升、可扩展性、NVM写延时敏感性等方面进行测试。实验证明，对于单线程性能，HiKV相比传统NVM的KV存储系统延迟降低86.6%。而对于多线程性能，HiKV在YCSB工作负载下将吞吐量提高了6.4倍。

段卓辉
2018级博士研究生
研究方向：内存计算
Email：zhduan@hust.edu.cn

追本溯源，拨云见月

叶晨成

苦苹果，亦名为苦黄瓜，是一种源自于地中海与西亚的攀缘植物，最早长于撒哈拉等沙漠中。其形状为直径3-10厘米的球形，果肉为白色，味道极其苦涩，虽汁水饱满，但具有较大毒性，可以致人肾脏衰竭甚至死亡。更为致命的是，苦苹果所有器官均具有毒性，以至于触碰藤蔓花叶即有中毒的可能性。与之相对的，夏日印象之一的西瓜则是甜美多汁，富含维生素，其目前三分之二产于中国[1]。令人惊讶的是，苦苹果通常被认为是西瓜的祖先[2]。问题在于，为什么最初人们会尝试驯化这种苦涩甚至危险的植物，以至于发展出了现代西瓜？同时，现代西瓜又是如何从遥远的大沙漠传入中国？回答这些问题的，是科研。

“我想做科研”，“那你来读博士吧”。第一次与金海老师相谈时的场景仍然历历在目。然而什么是科研？这是自博士第一天起便萦绕在我心头的疑惑。

回想最初加入实验室时，只是懵懵懂懂的小屁孩。彼时CPU刚迈入双核时代，多线程编程逐步普及。记得某日正在某个程序员QQ群中争论C++与Java孰优孰劣，争论焦点在于程序性能与易用性的权衡。对于信仰性能的我来说，鞭笞CPU不浪费任何一个时钟周期是唯一的目标。此时指导老师廖小飞老师发来信息，“学习一下CUDA编程”，一瞬间大量新鲜词汇闯入视野。CUDA？玩游戏的显卡还能跑程序？SIMD？似乎这就是摆在眼前的第一道沟壑。

上网搜索，查阅文档，阅读博客，尝试了各种能够想到的方法，努力去理解这种新的编程方式。慢慢的有了些眉目，同时更多新鲜词汇也喷涌而出。文档中出现了没有看过的专业名词，博客中的外部链接，文章中的参考文献，这一切似乎编织成了一个有关知识的大网，纵横交错，繁复层叠。

别无他法，只能继续追查下去，了解各个词汇的含义后，我终于开始编写第一个CUDA程序。紧随而来的是死锁，同步，随机访存导致的性能低下，Data Race…，一步步设计算法，一行行检查代码，一个个解决问题。最终，第一个程序成功运行后，迎来的是串行程序无法望其项背的性能与学会新技能的欢愉。随后的我马上投入了第二个程序的编写中，同样的问题纷至沓来，阅读文献，解决死锁，同步，随机访存，Data Race。同样的步骤，同样困难，同样的喜悦。“设计好的算法，编写好的代码，大概这就是科研吧。”

“你们去参加一下这个会议吧”，廖老师对学长们和我说道。沉醉于学习新的编程技巧的我起初并不理解老师的用意。那是第一次参加学术会议，对会议的了解仍然停留在口耳相传上，大概是很多学术大牛展示他们的研究成果。什么是研究成果？当时并不清楚。

记得那次会议选址在一家酒吧中，架起投影仪，摆好凳子，就可以开始了。楼上开会，楼下茶歇，倒是显得挺方便。演讲者描述着自己的工作，参会者有坐在地板上的，有时不时蹿下楼端着盘零食回来的，也有依墙而立的，这种随意似乎更像是茶馆，似乎分享的不是科研成果，而是对茶叶的品鉴。

[1] http://www.factfish.com/statistic/watermelons%2C%20production%20quantity

[2] Chromosome numbers, Sudanese wild forms, and classification of the watermelon genus Citrullus, with 50 names allocated to seven biological species

然而轻松的外表却掩盖不了一次次认知的冲撞。演讲者旁征博引，提问者引经据典，看似简单的问答时不时展现出满腹的经纶。如果用武侠小说比拟的话，大概是坐在地板上端着甜点的扫地僧吧。印象最深的几个报告，学者们从SSD的磨损问题起始，论证到RAID对Durability的影响，最终设计新型的RAID系统。亦或源自Non-Volatile Memory与Volatile-Memory 混合使用时的一致性问题，推演出Fault-Tolerance 编程模型。

巧妙的设计，严谨的验证，这些系统解决了一类又一类的问题。台下的我仔细倾听，费尽全力竟不能跟上节奏，时不时陷入对问题的不解，对设计的疑惑，节奏上的偶尔落后经常如同雪球中的第一颗雪花，越滚越大，越落越远。即便如此，从这次会议中获取到的，理解一类问题，了解设计的初衷，最终的成果，无一不使我兴奋。“科研大概就是针对一类问题，设计足够好的系统吧。”

再后来，金老师定期组织博士沙龙，每位博士挑选自己感兴趣的领域，对最新技术进行展示。例如新型内存ReRAM，这种在1963年被提出，目前处于蓬勃发展中的内存介质，不仅断电不丢失数据，甚至可以将存储单元转化为计算单元，进而具备计算能力。

在最初的沙龙中，我们讨论ReRAM中模拟信号与数字信号的转换，但是苦于数理逻辑早已沉底马里亚纳海沟，无从捞起，因此时刻保持着云里雾里的状态，偶尔附和一下以示仍然清醒。渐渐地内容转移至如何使用ReRAM中的逻辑门实现计算单元，虽然相关知识仍在深海之中，但对这种新型内存的神通广大有了朦朦胧胧的了解，慢慢地开始有了少许问题，或是对背景知识的咨询，或是对实现细节的疑惑。后来发展至对ReRAM计算单元总线链接方式的讲解，神秘的面纱逐渐被揭开，对于ReRAM的计算模式，存储能力有了更为全面的了解，就像是从伸手不见五指的海底逐渐上浮到水面，呼吸着新鲜空气，沐浴着明媚阳光。讨论渐渐增多，问题渐渐深入，仿佛一次学术会议。“这才是科研吧，接触最尖端的技术，从无知到博学。”

然而，海面上有的不只是一览无遗的风景，还有更加高远的天空，甚至有着无穷无尽的宇宙。在沙龙中对尖端技术的了解像是触发了匣子的开关，瞬间迸发出无数的问题。什么场景下可以使用ReRAM的计算能力，能否实现在线存储单元与计算单元的身份转换，亦或是传统编程模型是否与这种新型内存介质契合…看到的多了，却也发现还有更多目不能及。

新增的疑惑如此之多，好似一座大山劈头盖脸地压了上来。神奇的是千钧大山并没有将我沉回海底，反倒更像是填满了海沟，从此有了立足之处，可以向着天空攀爬。

科研或许是学习新的编程框架，或许是了解一类问题，设计一类系统，或许是钻研尖端技术。随着岁月的流逝，我对科研的理解不断变化，或许是更加深刻了，亦或许是返璞归真了。慢慢地，我发现，这些都是科研，或是屋檐一瓦，或是房中一门，或是厅堂一柱，这一切只是科研展现给我们生活的不同角度。而在这栋宏伟的建筑中，最默默无闻的是那深埋地下的，好奇心。

什么是科研？博士生涯回答了我，科研是满足对科学，对技术永不枯竭的好奇心。

写在最后

博士的生涯有苦涩，有充实，有焦躁，有泪水，当然也有笑容。或许在最初，我的认知中攻读博士学位是学习更多技术，掌握更多手艺。老师的教导，同学的交流，慢慢地让我发现，博士生涯似乎只是为了找寻学习的本源。与其说在CGCL的时光教给了我众多知识，不如说它告诉了我，我想要什么。

叶晨成

2011级博士

研究方向：局部性理论与优化，编译优化

Email：ccye@hust.edu.cn