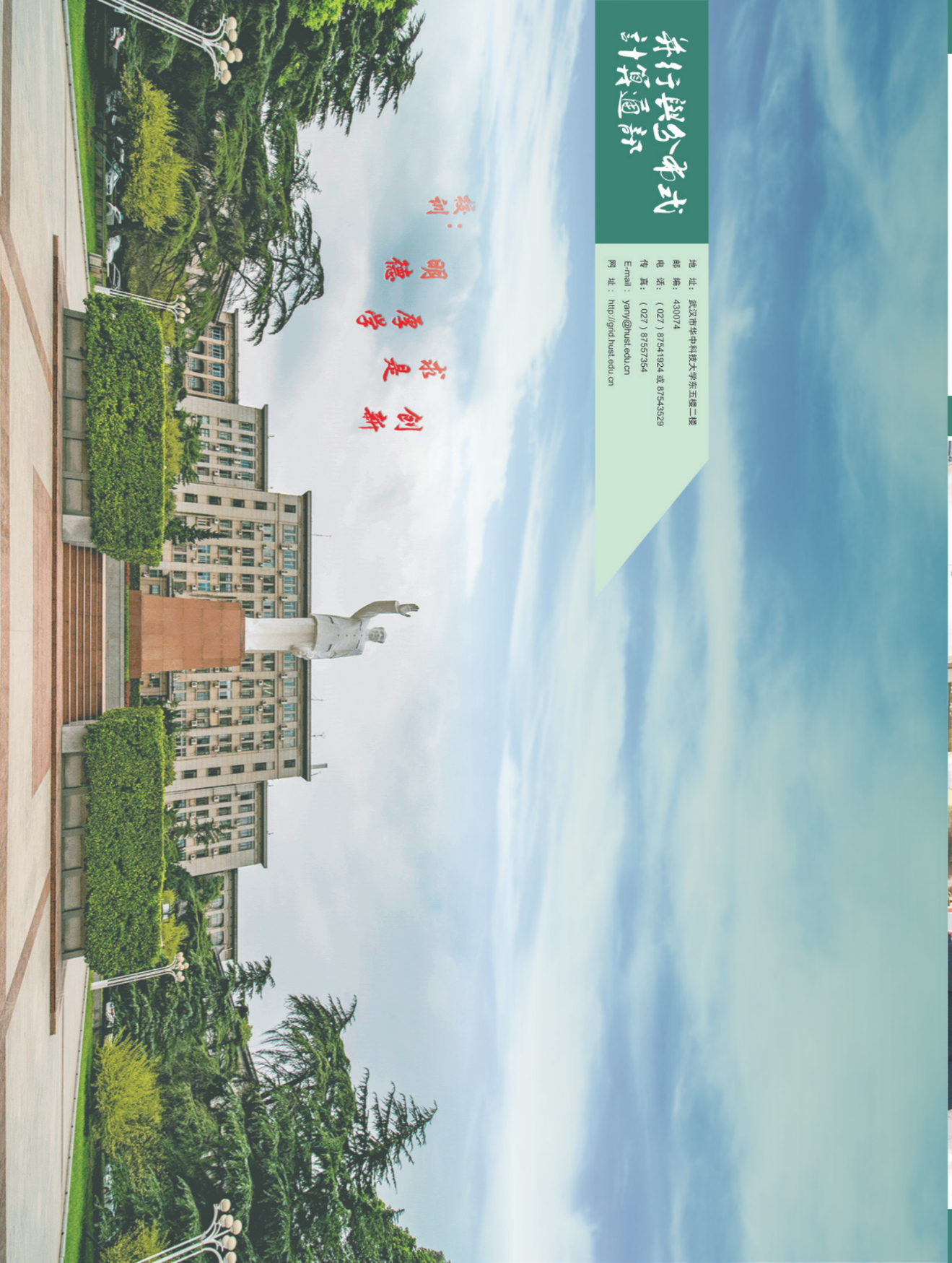


并行與分布式計算通訊

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：yanyj@hust.edu.cn
网　址：http://grid.hust.edu.cn

BDTS 大数据技术与系统国家地方联合工程研究中心
SCTS 服务计算技术与系统教育部重点实验室
CGCL 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

2021年第1期　总第44期　2021年03月

封面人物：高彬——行至水穷处，坐看云起时
2021年度实验室各研究方向规划
Deep Leakage from Gradients
开拓视野，合作交流会让科研事半功倍

http://grid.hust.edu.cn

服务计算技术与系统教育部重点实验室第十二次学术委员会会议

华科-华为数据中心架构联合创新实验室课题交流研讨会

聚焦国家战略需求、规划学科发展新方向

——2021年度实验室科研方向年度规划

2020年是极不平凡的一年，一场突如其来的病毒，打破了时间的节奏，严重干扰了大家正常的学习生活节奏。实验室的广大师生在家封闭期间，仍然通过网络坚持科研、办公，在过去的一年里取得了国家自然科学奖这样重磅的成绩。实验室老师们齐心协力，通过广泛讨论分析，提出了各自研究领域拟在2021年攻克的主要挑战以及符合国家战略需求和学科新方向发展趋势的研究内容。本期季刊将从分布式系统、系统软件和体系结构、网络空间安全和大数据这四个实验室研究方向出发，向读者展示实验室在新一年的具体科研规划。

在分布式系统方向，2021年将重点围绕内核虚拟化、容器核心系统、边缘计算、区块链等方面开展研究工作。在内核虚拟化方面，将重点研究云计算环境下操作系统内核虚拟化方法；在容器核心系统方面，将重点研究容器网络的控制和管理，以及分布式环境下数据容器关键技术；在边缘计算方面，将重点研究边缘计算数据中心高效能系统结构、边缘计算数据中心网络功能虚拟化、面向边缘容器的深度网络模型切分技术、面向多任务应用的边缘设备协同技术与系统，以及面向边缘环境的联邦学习技术；在区块链方面，将重点研究高并发新型区块链体系架构的设计理论与方法。

在系统软件和体系结构方向，2021年将重点围绕图计算加速器、异构内存系统、数据流体系架构、RISC-V处理器、chiplet芯片设计等几个主要方面展开研究。在图计算加速器方面，将重点研究基于FPGA的高可扩展图计算加速器架构、图加速器的可编程性，以及针对复杂图应用场景下的动态图计算加速器；在异构内存系统方面，将重点研究分布式异构内存池的资源管理和访存优化策略、RDMA网络中持久内存（PM）的数据高效持久化机制，以及简单易用且高效的分布式异构内存池编程模型；在数据流体系架构方面，将重点研究基于FPGA的高吞吐数据流处理模式及数据流加速器的仿真、面向大数据处理的高并发流水结构、数据流执行引擎和软件环境，以及分布式环境下大数据高效处理的数据流任务的动态划分与高效通信方法；在RISC-V处理器和chiplet芯片设计方面，将重点研究大数据应用背景下RISC-V的核心软硬件设计、chiplet体系架构的模拟器，以及chiplet的芯片设计和电子设计自动化（EDA）工具链软件设计。

在网络空间安全方向，2021年将重点围绕网络空间安全领域最新系列关键问题开展研究，主要包括：消费物联网安全、软件漏洞自动挖掘与利用、智能系统的数据与算法安全、区块链安全、云计算安全防护和基于机器学习的硬件安全攻防等六个方面。在消费物联网安全方面，将重点研究消费物联网中的供应链安全、消费物联网的协议安全，以及消费物联网的实践安全；在软件漏洞自动挖掘与利用方面，将重点研究面向开源软件的漏洞数据自动采集与标注、基于系统调用依赖覆盖率引导的内核模糊测试系统、面向Java程序模糊测试的种子生成与优化系统、漏洞利用可视化系统，以及高效和鲁棒的恶意软件检测；在智能系统的数据与算法安全方面，将重点研究基础数据层安全、中间算法层安全，以及应用平台层安全；在区块链安全方面，将重点研究区块链中的身份认证与访问控制、区块链数据一致性、合约安全，以及区块链应用安全；在云计算安全防护方面，将重点研究容器云环境中的动态安全防御、Spectre推测性执行的攻击与防御，以及云管理平台安全测试；在基于机器学习的硬件安全攻防方面，将重点研究新型安全的强PUF结构设计IPUF结构的新型机器学习算法，以及PUF结构的通用安全性测试方法。

在大数据方向，2021年将重点围绕大数据基础理论、大数据处理支撑技术、大数据管理与分析技术等关键问题开展研究。在大数据基础理论方面，将重点研究动态大图图性质维护的的低复杂度算法，以及资源受限情况下的最优间隔分布学习机；在大数据处理支撑技术方面，将重点研究图感知的缓存策略加速GNN训练过程的方法、深度学习系统多任务内存管理、基于硬件加速的内存数据库、实时关联大数据高时效处理系统、巨量图的计算与学习、针对多源异构大数据的高效数据管理，以及大规模异质时序网络表示学习方法；在大数据管理与分析技术方面，将重点研究基于数据共享与增强的医疗大数据建模学习与解析、移动边缘计算下的大数据查询处理方法，以及代码大数据分析。

衷心祝愿实验室在新的一年圆满完成各项科研任务，服务国家战略需求，瞄准计算机学科的新兴方向，为祖国的建设添砖加瓦，早日在新兴方向的发展上有所建树。

邵志远

二〇二一年三月

主　编：金　海
本期执行主编：邵志远
编　委：（按姓氏拼音排序）陈汉华、代炜琦、丁晓锋、耿　聪、顾　琳、胡　侃、胡胜山、华强胜、黄　宏、蒋文斌、廖小飞、刘方明、刘海坤、刘英书、陆　枫、马晓静、毛伏兵、羌卫中、邵志远、石宣化、陶　莉、万　瑶、王多强、王虹飞、文　明、吴　松、肖　江、谢　夏、徐　鹏、杨已彪、姚德中、叶晨成、余　辰、袁　斌、袁平鹏、张　腾、张　宇、章　勤、赵　峰、郑　龙、郑　然、邹德清
责任编辑：燕　燕
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：yany@hust.edu.cn
Homepage：http://grid.hust.edu.cn
（此刊仅供内部交流学习）

卷首语

……1

热点

……3

封面人物

行至水穷处，坐看云起时……陈　琼　7

专栏

分布式系统组研究规划
……吴　松，刘方明，余　辰，肖　江，顾　琳，姚德中　11
支持大数据高效处理的新型体系结构和系统软件
——体系结构与系统软件组科研规划
……廖小飞，蒋文斌，邵志远，郑　然
刘海坤，郑　龙，张　宇，毛伏兵，叶晨成　14
网络空间安全组研究规划……邹德清，羌卫中，徐　鹏，马晓静，代炜琦
文　明，胡胜山，杨已彪，王虹飞，袁　斌　19
大数据组研究规划……石宣化，陈汉华，赵　峰，袁平鹏，华强胜
陆　枫，丁晓锋，黄　宏，张　腾，万　瑶　24

声音

知识图谱与图神经网络的融合表示……魏昊辉　28
零知识证明在区块链中的应用……庹舒月　30
关于不完整数据集的聚类……朱玉莲　33
Zookeeper 的 Leader 选举……李少锋　35

动态

“华科-华为数据中心架构联合创新实验室课题交流研讨会”顺利召开
……毛伏兵　40
实验室学术委员会第十二次会议顺利召开……陶　莉　41

推荐

DELTA: A Security Assessment Framework for Software-Defined Networks
……任昆仑　推荐　42
Alleviating Irregularity in Graph Analytics Acceleration:
a Hardware/Software Co-Design Approach……陈绍鹏　推荐　44
Deep Leakage from Gradients……付佳韵　推荐　46
Connecting Embeddings for Knowledge Graph Entity Typing
……张照博　推荐　49
A Decentralized Blockchain with High Throughput and Fast Confirmation
……岳镜涛　推荐　51
Experience: Aging or Glitching? Why Does Android Stop Responding
and What Can We Do About it?……黄均浩　推荐　54
Livia: Data-Centric Computing Throughout the Memory Hierarchy
……陈申彪　推荐　56
PaGraph: Scaling GNN Training on Large Graphs via Computation-aware
Caching and Partitioning……何海恒　推荐　57
DeepMutation++: A Mutation Testing Framework for Deep Learning Systems
……王　健　推荐　59
Smart Contract Vulnerability Detection Using Graph Neural Networks
……祝　语　推荐　61

交流

开拓视野、合作交流会让科研事半功倍……吴月明　64

NVIDIA推出全新DPU处理器 透露未来三年DPU发展路线图

（陈佳杰 整理）

在GTC 2020 秋季站上，NVIDIA创始人兼CEO黄仁勋透露了未来三年NVIDIA DPU发展路线图。其中，包括全新的NVIDIA BlueField -2系列DPU和NVIDIA DOCA 软件开发套件，可用于在DPU加速的数据中心基础设施服务上构建应用程序。

黄仁勋表示："数据中心已成为新型计算单元。在现代化、安全的加速数据中心中，DPU已成为其重要的组成部分。CPU、GPU和DPU的结合，可构成完全可编程的单一AI计算单元，提供前所未有的安全性和算力。"

据介绍，经过优化的BlueField-2 DPU可从CPU上卸载关键的网络、存储和安全任务，使企业能够将其IT基础设施转变为最先进的数据中心。此类数据中心可实现加速、具有完全可编程性，并具有"零信任"安全功能，防止数据泄露和网络攻击。

单个BlueField-2 DPU可以提供相当于消耗125个CPU内核所提供的数据中心服务。由此释放宝贵的CPU内核，以运行各种其他企业应用程序。

BlueField-2 DPU集成NVIDIA Mellanox ConnectX-6 Dx智能网卡，与Arm核心相结合，实现完全可编程，提供200Gb/s的数据传输速率，并加速关键的数据中心安全性、网络和存储任务，其中包括隔离、信任根、密钥管理、RDMA/RoCE、GPU Direct、弹性块存储、数据压缩等。

BlueField-2X DPU具备BlueField-2 DPU的全部关键特性。此外，可应用于数据中心安全、网络和存储任务的NVIDIA Ampere GPU的AI功能也在不断强化这些关键特性。BlueField-2X DPU基于NVIDIA第三代Tensor Core核心，可采用AI进行实时安全分析，包括识别提示窃取机密的异常流量、线速加密流量分析、恶意活动的主机自检、以及动态的安全编排自动化响应（SOAR）。

据悉，全球领先的服务器制造商均计划在其企业级服务器产品中集成NVIDIA DPU，包括华硕、源讯、戴尔技术、富士通、技嘉科技、新华三、浪潮、联想、云达科技和超微。BlueField-2 DPU目前可提供样品，预计2021年将在领先服务器制造商的新系统中使用。BlueField-2X DPU正在开发中，预计将在2021年上市。

（参考链接：http://www.techweb.com.cn/ucweb/news/id/2806313https://baijiahao.baidu.com/s?id=1679773212896651295&wfr=spider&for=pchttps://www.sohu.com/a/435513498_505888）

工业界热点 "大数据杀熟"是否构成垄断？

（史瑞泽 整理）

近年来，我国平台经济迅速发展，但与此同时，关于平台经济领域经营者要求商家"二选一"、大数据杀熟等涉嫌垄断问题的反映和举报日益增加。这些行为损害了市场公平竞争和消费者合法权益，不利于促进平台经济创新发展。为预防和制止平台经济领域垄断行为，促进平台经济规范有序创新健康发展，国务院反垄断委员会于2月7日发布了《国务院反垄断委员会关于平台经济领域的反垄断指南》，强调反垄断法及配套法规规章适用于所有行业。《指南》共六章24条，包括总则、垄断协议、滥用市场支配地位、滥用行政权力排除限制竞争和附则等内容。

与传统产业领域相比，平台经济领域的垄断协议行为呈现一些新的特点：

1. 行为更具隐蔽性。数据、算法等的运用使发现和判定垄断协议更加困难。数据、算法可能会帮助经营者迅速交换价格等敏感信息，并及时监督共谋者履约情况，且算法等取代了电话、邮件等传统沟通方式，可能通过数字信号交换特定敏感信息，增加执法机构发现及调查取证的难度。

2. 更易于达成轴辐协议。平台经营者组织双边或多边群体进行交互和匹配，对价格机制、交易机制等进行设置和干预。以平台经营者为轴心，平台内经营者为辐条，借助算法等技术工具达成和实施垄断协议的行为更容易实施。

3. 平台经营者可能对平台内经营者与其他竞争性平台的交易条件提出要求。比如，平台经营者要求平台内经营者在价格、数量等方面向其提供等于或优于其他竞争性平台交易条件的行为较为常见。

针对社会各方反映较多的“二选一”“大数据杀熟”等热点问题，《指南》明确，认定平台经济领域滥用市场支配地位行为，通常需要先界定相关市场，分析经营者在相关市场是否具有支配地位，再根据个案情况分析是否构成滥用市场支配地位行为。

下一步，反垄断执法机构将深入把握平台经济发展规律和特点，强化平台经济反垄断监管，不断健全平台经济监管规则，严格规范公正文明执法，保护市场公平竞争，维护消费者利益，促进我国平台经济规范有序创新健康发展。

（参考链接： https://news.cctv.com/2021/02/09/ARTIbU2jHzfavaS7dorcALNz210209.shtml）

汽车网联安全面临挑战

（黄轲轩 整理）

国家标准委前不久正式发布《信息安全技术汽车电子系统网络安全指南》，《安全指南》将于今年11月1日实施。这是我国第一项汽车电子领域网络安全国家标准。

电影《速度与激情8》有这样的情节，网络恐怖分子攻击周边几公里范围内的汽车，把这些受遥控的“僵尸车”作为攻击武器。这个情节看似科幻，但已现实存在了，尤其在智能网联汽车加速发展的当下。

汽车社会应当高度重视，针对黑客攻击做好各种防范。

专家介绍，早在2015年，克莱斯勒的JEEP车型娱乐芯片漏洞就曾被黑客远程攻破；2016年，特斯拉车载系统遭无接触远程破解，黑客可以在车辆行驶过程中打开汽车后备箱甚至让车辆突然刹车……

传统汽车安全配置分主动安全和被动安全配置。如今，守护汽车的网联安全同样重要。因为在碰撞风险之外，新的风险隐患正在酝酿。回顾汽车安防技术的发展史，长期以来主要围绕防盗技术进行升级。然而，伴随汽车工业与信息产业融合发展，尤其是5G、智能网联技术等带来一系列新应用场景，网络、蓝牙、WIFI、USB等对外连接，都成为黑客眼里“脆弱的窗口”。汽车安防早已超越传统意义的防盗概念，向电脑和网络防火墙的作用进化，防止汽车被远程劫持、被远程监听监视、被云上攻击造成大规模信息泄露等成为防守重点。

电子科技大学教授罗蕾介绍，我国数十家汽车整车厂采用或规划的具有联网功能的汽车电子产品中的大部分尚不具备信息安全机制，存在安全风险。

Upstream Security发布的2020年《汽车网络安全报告》显示，自2016年至2020年1月，汽车网络安全事件的数量增长了605%，仅2019年一年就增长了一倍以上。过去十年，前三大攻击媒介分别是无钥匙进入系统（30%）、后端服务器（27%）和移动应用程序（13%）；安全事件造成的后果位列前三的分别是汽车盗窃/入侵（31%）、对汽车系统的控制（27%），以及数据/隐私泄露（23%）。

当我们畅想智能网联汽车带来的各种便捷时，必须未雨绸缪，构筑好守护汽车网联安全的防火墙。

守护汽车网联安全，应当高度重视产业顶层设计。今年初，国家发展改革委等11部门联合出台《智能汽车创新发展战略》，其中的重要内容就是围绕安全管理联动机制、网络安全防护能

力、数据安全监督管理等方面进行规划，引导构建智能汽车网络安全体系，推进智能汽车创新发展。相关厂家、协会应当围绕汽车网联安全，加大研发投入和产业跨界对接力度，参照《安全指南》构建全面高效的智能汽车网络安全体系。

守护汽车网联安全，还应当高度重视立法工作的衔接。2017年6月1日《中华人民共和国网络安全法》正式实施，同样能为汽车上的网络空间提供守护。未来，针对汽车网联安全有可能发起的各类攻击，道交法、反恐法、刑法等法律修订时都应当考虑设置应对条款。

汽车网联安全，也对汽车召回监管提出了新挑战。2015年，因黑客远程控制，菲亚特克莱斯勒宣布召回约140万辆存在软件漏洞的汽车。同年，因ConnectedDrive功能存在漏洞，宝马宣布召回约220万辆汽车。

可以预计，在黑客之剑与防火墙之盾的较量中，会有更多厂家产品的漏洞被发现。如何及时有效进行召回监管，确保公共安全至关重要。与汽车召回相关的法规也应当及时升级，把召回监管的触角覆盖到汽车网联安全。

（参考链接：http://lw.xinhuanet.com/2020-08/26/c_ 139318557.htm）

下一代全场景虚拟化平台StratoVirt

（王洪磊 整理）

StratoVirt是计算产业中面向云数据中心的企业级虚拟化平台，实现了一套架构统一支持虚拟机、容器、Serverless三种场景。StratoVirt在轻量低噪、软硬协同、Rust语言级安全等方面具备关键技术竞争优势。

为什么需要StratoVirt?

在数据中心领域，虚拟化是资源隔离的重要手段，能以多种粒度提供安全的虚拟运行时环境。传统的虚拟化软件Qemu存在代码量庞大、CVE安全漏洞频出的问题，业界逐步演进出以Rust语言实现microVM形态的趋势。安全、轻量、高性能、低损耗，组件灵活拆分，全场景（数据中心、终端、边缘设备）通用的虚拟化技术是未来的趋势。

作为华为自研的下一代全场景、性能领先的虚拟化平台，StratoVirt以极致轻量快速的特质、灵巧的组件配置能力，为轻量虚拟化场景注入澎湃动力，同时也为向标准虚拟化场景的演进提供了无限可能。

StratoVirt优势

更轻、更快、更强！StratoVirt具有极大的行业竞争力，在于

·强安全性与隔离性

· 采用内存安全语言Rust编写，保证语言级安全性；

· 基于硬件辅助虚拟化实现安全多租户隔离，并通过seccomp进一步约束非必要的系统调用，减小系统攻击面；

·轻量低噪

· 轻量化场景下冷启动时间<50ms，内存底噪<4M；

·高速稳定的IO能力

· 具有精简的设备模型，并提供了稳定高速的IO能力；

·资源伸缩

· 具有ms级别的设备伸缩时延，为轻量化负载提供灵活的资源伸缩能力；

·全场景支持

· 完美支持X86和Arm平台：X86支持VT，鲲鹏支持Kunpeng-V，实现多体系硬件加速；

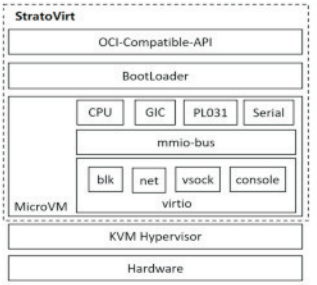
· 可完美集成于容器生态，与Kubernetes生态完美对接，在虚拟机、容器和serverless场景有广阔的应用空间；

·扩展性

· 架构设计完备，各个组件可灵活地配置和拆分；

· 设备模型可扩展，可扩展PCIe等复杂设备规范，实现标准虚拟机演进；

StratoVirt架构



麻雀虽小，五脏俱全。StratoVirt核心架构自顶向下分为三层：

OCI兼容接口：兼容qmp协议，具有完备的OCI兼容能力。

BootLoader：抛弃传统的BIOS + GRUB启动模式，实现了更轻更快的BootLoader，并达到极限启动时延。

MicroVM：充分利用软硬协同能力；精简化设备模型；低时延资源伸缩能力；

StratoVirt未来

StratoVirt的发展路标为，通过一套架构，支持轻量虚拟机和标准虚拟机两种模式：

轻量虚拟机模式下，单虚机内存底噪小于4MB，启动时间小于50ms，且支持ms级时延的设备极速伸缩能力，当前已经开发完毕，2020年9月已经在openEuler社区开源；

标准虚拟机模式下，可支持完整的机器模型，启动标准内核镜像，可以达成Qemu的能力，同时在代码规模和安全性上有较大优势。

（参考链接：https://openeuler.org/zh/blog/wangzhigang/ 2020-09-25-stratovirt-intro.html）

Docker发布2021开发者趋势

（柏东山 整理）

由于全球流行的新冠疫情，人类活动的方方面面逐步从离线向在线转移，这给开发团队带来了新的机遇和挑战。SCOTT JOHNSTON发布了2021年与开发者相关的三大趋势，主要包括分布式协作、微服务采用率以及AI/ML的加速普及。

第一是开放的分布式协作，2020年Docker团队对数千名Docker开发人员工作方式的调查显示，有51%的人更喜欢远程工作。尽管依然存在着挑战，但是已经可以看到拥抱分布式协作的团队可以取得的成功，并且还可以利用这种方式完成以前无法做到的事情。远程开发工作预计将在 2021 年完成。为实现远程团队成员间一致的开发环境和应用程序堆栈。其开发人员会在 Docker Desktop 本地使用 docker-compose.yml 从 Docker Hub 定义并提取其团队的应用程序堆栈共享图像。

第二是更多的微服务应用，降低复杂性。SCOTT JOHNSTON 认为，微服务应用在 2020 年证明了其价值，特别是压缩了从设计到生产的时间，并能快速响应需求。在 Docker 开发人员社区中，有 65% 的人表示他们的组织已经在使用微服务。

第三是加速 AI/ML 的普及。Docker简化了开发团队和数据科学家对AI/ML工具的采用，帮助他们构建了可共享、可复制的环境。尽管 35% 的调查受访者表示，他们的组织正在使用机器学习，但依旧有许多开发人员将 ML 作为他们最希望组织采用的技术。因此 Docker 团队预计，2021 年开发团队在其应用程序中使用 AI/ML 工具的情况会有所增加。

（参考链接：https://www.docker.com/blog/top-developer-trends-for-2021/）

新型光量子计算芯片在这里诞生

（李星儒 整理）

国防科技大学计算机学院QUANTA团队，联合军事科学院、中山大学等国内外单位，研发出一款新型可编程硅基光量子计算芯片，实现了多种图论问题的量子算法求解，有望未来在大数据处理等领域获得应用。国际权威期刊《Science Advances》（《科学进展》）于北京时间2月27日发表该成果。

具有实用价值的量子计算机，是量子计算领域最重要的发展目标。然而，在量子比特数目少、有效量子操作深度浅等现阶段量子技术水平下，如何最大化利用量子资源、设计可编程运行有实用前景量子算法的量子装置，是量子计算领域的重要挑战。

国防科技大学、军事科学院、中山大学等国内外机构研究人员，研发出可编程运行图论问题量子算法的光量子芯片，提出可动态编程实现多粒子量子漫步的光量子芯片结构。基于所提结构，采用硅基集成光学技术，设计实现了可编程光量子计算芯片。芯片上集成了纠缠光子源、可配置光学网络等，通过电学调控片上元件实现对光量子态的操控，从而实现量子信息的编 和量子算法的映射，具有高集成度、高稳定性、高精确度等优势。

研究人员还通过对所研制光量子计算芯片的编程运行，演示了顶点搜索、图同构等图论问题量子算法的求解。随着芯片规模和光子数目的增加，芯片可支持实现的图问题规模快速增长。该成果展示了硅基光量子芯片技术实现特定量子计算应用的巨大潜力。

（参考链接：https://advances.sciencemag.org/content/ 7/9/eabb8375
https://www.nudt.edu.cn/kxyj/fe203f208d8542d8a 635c690c3ef99c2.htm）

Informer—— 效果远超Transformer的最新长序列预测模型

（罗子涵 整理）

2021年2月2日至2月9日，AAAI Conference on Artificial Intelligence 通过虚拟会议的形式成功召开，其中由北京航空航天大学的周号益等人发表的论文：Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting 成功获得本届会议的最佳论文奖。

该论文聚焦于解决长序列预测问题，而长序列预测（Long Sequence Time-Series Forecasting）任务的研究在政策计划和投资避险等多种长序列情景下具有重要意义。然而，目前的方法多专注于短期预测，模型缺乏长期预测能力。虽然Transformer 具有较强捕获长距离依赖的能力，但是，在计算时间复杂度和空间复杂度以及如何加强长序列输入和输出关联上都需要优化。本篇论文所提出来的Informer 的主要工作是使用Transfomer 实现长序列预测。Informer模型针对Transfomer 在长序列预测中的不足：1. Self-attention 的平方时间复杂度；2. J个encoder/ decoder的stack会导致内存的使用为O(J・L2)，存在由长输入导致的内存瓶颈；3. 预测长输出的速度骤降，分别使用 ProbSparse self-attention机制、self-attention distilling操作和生成式解码器等模块解决或缓解上述问题。在四个长序列预测数据集上的实验结果表明，时间复杂度和内存使用率上分别从原来的O(L2)和O(J・L2) 降至O(LlogL)的情况下，Informer还能在多数数据集下取得最优表现。Informer模型成功地提高长序列预测问题的预测能力，验证了类Transformer 模型的潜在价值，为长序列预测问题提供了一种新的解决方案。

参考文献：

Zhou H, Zhang S, Peng J, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In: Proceedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence 2021.

顶级云计算平台面临“云水坑攻击”

（黄 娇 整理）

2021年2月26日，云计算网络弹性专家Accurics发布了最新研究报告《Accurics云计算网络弹性报告》，根据报告显示伴随托管基础设施的云服务快速增长，“云水坑攻击”呈现爆发式增长。

所谓水坑攻击就是黑客利用平台弱点，预先“蹲点埋伏”，向使用平台服务（例如网站等公共网络资源）的最终用户分发恶意软件，未经授权访问其生产环境、数据或完全破坏目标环境。

根据报告，在已经发现的所有“云水坑攻击”事件中，有23%针对配置不当的托管服务产品，所谓的配置不当，主要指使用默认安全配置或提供过多权限的错误配置。

研究人员指出，在云环境中，水坑攻击可造成更大的破坏，因为托管的云中开发流程暴露于互联网中，而不是像内部环境中的开发流程那样被隐藏在组织内部。当不法分子成功利用云端开发管道中的错误配置时，不仅会给公司造成灾难，还会给客户造成灾难。为缓解此风险，企业应假定整个开发过程都易于被非法访问，并遵循最小化权限原则，将访问权限限制为仅向需要它的用户开放。

开发上云已经是不可逆转的趋势，越来越多的团队正在加速采用托管服务，这肯定会提高生产力并提升开发速度。但不幸的是，一些团队的安全意识和能力无法跟上相关的风险——例如使用默认的安全配置文件和过度授权。

报告指出：根据历史经验，就像几年前存储桶业务所经历那样，消息服务和FaaS也正进入网络安全问题集中爆发的危险阶段，这些服务的不安全配置将导致更多的违规行为。

研究表明，在所有环境中，威胁缓解的平均时间（MTTR）为25天，这给潜在的攻击者留出了极为宽松的时间窗口。MTTR对于云安全来说特别重要，因为它与偏离（drift）有关，当运行时（runtime）的配置发生变更时，会导致云风险状况偏离已建立的安全基准，而偏离安全基准的MTTR约为8天。

甚至随着时间的流逝，那些在配置基础设施时已经建立安全基准的组织也会产生偏离，例如案例亚马逊AWS S3存储桶。2015年AWS S3存储桶被添加到云环境时的配置是正确的，但五个月后，为解决问题而进行的配置更改在工作完成后并未正确重置，直到近五年后，这种偏离才被发现并得到解决。

云基础架构面临的主要风险：

1）尝试部署基于角色的访问控制（RBAC）的Kubernetes用户往往未能以适当的粒度定义角色。这增加了账户重用和滥用的机会。实际上，有35%的企业和机构在这方面表现糟糕。

2）在Helm图表中，有48%的问题是由不安全的默认值引起的。最常见的错误是对默认名称空间的不正确使用（在其中运行系统组件），这可能使攻击者可以访问系统组件或机密。

3）报告首次在生产环境中发现通过基础结构即代码（IaC）定义的身份和访问管理，并且此报告中检测到的IAM偏离中有超过三分之一（35%）源自IaC。这表明IAM即代码（IAM as Code）正在快速流行，但可能导致角色配置错误的风险。

4）硬编码的机密信息几乎占违规行为的10%，其中23%是因为用户错误配置了托管服务产品。

5）在接受调查的企业中，有10%实际上为从未启用付费购买高级安全功能。

6）虽然修复基础设施配置错误的平均时间约为25天，但基础设施中最关键的部分通常需要花费最多的时间来修复。例如，负载平衡服务平均需要149天的时间才能修复。由于所有面向用户的数据都需要流经负载均衡服务，因此理想情况下，应该优先以最快的速度修复此类资产。

保护云基础架构需要一种全新的方法，必须在开发生命周期的早期阶段嵌入安全性，并始终保持安全状态。企业需要持续监控云基础设施运行时段配置变更并评估风险。在配置变更带来风险时，必须根据安全基准重新部署云基础架构，这样可以确保意外或恶意进行的任何更改都会被自动覆盖。

（参考链接：https://mp.weixin.qq.com/s/Xmk 7pKn FzgP-tqM0kXfawA）

行至水穷处，坐看云起时

楔　子

十分荣幸能收到实验室的邀请，成为本期的封面人物。时光流逝，离开实验室已过半年，但实验室的点点滴滴历历在目，恍若昨日才发生。清楚记得第一次与实验室结缘是在东五楼会议室，聆听金海老师和廖小飞老师主讲的招生宣传。当时我刚从本科的填鸭式教学中脱离，对科学研究的认识与刚出生的孩子无异，听到金老师和廖老师介绍实验室的成果时，无比向往与憧憬。正是这第一次的惊叹，才有了后面的一系列故事。在那之后，我选择加入了云组刘方明老师的麾下，正式成为CGCL的一份子，这也为我的未来烙上了东五楼的痕迹。

1. 转身

和那些在本科阶段就早早踏入实验室做起研究的同学相比，我属于很晚起飞的鸟。我是在大四上学期保研结束之后，才在OpenCloud NeXt小组开启我的研究之路。在我们小组，刘方明老师会为每个将要从事研究的同学安排一个较为前沿的课题作为毕设题目，在完成毕业设计的同时，对“科研是什么”这个问题也逐渐产生了清晰的认识，渐进成长。我们组有许多优秀的先例，做完毕业设计后就孵化出了棒棒的成果，经过“修饰”之后就能发表在顶会上。我当时选取了跨域数据处理作为毕设题目，跟随小组优秀的周知学长学习和科研。当时，我们的期冀是做一个理论与系统并重的跨域数据处理系统，刘老师也积极支持我们，在亚马逊云上租了多个地区的节点供我们开发。我们完成了原型的开发，并由此产出一篇文章。因为经验不足，文章辗转投了三次，尽管得到了一些审稿人的赏识，尽管我们对文章修改了很多次，但文章的质量确实未能达到顶会的要求，因此一直未能上榜。随着时间的流逝，文章的新意也在逐渐淡化，因此在对这篇文章进行修改的同时，我转而思考新的方向和idea。因为有了前面科研能力的积累，后续无论是思考还是写作，都有了极大的提升，在积累之中，新idea新paper在INFOCOM上一击就中。却道偶然，实是必然。

2. 养性

科研作为本分，理应极尽努力，但是科研以外的部分也应当重视。首先，要“保重”身体。研究生阶段，是一个考验主观能动性的阶段，很多事儿由自己安排和计划，部分同学开始放飞自我，经常性熬夜，饮食不规律、不均衡，导致身体产生了很多隐患，这些隐患往往易燃易爆，在高强度的科研工作下很容易被诱发，甚至变成一辈子的毛病。因此，要特别注意养成良好的生活习惯，良好的身体是支撑好的科研成果的必备基础。其次，要学会维护自己的精神状态。科研之路有很多荆棘，往往会使人精神憔悴，这会助长负面情绪的嚣张气焰。当你觉得自己卡在某个瓶颈无法突破，整个世界都变得不美好时，不妨放下手中的活，约三两好友，找一点乐事去做，等神经放松，再次转身回头，也许原来的困难就会迎刃而解，之前只是自己被自己的精神状态卡脖子了。

3. 善于请教，广于交流

一项工作能否既快又好的完成，不仅依赖于科研人员的个人能力，也与所在平台息息相关，很多时候，科研也是一种继承、发扬和传承。首先，选择一个实验室、选择一个导师，意味着继

承。实验室和老师的研究风格与研究品位决定了一个人可以飞多高。CGCL实验室无疑是国内乃至国际上头等的顶尖平台。其次，选择了一个平台，就要善于利用平台的软硬资源。东五既有良好的硬件实验平台，也有超高质量的师生团队，当你遇到难解决的问题，要多与老师和同学交流。大多时候，我们遇到的问题，师长们都曾遇到过，他们往往可以一针见血的解决疑难杂症，节省大量钻牛角尖的时间。

4. 天才是努力的结果

保持刻苦工作踏实学习的态度，是成功的第一步。成功不止需要天分，还需要极多努力和汗水。天分从踏入华科的那一刻起，就已被证明不缺，因此努力至关重要。你永远不知道成功者在背后做了多少努力。曾经同带我的周师兄聊天，印象最为深刻的不是他的指导，而是他告诉我，自己曾经在和女朋友逛街时也带着paper，在女朋友试衣服的间隙阅读。这使我极为震撼，别人都说某个人如何的聪明，取得了多少成绩，但是并不知道其背后付出了多少努力。因此，要在能够努力的年纪，付出足够的努力，成功便握住了一半。有时候你会觉得自己做的很多事都是无意义的重复，甚至是钻牛角尖，但是这些过程是必然会经历的，当你经历之后，就不会重蹈覆辙。比如在我们组，刘老师对我们组的科研有着极高的要求，无论是工作的品味，还是组会的presentation，或是论文的草稿，都要求极力做成一件珍品，就像打造一件传世的艺术品一样。记得我第一次写完论文草稿，当时自我感觉不错，但是刘老师依旧锐利地从文章的多个方面挑出来很多问题，包括错别词、写作、图片，甚至是文章排版等等都指出了存在的短板，然后教导我打造文章要如同打造艺术品一般，方方面面都要做到无可挑剔。虽然，当时不太理解老师的良苦用心，觉得他过于挑剔，但是在经历那个挑剔的阶段之后我才发觉我无论做什么写什么，都会自然而然注意很多细小的问题，高要求极力做到最好，这些东西已经渗透进了我行为做事的方方面面中。很多时候，成长就在不经意间。Every step counts!

5. 正确对待科研

很多人其实在进入实验室之前都没有正确认识科研，读研也是亦步亦趋，当真正面对科研时，逐渐兴趣寡淡，进而消极对待。我想，在生活中，每件事都是自己喜欢做的概率极小，当不得不做时，我们要转变视角，思考能从现在做的事情中获得什么成长，毕竟成长是终身的。实际上，就科研来说，无论未来的计划是继续学术之路还是投身工业界，在东五楼完成的科研训练都能让人在未来将要扮演的不同角色中游刃有余。在科研训练的过程中，我们得到的是全方位的成长和提升，经历了发现问题、抽象问题、解决问题等阶段，在这个过程中，我们获得的是一套思维体系和方法论，这些能力在未来生活工作面临的挑战中，是我们具备迅速制敌致胜的法宝。

致谢

衷心感谢实验室的金老师和诸位老师，能够为我研究生的成长提供一个强大的平台，使我有了飞翔的起点；感谢认识的同学们，你们的陪伴和激励，是我不断前进的动力；感谢OpenCloudNeXt小组和刘方明老师，在小组学到的一切使我一生受用。也祝愿各位CGCLers前程似锦！

高　彬
2017级硕士研究生
研究方向：云计算网络
Email：bingao.cs@gmail.com

分布式系统组研究规划

吴　松，刘方明，余　辰，肖　江，顾　琳，姚德中

本方向在2021年将重点围绕内核虚拟化、容器核心系统、边缘计算、区块链等方面开展研究工作。包括操作系统内核虚拟化、容器网络的控制和管理、数据容器系统、边缘计算数据中心系统结构、多任务边缘设备协同、智能边缘计算系统、高并发区块链体系架构等。主要研究规划和重点关注的研究内容如下。

（1）云计算环境下操作系统内核虚拟化方法

在云计算平台中，随着运行环境的容器化、应用需求的多样化、硬件系统的异构化，传统的通用服务器操作系统内核架构难以满足同时运行的多类型应用及其特殊需求，从而造成云平台服务质量下降。在本年度开始执行的国家自然科学基金重点项目支持下，拟重新思考和分析服务器操作系统内核整体架构，研究探索操作系统内核虚拟化方法。总体研究思路是：分析云计算环境下服务器操作系统内核架构面临的挑战，研究内核虚拟化的抽象模型与架构设计，并基于此分别研究面向应用特征和体系结构特征的内核虚拟化方法，灵活满足云计算环境下多种应用和硬件环境对操作系统内核的特性需求。

（2）容器网络的控制和管理研究

容器作为一种轻量级虚拟化技术，已被广泛认为是未来在线服务的发展趋势。容器技术以更少的资源实现更快的服务部署，使得平台从提供服务向提供功能的转变，提供更高的运行效率，但其资源弱隔离的特性使得服务性能无法保障。尤其在服务需求呈现高度的动态性，多源数据和多种设备交互日渐复杂的情况下，多模式下的容器网络控制和管理呈现出高度的复杂性。为提供跨模式智能化的容器网络管理技术主要需要解决以下三点：1）构建多资源跨模式的网络控制框架。分析和总结现有的容器网络模式，提出一个基于容器的扩展ID结构追踪容器的流量，实现在不同的网络模式下容器流量的可见可控。2）建立多容器跨节点的性能分析模型。挖掘多个网络节点、服务关联容器、不同资源之间的竞争对系网络性能的影响，建立容器系统性能的理论分析模型，通过现有的性能参数信息反推网络的预期性能，实现提前预警和控制。3）实现多租户智能化的网络调度算法。对容器的历史数据进行分析，提取用户的数据操作行为特征，针对不同容器实现生命周期、启动频率、任务负载和用户习惯等方面的容器画像，设计智能的化容器编排和多租户资源分配算法，来满足不同服务QoS需求。

（3）分布式环境下数据容器关键技术

针对大数据处理的流式特征和实际需求，研究数据流处理任务的数据容器理念、技术与系统，探索面向特定应用的大数据处理网络化、平台化、通用化的新型运行模式。数据容器是用来处理海量、动态、实时流式数据的专用容器系统，主要研究内容包括：1）数据流特征驱动的容器动态垂直扩展技术。容器系统配置或者其数据处理能力可以随着数据流特征的变化而动态变化，解决数据流处理任务的弹性扩展问题。2）针对数据流处理任务的容器镜像管理技术。容器系统可以通过分级镜像管理和版本控制，满足容器中运行任务的参数多样性需求，解决数据流处理任务的动态配置问题。3）数据容器的低延时部署与启动技术。容器系统可以将数据流处理任务快速部署到边缘节点，解决数据流处理任务的低延时部署与启动问题。

（4）边缘计算数据中心高效能系统结构

为了达到边缘数据中心异构低耗的目标，需要基础设施原生优化与配套设施辅助优化齐头并进。一方面，需要探索边缘数据中心在空间和资源约束条件下对CPU、GPU/XPU、FPGA、Smart NIC等多源异构硬件的高密部署方式，同时根据不同硬件特性协调多样化边缘计算负载，并优化冷却散热方式，从而联动降低IT系统能耗和供电制冷等非IT设施能耗。另一方面，能耗和成本约束促使边缘数据中心拓展与智能电网、城市三联供等其他基础设施之间的交叉生态机遇和辅助优化手段，例如通过需求响应、废热回收利用等方式缓解电网压力、降低用电成本、提高能源利用率，拟为边缘数据中心集约化、协同化、绿色化发展提供坚实基础。

（5）边缘计算数据中心网络功能虚拟化

为了应对广泛多样的边缘应用场景，需要在边缘数据中心跨域互联的基础上，进一步结合新兴的5G和网络功能虚拟化技术构建按需切片和编排的虚拟网络功能服务链部署与调度方案，满足差异化的应用性能需求。例如，车联网和自动驾驶等应用对通信时延要求极为严格，若将虚拟网络功能部署在同一服务器上会产生性能干扰，造成网络吞吐量下降，甚至影响用户安全。为此，需要降低共置虚拟网络功能的性能干扰，通过自适应部署的虚拟网络功能来实时响应车联网服务请求。进而，面对不同流量特征和不断变化的网络服务请求，需要自动对网络功能服务链进行在线动态调度，在降低资源占用与开销的同时提升网络吞吐量。总之，跨域互联、切片组网与按需调度互相配合，拟为实现边缘计算数据中心分布式跨域协同目标提供核心技术。

（6）面向边缘容器的深度网络模型切分技术

将深度网络模型计算任务从云端迁移部署在边缘侧，不但可以提高任务的响应时延，同时可以提高可靠性，避免由于云服务或网络问题造成的服务不可用。然而，边缘侧设备由于受到成本、体积及功耗的限制，计算和存储能力往往弱于云端服务器，因此许多复杂的深度网络模型无法直接部署在边缘设备端。考虑到深度网络模型的层级结构，因此可以采用模型拆分的方法，把深度网络模型分解成不同的任务单元。基于此思路，本方向将研究深度神经网络模型切分技术。首先针对深度神经网络模型结构进行拆解，定义给定网络带宽及设备算力下的最小任务单元，使得切分的任务单元可以在边缘侧设备上执行，同时确保网络带宽满足各任务单元输出结果的传输时延。然后，研究采用轻量级容器化的任务单元的部署方案，将切分的任务单元部署到异构的边缘设备端，使得深度网络模型更具有普适运行能力。

（7）面向多任务应用的边缘设备协同技术与系统

随着物联网技术的蓬勃发展，各类终端采集设备的数量剧增，其源源不断产生的数据对海量数据处理及智能分析的实时性提出了新的挑战。将计算任务从云端迁移到边缘侧可以提高响应时间和可靠性。然而边缘侧设备资源远远比不上云端服务器。为了充分利用边缘侧资源，将针对多任务应用的不同执行阶段，设计合理的任务调度机制，实现各个边缘设备负载均衡，并研究分布式机器学习与深度学习任务在边缘环境中的性能优化，从而减少任务执行的延迟、成本与能耗。然后，考虑到边缘侧设备的移动性，将研究边缘设备的接入、接出机制，建立完整的资源与任务管理策略，实现边缘设备的动态迁移。最后，将以人脸识别典型应用为例，研究监控场景下多摄像头的人脸识别在边缘场景下的部署方法，实现边缘设备的协同系统原型，验证提出的核心技术、机制及策略的有效性。

（8）面向边缘环境的联邦学习技术

边缘计算系统一般由边缘网关以及多个传感设备组成，传感设备负责数据采集，然后在边缘网关汇集分析。边缘数据种类繁多而且来源各异，在边缘端应用深度学习训练来分析处理这些数据面临以下挑战：1）来源分散，格式

不同；2）数据更新频率不同，不均衡，存在缺失；3）数据具有统计异质性；4）数据具有多模态特性。并且有的设备采集的视频数据体积较大，将多路数据汇集于一个边缘设备进行学习训练，会超出边缘设备的计算能力，而且对敏感数据考虑到隐私安全问题，不便于将数据传输到其他设备上再进行学习训练。针对这些边缘分布式数据，通常采用联邦学习的框架进行分布式训练，但这虽然避免了数据在设备间传输，模型仍然需要传输至中心服务器汇集训练学习后的结果，因为深度模型体积较大，模型更新的参数较多，每次训练模型更新汇集通信仍然对网络造成较大压力。针对上诉挑战，本方向研究工作拟从两个方面展开：1）研究模型稀疏化技术，通过对训练模型稀疏化约束，减小深度模型体积，降低模型更新的参数，从而减少通信代价，提高整体训练速度；2）研究全局模型与客户模型初始化、权值更新汇集和小样本共享策略，完成对不平衡、非独立同分布的边缘数据样本学习训练不会陷入局部最优。

（9）高并发新型区块链体系架构的设计理论与方法

当前全球主要国家都在加速区块链技术的布局，区块链应用场景的覆盖范围愈发地广泛。针对当前区块链难以支撑实际应用场景的高频次高并发需求，从体系架构方面进行创新，本方向拟研究新型区块链体系架构的设计理论与方法，具体包括：1）研究高并发的图链混合区块链体系架构：混合采用链式结构和图式结构构建区块和交易拓扑关系，设计适用于图链混合架构的共识算法，提出动态的交易验证机制和确认机制，并设计高效的网络传输方法和通信机制，多维度突破可扩展性的瓶颈，满足不同应用场景的实际需求；2）研究支持原子互操作的多链协同互通的可信数据共享：针对异构区块链之间因多维异构性导致互联互通难的问题，提出高效安全的跨链网络传输协议和事务处理与验证机制，并设计可信协同的跨链共享方法；3）研究面向新型体系架构的数据查询分析方法：通过数据裁剪实现冷热数据分离，建立高效的索引机制，优化链上数据存储与查询，同时结合链上链下数据，对通用交易数据模式特征进行挖掘，实现对非法交易行为的检测识别。

吴　松
教　授
研究方向：云计算、虚拟化、操作系统
Email：wusong@hust.edu.cn

刘方明
教　授
研究方向：云计算与边缘计算，数据中心与绿色计算、分布式系统与网络
Email：fmliu@hust.edu.cn

余　辰
教　授
研究方向：普适计算
Email：yuchen@hust.edu.cn

肖　江
副教授
研究方向：区块链、分布式计算与系统、无线定位与智能感知
E-mail：jiangxiao@hust.edu.cn

顾　琳
副教授
研究方向：网络功能虚拟化
Email：anheeno@mail.com

姚德中
副教授
研究方向：普适计算、物联网、分布式机器学习
Email：dyao@hust.edu.cn

支持大数据高效处理的新型体系结构和系统软件

——体系结构与系统软件组科研规划

廖小飞，蒋文斌，邵志远，郑　然，刘海坤，郑　龙，张　宇，毛伏兵，叶晨成

1 背景与挑战

大数据分析处理主要从计算和存储两个方面对计算机体系机构提出更高层次的需求，也极大地推动了相关技术的快速发展。计算方面，诸如FPGA、ASIC、RISC-V、chiplet、数据流体系架构等新型处理器、加速器架构的出现和发展，在为大数据处理提供越来越强的的计算资源的同时，也为其提供了处理器的高可定制性、高能效比以及系统的高可扩展性。存储方面，诸如PCM、ReRAM等新型存储器件技术的快速发展，极大地提高了大数据的存取速度和存储容量，为实现低延迟、高吞吐的大数据处理提供了可能。

然而，总的来说，计算机体系结构的发展仍然明显不能满足快速增长的大数据分析处理的需求（如图计算、图神经网络等典型大数据应用），无论是在系统架构、编程模型、数据放置，还是在运作机制和任务调度上都存在巨大的挑战。

1）各类大数据应用高时效、高可扩展处理需求对处理器体系结构提出的新挑战

万物皆关联，图是表达和处理不同实体间的内在关系的高效数据结构。过去几年，我们在图加速器方面已取得初步进展。但仍然面临如下三类重要挑战：（1）高可扩展的硬件架构设计方案。现有FPGA图加速器由于集中式的访存设计，通常流水线数目较少。有限的并行性难以充分利用新兴存储设备（如HBM较传统内存可提供近10倍的速度提升）带宽特性，为此探索高可扩展的硬件架构设计方案具有极其重要的现实意义；（2）图加速器的软件编程与编译工具链。面对纷繁复杂的图分析应用场景，迫切需要研究编程环境，编译器支持和工具链，支持“由软件定义的硬件”，即从软件规范中定义图计算加速器的设计来针对不同的场景和应用进行优化；（3）突破简单图计算的研究范畴。现有研究主要集中于简单图计算，但现实图应用往往都非常复杂。首先，图的应用在不断变化发展，如新兴的图神经网络对计算和访存都提出了更高的需求。其次，很多应用图的结构是随时间动态变化的，而现有的研究大多限于静态图结构。动态图计算是一个热门研究课题。

另一类挑战来自于新型RISC-V指令集的出现带来的体系结构研究的新问题。主要集中在面对新型大数据应用场景的定制优化方向。主要包括：（1）在现阶段，RISC-V的目标应用领域主要集中在智能嵌入式系统（AIoT）上，针对这类设备，如何通过对RISC-V系统本身的定制，提高嵌入式应用的按时响应，降低不满足实时要求的响应比例是重要的研究议题；（2）如何对芯片设计进行快速全系统验证也是重要的研究议题。处理器芯片以及围绕处理器芯片所设计的软件系统的全系统验证，从来都是各大芯片公司所关注的重点和难点问题。如

果芯片在设计阶段的行为与流片后的表现不一致，将导致大量的投资损失。利用RISC-V的可定制特点，结合芯片设计后端的知识、软件工具，对所设计的计算机系统进行快速且有效的全系统验证，也是伴随RISC-V兴起的新研究问题。

另外，芯片设计方面，chiplet技术的发展在给芯片设计领域带来巨大机遇。模块化集成不仅有效地加快了开发速度，而且降低了开发成本和门槛。然而，chiplet的成熟和广泛应用仍面临许多挑战。主要表现为：（1）统一接口和标准对于异构集成系统至关重要。各种异构芯片互连接口和标准的设计在技术和市场竞争方面难以平衡性能和灵活性。例如，芯片设计商难以为每个互联结构设计芯片；（2）chiplet的核心封装技术在性能，功耗，成本等方面面临挑战；（3）支持芯片设计和电子设计自动化（EDA）工具链实现难度很大，而这些工具决定了chiplet生态系统是否完整和可持续。

大数据的高效实时处理的需求同样给基于数据流的体系结构带来了诸多的挑战。主要表现为：（1）大数据特征的多样性及其分析处理方法的差异性对数据流体系结构的设计提出了严峻挑战。如图计算任务的处理与深度学习任务的处理就存在很大的差异性，而对于图神经网络任务则又存在明显的不同。因此，需要根据应用领域的不同，设计一套灵活支持多领域大数据应用的数据流体系结构；（2）资源的复杂性和不同任务的异构特征对数据流计算任务的调度机制提出了新的挑战。当前，对于不同性质的任务和底层硬件资源，出现了不同级别的数据流并行模式（如指令级、程序块级以及线程级等）。当前，这些模式在很大程度互相独立，无法融合成一个有机的统一整体；（3）大规模数据处理的高时效性对数据流计算任务的分布式处理方法和策略提出了挑战。分布式环境下按数据流模式部署并行执行的任务，需要对大粒度的任务充分解耦，并充分考虑大数据计算任务的局部性和迭代周期等特征，设计数据流任务的高效处理和高通量调度机制。

2）高时效大数据处理需求对内存管理带来的挑战

数据的存储与管理是大数据高实时处理不可缺少的环节，当前同样也面临诸多挑战。随着新型非易失性存储器的逐渐应用，单台服务器的内存容量很容易扩展到几十甚至几百TB。构建跨节点的分布式异构内存池，并通过远端内存直接访问（RDMA）协议进行跨节点内存共享，将极大地提高异构内存资源的利用率，并满足大数据应用如机器学习、图计算等极大容量的内存资源需求，同时大幅减小并行分布式应用的网络I/O开销。如今，包括Intel、Microsoft、Mellanox等IT巨头都在探讨NVM内存的池化和共享方法，已成为学术界和工业界研究的一个热点问题。然而，构建高效易用的分布式异构内存池还面临NVM内存池访问延迟高、持久化开销大、应用移植困难等诸多挑战性问题。

（1）远端NVM访问较高的延迟会影响应用程序的性能，如何有效降低对远端NVM的访问开销？NVM内存池的性能将直接决定其应用价值。目前的NVM存储器还存在读写速度不对称，数据写入功耗较高，介质写次数存在上限的局限性，并且读写性能要差于传统DDR内存，目前尚无法完全取代DRAM。因此，将NVM与传统DRAM 相结合充分利用NVM 和DRAM各自的优势，构建大容量、高性能的异构内存池仍然是优化分布式内存池性能的重要途径。通过把远端NVM内存池中的热点数据缓存到DRAM缓存中，可以有效减少对远端慢速NVM的访问，提高应用的访存性能。

（2）跨RDMA网络的NVM内存持久化开销大，如何结合持久内存特性，设计新的RDMA通信协议和操作原语减少通信开销？对于非易失性持久内存，如何保证在系统突然断电或崩溃的时候，仍然能够保证数据的一致性是持久内存研究的一个核心问题。单机环境下，使用mfence原语及clflush原语可以协同保证数据的持久化，但是对于NVMM内存则会引入高达47%的性能开销。在RDMA网络环境下，由于网卡缓存的存在，使得批量数据持久化到PM缺乏顺序保障机制。目前跨网络的持久化往往通过客户端应用程序先写入，再读回校验码进行验证的方式来保证持久化的正确性，这种“写后读”的模式增加了网络通信开销，进一步放大了NVM内存持久化的开销。因而，为NVM内存池设计新的RDMA原语，保证NVM的远端持久化并提高持久化效率具有重要意义。

（3）如何提供简单易用的分布式异构内存池编程接口，并透明支持已有的遗留应用程序无缝移植到分布式异构内存池平台？绝大多数应用程序在编程时默认都使用本地的DRAM内存和文件系统。然而，在分布式异构内存池环境下，既有本地DRAM，NVM，还有远端DRAM和NVM，且RDMA编程本身比较复杂，给应用程序员带来了诸多困难。因此，为编程开发人员提供简单易用的API来使用分布式异构内存池至关重要。此外，传统的很多应用程序都是基于文件系统或块设备来访问数据的，它们无法直接移植到分布式异构内存池中运行。因此，在分布式异构内存池之上还需要提供内存文件系统的访问接口，以兼容传统遗留应用程序直接运行在分布式异构内存池上。

2 主要研究内容

面对上述体系结构各方面的挑战，本年度，本研究小组仍然继续遵照过去几年既定的研究路线，围绕图计算加速器、RISC-V处理器、异构内存系统、数据流体系架构、chiplet芯片设计等几个主要方面展开研究，具体如下：

1）围绕现实场景下图计算加速器亟待解决的高可扩展架构设计、编译系统综合和典型应用支撑三个关键科学问题，开展面向图计算加速器的从硬件架构到软件系统的全栈式研究。

第一，开展基于FPGA的高可扩展图计算加速器架构研究。现有的图加速器架构在部署至FPGA-HBM平台时，因为采用内存共享访问模式，出现了难以充分利用HBM高带宽的现象。我们计划采用更为简单的网状片上互联网络，最大限度地降低硬件开销，实现硬件架构的高扩展性。同时，通过高效的数据划分方法和网络路由机制，实现图计算应用在大规模流水线下的高效执行。

第二，开展图加速器的可编程性研究。可编程性是图加速器设计中的核心问题。针对图计算应用的高迭代、细粒度和低局部性等特征，手动设计参数化硬件模板。分析由图计算算符组成的操作序列，实现硬件模板的计算流程和片上资源调度方法。通过选取合适的硬件参数，来满足算法多样性对片上资源可配置性和可扩展性的需求。

第三，将进一步研究图计算加速器如何支持现实场景下复杂的图应用和动态的图结构。首先，以图神经网络为代表的新兴图分析应用对硬件架构设计提出了更高的计算和访存需求。我们将基于现有加速器架构结合图神经网络执行特征，优化访存机制，定制硬件处理单元以高效支撑图神经网络应用。其次，我们将研究适配图加速器的新数据结构以支持动态图，并定制算法库，以保证动态图加速器的高

效执行。

2）在面向大数据高时效处理的异构内存系统方面，研究基于RDMA的异构内存池系统，包括设计高效的异构内存资源管理机制和RDMA通信优化机制，基于可编程网卡设计高效远端数据持久化机制，设计可透明移植遗留应用的异构内存池编程接口和文件系统。

第一，研究分布式异构内存池的资源管理和访存优化策略。主要包括：（1）全局内存地址空间抽象和映射机制；（2）综合考虑应用访存特征、RMAD拓扑结构、和访存介质特性的全局最优内存分配策略、数据迁移策略和负载均衡策略；（3）RDMA网络通信协议下的热点数据监测策略和缓存机制；（4）远端NVM的RDMA写操作的优化机制。

第二，研究RDMA网络中持久内存（PM）的数据高效持久化机制。计划基于可编程RDMA网卡研究高效支持PM持久化的RDMA原语和通信协议，通过硬件原语来保障分布式环境下PM持久化的可靠性，并降低PM持久化的RDMA网络开销。

第三，研究简单易用且高效的分布式异构内存池编程模型。对RDMA操作和数据持久化机制进行抽象和封装，为应用程序员提供简单易用的分布式异构内存池编程接口，并为上层应用提供可选的数据并发控制、容错、一致性协议等策略库，降低应用程序员使用分布式异构内存池的难度。对于很多遗留应用，将研究兼容传统内存分配接口malloc和文件I/O操作的抽象接口，通过扩展编程库和分布式内存文件系统来支持对近端内存和远端内存访问的透明支持。

3）继续针对数据流体系结构当中硬件层面的灵活设计、执行层面的高效调度和分布式层面的高效处理三个方面，开展面向复杂大数据分析处理的体系结构的研究。

第一，对于数据流加速器，主要开展基于FPGA的高吞吐数据流处理模式及数据流加速器的仿真、面向大数据处理的高并发流水结构等方面的研究。从而进一步提升大数据处理的单机处理能力；对于多加速器互联，进一步开展加速器–主机快速通信与同步协议及基于数据流的非对称协同、计算与数据高效调度方法、基于高速通信协议的多加速器互联技术等方面的研究。

第二，进一步研究面向大数据高效处理的数据流执行引擎和软件环境。在数据流加速器多级存储与管理方面，计划进一步研究主存和加速器内存间的多级智能放置策略、加速器片上存储的数据替换及预取策略、访问特征敏感的数据流任务调度映射等；在编程与适配环境方面，继续开展统一中间层抽象设计及用户程序对异构计算单元的自动适配研究，并研究存算一体架构下的数据流并行编程模型及编程原语，并基于并行编程模型，设计适应于存算一体模式的数据流图抽象模型。

第三，研究分布式环境下大数据高效处理的数据流任务的动态划分与高效通信方法。研究针对数据与资源利用和网络拓扑感知的分布式数据流图动态划分策略，在此基础上，研究低复杂度分布式数据流在线任务动态划分方法，使得任务划分更能符合实际需要；研究子数据流图间的独立任务分布式调度策略，重点考虑数据流体系结构乱序执行的特点。

4）继续推动RISC-V体系结构与大数据应用需求结合的加速器研究，并考虑面向chiplet的成熟和广泛应用的挑战，研究基于chiplet的异构体系架构。

第一，结合大数据应用需求，围绕RISC-V软硬件层面的核心问题，进一步开展RISC-V处理器（大数据处理器，DPU）研发。

第二，开展基于chiplet体系架构的模拟器

研究。模拟器能非常容易的集成到加速器中，兼容不同的网络，并能够提供功能和寄存器传输级仿真，支持RSIC-V。模拟器的准确度能够通过RTL和SPICE进行验证。

第三，开展支持基于chiplet的芯片设计和电子设计自动化（EDA）工具软件链的前期研究，为设计和开发高性能、低成本、低能耗的处理器服务。

第四，利用chiplet的可扩展性、异构特性开展不同体系结构差异性的前期研究。利用chiplet组成不同的系统架构，分析不同系统架构间的差异，探索高性能，低延时，高可靠性，高质量的系统结构的设计方法。

第五，开展基于chiplet的编程的前期研究。考虑研究支持chiplet的操作系统，探索多个底层计算单元间的交互、管理。实现多内核操作系统管理异构计算资源。

3 小结

2021年，本研究方向将重点推进图计算加速器、异构内存、数据流体系结构研究，并在RISC-V、chiplet方向开展预先研究。

廖小飞
教　授
研究方向：系统软件、多核与虚拟化、体系结构
Email：xfliao@hust.edu.cn

蒋文斌
教　授
研究方向：并行分布式计算、深度学习系统、计算机网络
Email：wenbinjiang@hust.edu.cn

邵志远
教　授
研究方向：可重配置计算、计算机体系结构
Email：zyshao@hust.edu.cn

郑　然
副教授
研究方向：并行分布式计算及应用
Email：zhraner@hust.edu.cn

刘海坤
副教授
研究方向：虚拟化技术、内存计算
Email：hkliu@hust.edu.cn

郑　龙
副教授
研究方向：大数据处理体系结构及系统软件
Email：longzh@hust.edu.cn

张　宇
副教授
研究方向：大数据处理、编程模型、体系结构和系统软件
Email：zhyu@hust.edu.cn

毛伏兵
讲　师
研究方向：系统软件和体系结构、集成电路物理设计、机器学习
Email：fbmao@hust.edu.cn

叶晨成
博士后
研究方向：内存管理、局部性理论与优化、非易失内存
Email：yecc@hust.edu.cn

网络空间安全组研究规划

邹德清，羌卫中，徐　鹏，马晓静，代炜琦，文　明，胡胜山，杨已彪，王虹飞，袁　斌

2021年，网络空间安全组将重点围绕网络空间安全领域最新系列关键问题开展研究，主要包括：消费物联网安全、软件漏洞自动挖掘与利用、智能系统的数据与算法安全、区块链安全、云计算安全防护和基于机器学习的硬件安全攻防等六个方面。

一、消费物联网安全

近年来，物联网技术的飞速发展催生了消费物联网市场。区别于工业物联网，消费物联网是面向普通消费者个人或家庭的物联网应用，包含智能家居、个人健康监测、可穿戴智能设备等领域。目前，消费物联网市场处于快速扩张阶段，国内外的一流IT企业都纷纷推出各自的消费物联网系统（包括协议、产品和服务等）。然而，由于缺乏实现标准，目前的消费物联网系统的实现不尽相同，使用了大量自定义的设计。如此混乱的消费物联网生态存在诸多安全隐患，为此拟开展消费物联网安全研究，探究消费物联网生态的安全现状，发现其安全问题，并提出防御方案。具体研究内容规划如下：**（1）消费物联网中的供应链安全研究。**目前消费物联网设备厂商往往需要从不同的上游厂商获得不同的设备组件，如通讯模组、开发套件、云服务后台、移动控制前端等，通过组装这些组件来构建一个完整的消费物联网系统，在这个过程中，往往容易出现安全责任界限模糊、不同厂商的安全假设不同等安全隐患，为此开展面向消费物联网供应链的安全分析研究，总结典型的消费物联网系统构建模式，剖析其安全风险，提出防御方案；**（2）消费物联网的协议安全研究。**消费物联网中涉及许多不同的协议，在通信协议方面，既包括传统的WiFi、蓝牙等协议，也有Zigbee、Z-wave等专门适用于物联网应用的协议，在应用层协议方面，也包括HTTP、HTTPS、MQTT、CoAP等许多协议，而这些协议之间的交互在各类应用场景下将变的十分复杂，如多用户场景下，不同用户使用不同的协议来控制同一个设备，如此复杂的交互带来授权管理、用户管理、隐私保护等安全方面的挑战。因此，开展针对消费物联网协议的安全研究，分析协议内部以及协议交互时的安全问题，提出解决方案建议，推动消费物联网的标准化；**（3）消费物联网的实践安全研究。**目前，各个消费物联网的服务提供商往往会对现有协议（如MQTT、OAuth等）进行不同程度的改造，来实现各自的消费物联网系统。然而，厂商为了尽快推出服务以抢占市场，其对协议的改造和系统的实现往往没有经过严格的安全检查，存在安全隐患。为此开展面向消费物联网实践的安全研究，拟通过形式化验证、代码分析、漏洞验证等手段，对真实的消费物联网系统的安全性进行分析与固化，增强消费物联网的实践安全。

二、软件漏洞自动挖掘与利用

近年来，全球软件系统漏洞数量屡创新高，其变种多、体量大，且出现速度快。然而，漏洞响应及修复费时费力，导致其可受攻击时间窗口长。因此，开展针对软件系统漏洞自动化攻防的研究势在必行。本项目聚焦开源软件、操作系统内核安全，展开漏洞挖掘、漏

洞利用和漏洞可视化等研究，旨在构建一套自动化攻防系统，实现漏洞的精确检测、有效利用以及可视化分析。其具体研究内容包括以下五个方面：（1）**面向开源软件的漏洞数据自动采集与标注系统。**通过启发式规则智能化构建漏洞数据集，以减少人工标注成本，提高构建效率。针对软件代码行进行数据标注，可满足不同粒度漏洞数据需求。标注的数据集有助于开展漏洞的理解、挖掘、利用以及可视化等工作；（2）**基于系统调用依赖覆盖率引导的内核模糊测试系统。**通过系统调用依赖关系模型构建技术、覆盖率引导算法等，解决现有内核模糊测试系统性能受限和智能化程度低的问题，拟开发出以系统调用依赖覆盖率为引导的智能化、高效的操作系统内核漏洞挖掘原型系统；（3）**面向Java程序模糊测试的种子生成与优化系统。**通过种子生成与优化技术，提高模糊测试中测试样例的代码覆盖率，以检测出系统中的更多漏洞。与此同时，减少测试样例的冗余程度，进而提升触发程序崩溃和异常的速度，以提升Java程序漏洞挖掘的性能；（4）**面向C/C++程序的模糊测试的种子生成与优化系统。**通过定向模糊测试技术，对源代码进行静态分析来提取语义信息，进而指导模糊测试的种子变异过程。此外，通过结合种子能量调度算法，不断调整变异策略，以达到提升漏洞挖掘速度的目的；（5）**漏洞利用可视化系统。**操作系统内核利用可视化和应用软件攻击可视化。通过数据采集工具记录运行程序的状态，使用重放工具再现漏洞攻击过程，跟踪记录控制流和数据流，生成攻击思路和攻击流程的可视化报告；（6）**高效和鲁棒的恶意软件检测。**深度学习技术的应用展现了强大的泛化能力，可以高效地应对飞速增长的恶意软件种类和数量所带来的挑战。然而研究表明现有的基于深度学习的恶意软件检测方法健壮性较差。例如，静态检测方法容易被代码混淆技术干扰，常见的加壳技术会使得分类结果产生较大偏差；而基于行为的动态检测方法虽具较好的健壮性，但无法应对抗样本技术的挑战。为此，针对加壳技术和对抗样本技术的扰动，拟研究和设计一种具备高鲁棒性的基于深度学习的恶意软件分类器，结合神经网络和程序执行指令特征数据来快速、精确地检测恶意软件，以解决当前恶意软件检测存在的问题。

三、智能系统的数据与算法安全

智能时代已经来临，无处不在的智能系统，也暗藏着无处不在的安全缺陷。联邦学习作为一种新兴的人工智能理念与基础技术，能有效帮助多个机构甚至数百万用户在满足用户隐私保护、数据安全和政府法规的要求下，开展数据利用和机器学习建模。本研究拟针对联邦学习的若干重要安全问题，开展以下三个层面的研究：（1）**研究基础数据层安全。**现有联邦学习在数据隐私保护，特别是本地用户训练模型保护的方面，存在安全性弱、效率低、灵活度差等问题。例如，在真实场景下，服务器与某些用户可能合谋窃取其他用户隐私信息，大规模用户可能随机掉线导致全局模型无法聚合等等。拟研究基于同态加密、安全多方计算、差分隐私、可信硬件等新型密码学技术的模型安全聚合方法，保证用户更新模型等隐私信息不被泄露；探索大规模用户协同参与下的密钥建立、分发、更新、撤销等机制，提高联邦学习的灵活性与可扩展性；研究基于区块链的分布式数据完整性校验与一致性保障协议，探索联邦学习场景下的用户激励机制，构建完全去中心化的联邦学习框架；（2）**研究中间算法层安全。**由于参与用户种类繁杂、来源不清，聚合模型易遭受来自恶意用户的注入攻击、女巫攻击、后门攻击等，导致全局模型的准确率降低甚至存在恶意后门，极大降低了智能系统的可靠性。拟研究针对联邦学习各种攻

击手段的检测与防御技术，在保证用户隐私数据不被泄露的前提下，实现模型的高效检测与聚合；（3）**研究应用平台层安全**。联邦学习应用场景广泛，构建通用的联邦学习平台具有重要意义。拟研究适合大规模轻量级用户的隐私保护模块、攻击检测模块，搭建基于云的联邦学习平台，实现适应各种机器学习算法的快速开发与部署。

四、区块链安全

区块链被认为是第四次工业革命的关键技术，由于其拥有不可篡改、可追溯，以及在分布式系统中实现去中心化的信任机制的特点，区块链技术的研究与应用呈现出爆发式增长态势。但是目前区块链技术还不完美，本研究方向针对区块链安全上的一些问题开展研究，主要研究内容和规划如下：（1）**区块链中的身份认证与访问控制**。目前无钥环境下的身份认证问题很少受到关注，而传统的密码恢复方案如电子邮件、短信服务、可信任人员等多方验证的方案进行身份认证后重置密码，无法直接应用到去中心化的区块链系统。而已有的区块链的技术依赖于第三方钱包或者第三方服务器保管私钥的方法以及零知识证明、非对称加密等其他加密技术，存在隐私暴露风险以及第三方可信度问题。针对这种问题，为保障区块链中用户的资产安全，实现除用户私钥外的身份认证方法，通过基于属性的加密方式实现分布式环境下用户对于自身资产的细粒度控制；（2）**区块链数据一致性**。数据从中心化系统向区块链系统迁移的时候，由于网络因素或多机构上链造成的数据冲突，从而使得链上链下数据难以达成一致，同时由于区块链系统是去中心化结构，并没有传统系统中的可信中心，在特定场景下，难以保证上链数据的真实性。通过设计数据迁移机制，解决了区块链应用数据迁移的问题；通过设计区块链修复机制，解决了随机数据的共识问题和特定数据修改复原的问题；通过设计数据源可信保障机制，确保引入数据源的可信可靠；（3）**合约安全**。智能合约因与巨额数字资产挂钩，频繁遭受攻击，亟待全方位的安全防护措施。现有方案多集中在全程序漏洞的静态检测方面，但由于技术的限制如符号执行的路径爆炸、模糊测试的有效输入生成难题，无法保证合约不存在漏洞，因此也需要针对疑似漏洞的高效验证技术；另一方面，合约一旦上链便无法修改，即使出现漏洞也无法像修复传统程序一下直接打补丁，而且合约分布式、公开透明执行的特点要求漏洞合约必须被即时修复，以避免合约漏洞被抢先发现进而恶意利用，因此急需一种适应于区块链上的合约修复技术。为解决这两个问题，从合约漏洞定向安全验证和智能合约链上自动化修复两个角度进行研究，以结合现有合约安全技术，实现合约全方位的安全保障体系；（4）**区块链应用安全**。物联网数据在共享的时候，除了要保证数据的不被泄漏，还要隐藏共享双方隐私（如用户定位和属性等），以及数据共享出现安全问题和隐私泄漏后可追溯问责；而现有区块链结合密码学技术仅能实现数据的安全共享和追溯问责，在共享过程中用户隐私无法得到保护。为了解决这一问题，保护数据共享过程中的用户隐私，基于零知识证明的方法，与基于属性加密和区块链相结合，实现物联网环境下的数据安全共享和隐私保护。

五、云计算安全防护

云计算通过虚拟化技术，实现了硬件资源的共享，同时还能根据用户的需求提供弹性的服务，不仅节约了用户成本，同时也提高了资源的利用率。云计算也出现了新的发展趋势，同时也带来了新的挑战，本研究方向主要研究内容和规划如下：（1）**容器云环境中的动态安全防御**。容器技术和微服务架构的结合使得容

器云相较于基于虚拟机的云环境更高效与灵活。与基于虚拟机的云环境相比，容器云环境中镜像来源多样化、服务间隔离薄弱化、服务状态变更高频化。同时，微服务架构的使用进一步使得服务调度和扩展变得更加灵活，大幅增加了云环境的动态性和复杂性。这些为云环境带来高可用性的新特点也大幅地导致云环境中的安全隐患动态变化。为解决因容器云动态性对传统安全防御机制带来的影响，以及由动态性带来的新安全问题，通过基于软件定义和形式化建模的方法动态地定位容器云环境中的安全隐患，实现容器云环境动态可控的安全防御机制；**（2）Spectre推测性执行的攻击与防御。**诸多漏洞实例表明，对于现代处理器性能提升至关重要的推测执行等优化技术会触发推测性执行侧通道漏洞，这些漏洞违反了许多软件安全性机制所依据的安全性假设，这些机制包括操作系统进程分离、静态分析、容器化以及即时编译等。Spectre有别于传统的缓存侧信道漏洞，其影响面更广，缓解难度更大。针对Spectre推测执行攻击的安全问题，目前没有低开销的系统性防护方法。因此，设计和实现一套具有较低的开销、轻微地修改硬件、能全面抵御Spectre攻击的防御机制，对于提高现代处理器的安全性具有重大的意义；**（3）云管理平台安全测试。**诸多漏洞实例表明，云平台中虚拟机的快速创建和删除会触发竞争漏洞，这些漏洞没有及时地上报和修复会造成巨大的安全风险，而且引发的故障会在系统中通过云平台的组件传播，造成更严重的后果。针对竞争漏洞的安全问题，目前没有系统性的竞争检测方法。因此，设计和实现一套可集成于云平台的模糊测试框架，对云平台中存在的竞争漏洞进行系统、高效地检测，并及时地将检测出的安全漏洞进行上报，对于提高云平台的安全性具有重大的意义。

六、基于机器学习的硬件安全攻防

随着物联网时代的来临，安全问题主导着物联网的可持续发展。设备身份验证是解决物联网安全问题的关键技术。传统的设备身份验证基于高复杂度的加密/解密算法和高成本的密钥存储技术。但是，在许多物联网应用中，CPU，内存和电池电量等资源是非常有限的。因此，轻量级解决方案迫在眉睫，而物理不可克隆函数（PUF）是针对各种应用程序的最有希望的替代轻量级解决方案。自从2002年首次引入PUF以来已经提出了许多PUF结构，包括MPUF、rMPUF、XOR Arbiter PUF、IPUF等等，与此同时对于PUF的机器学习攻击理论的发展也从未停止，对于攻击PUF的最著名的机器学习可以分为两种类型：第一类攻击使用CRP的非重复测量，称为经典机器学习(CML)；第二类机器学习攻击是使用重复CRP测量的攻击，将这些类型的攻击称为基于可靠性的机器学习（RML）。2019年提出的IPUF是目前最为强壮的PUF结构，但是随后提出的新颖的基于逻辑回归的分治算法攻击对于IPUF进行了成功建模。就如其他安全原语经历的迭代过程，在最终找到可接受的安全解决方案之前，在强PUF领域，强PUF的设计与机器学习算法的攻击之间的博弈在未来的几年中不会停止。为此开展以下的研究：**（1）新型安全的强PUF结构设计的研究。**基于目前提出的PUF结构，针对于已有的的机器学习理论的攻击特点，提出一种新型的PUF结构是强PUF设计领域未来的发展方向，目前最为强大的PUF结构—IPUF是基于XOR PUF进行设计的，通过分析IPUF的结构以及对其成功建模的分治算法的攻击特点，不是对于IPUF的简单组合，而是通过更加新颖的思维与IPUF进行结合提出一种新的IPUF的变体来对于IPUF的脆弱点进行加固从而抵御机器学习

的攻击；（2）**IPUF结构的新型机器学习算法研究**。基于可靠性的机器学习算法CMA-ES对于有噪声的传统的APUF以及XOR PUF的建模十分有效，但对于IPUF的直接建模同样不成功，目前提出的分治算法是基于传统的机器学习算法的发展，基于目前提出的分治算法来对于利用可靠性信息进行建模的CMA-ES算法进行改进，从而发展CMA-ES攻击理论，可信的观点是同样基于分治算法的思想对于IPUF进行基于可靠性信息的建模攻击；（3）**PUF结构的通用安全性测试方法的研究**。目前针对于PUF的机器学习的理论方法有很多，对于一种新型的PUF结构的提出需要对其安全性进行测试的方法繁多而且比较复杂，在IPUF结构提出时就针对于各种的已经提出的机器学习算法进行了编码测试验证，为了PUF领域的快速发展提出一种通用的测试方法迫在眉睫。

目前，网络安全研究组已部署人力围绕上述问题开展相关研究，同时也会持续重视科研成果转化，解决关键领域中实际的安全问题。

邹德清
教　授
研究方向：系统与软件安全
Email：deqingzou@hust.edu.cn

羌卫中
教　授
研究方向：系统与软件安全
Email：wzqiang@hust.edu.cn

徐　鹏
教　授
研究方向：密码学
Email：xupeng@hust.edu.cn

马晓静
副教授
研究方向：多媒体安全、人工智能安全、隐私计算
Email：lindahust@hust.edu.cn

代炜琦
副教授
研究方向：区块链、云安全、隐私技术、可信计算
Email：wqdai@hust.edu.cn

文　明
副教授
研究方向：软件安全、程序分析与测试、代码大数据
Email：mwenaa@hust.edu.cn

胡胜山
副研究员
研究方向：人工智能安全、应用密码学、大数据隐私保护
Email：hushengshan@hust.edu.cn

杨已彪
副教授
研究方向：软件测试与分析
Email：yangyibiao@hust.edu.cn

王虹飞
副研究员
研究方向：芯片测试与诊断、芯片失效分析、人工智能芯片
Email：hongfei@hust.edu.cn

袁　斌
博士后
研究方向：软件定义网络安全、物联网安全
Email：yuanbin@hust.edu.cn

大数据组研究规划

石宣化，陈汉华，赵　峰，袁平鹏，华强胜，陆　枫，丁晓锋，黄　宏，张　腾，万　瑶

本组在2021年的研究重点是大数据基础理论（如面向动态大图图性质维护的的低复杂度算法和机器学习理论中的最优间隔分布学习机）、大数据处理支撑技术（如异构图计算、大规模深度学习、实时关联大数据高时效处理系统等）、大数据管理与分析技术（如医疗、社交和代码大数据分析）等。主要的研究规划和关注的研究内容如下：

（1）动态大图上图性质维护的低复杂度算法

如何在动态大图上快速进行图性质维护受到了越来越多的关注，但是目前已有的动态大图算法绝大部分仅仅依靠实验结果来支撑其图性质维护算法的高效。虽然也有部分研究聚焦于设计仅仅依赖于△G(图的变化量—如插入/删除/更改的节点/边/权值)和△O(输出变化—如哪些节点或边的性质因图改变而发生了变化)的具有最坏性能保证的动态图性质维护算法，但是一方面这些研究缺乏和最优算法的比较，另一方面已有研究也仅仅局限于个别图性质的维护。

为此我们拟对大图分析中常用的一系列图性质维护问题设计能够匹配或者近似匹配下界的低复杂度图性质维护算法；另一方面我们拟对不同图性质依据图性质维护难度进行精细划分，譬如哪些图性质能够在常数，或者在仅仅依赖于△G和△O的线性、平方抑或立方时间能能解决，而哪些不能够在相应时间内能解决。

（2）资源受限情况下的最优间隔分布学习机

随着数据规模的不断扩大，以及现实应用条件越发复杂苛刻，最优间隔分布学习机（ODM）效率不足的问题日渐暴露出来，这个问题直接制约了ODM进入更多的实际应用，尤其是计算存储资源受限如智能手机等终端设备上的应用，以及时间资源受限如对实时性要求高的量化交易、人机交互等应用。为此我们拟从学习模型生命周期的四个阶段出发，重点对以下问题进行研究：1）模型训练：解决非线性核函数带来的计算存储开销；2）模型更新：应对样本分布实时变化对性能带来的不利影响；3）模型评估：解决超参数组合过多模型选择开销过大的问题；4）模型复用：将在评价指标a下训练好的模型复用到与其相似的评价指标b下从而节省计算开销。

（3）图感知的缓存策略加速GNN训练过程

随着深度模型在图像、文本等传统领域的取得了不断突破，如何将深度学习模型引入到非欧式距离的图结构成为研究的热点，GNN通过消息传播机制将图的结构信息和属性进行联合高效的学习，在节点分类、链路预测等领域取得了超过传统模型精度效果，然而GNN训练过程中由于频繁的采样带来了大量的图结构数据和中间数据传输，对共享的参数的更新带来的大量的原子操作以及图结构的稀疏性带来不规则的访存，严重限制了异构系统（CPU-GPU）的GNN训练系统效率。

为此，我们提出了基于图结构感知的GNN训练通信和访存优化技术，重点展开以下方面研究。1）研究图结构感知的高效缓存技术，减少GNN训练过程中的pcie通信开销。2）提出采样感知的去重与压缩技术，减少数据传输量。3）研究GNN在异构加速器多层存储架构下的访存调度技术，减少训练过程中的原子操作开销。

（4）深度学习系统多任务内存管理研究

新型加速硬件的片上内存越来越大，例如A100 GPU上有80GB显存，现有的典型深度学习模型往往用不到如此大的内存。以PyTorch和Tensorflow等系统为代表的深度学习框架是针对单个任务进行资源管理的，多个任务同时运行需要将彼此之间的内存区域完全分隔开，但是由于智能模型计算过程中内存的使用是动态的，粗粒度且静态的内存分配会导致内存资源不能被充分利用，使得新型加速硬件上能支持的多任务大小及数量有限。

为此，拟提出深度学习多任务内存管理的研究方向。可依此开展以下研究：1）探究多任务环境下合适的计算调度粒度，太细（例如：operator）的粒度会导致调度开销过大，太粗（mini-batch）的粒度又很难使得总体内存达到最优；2）设计复杂度合理的计算调度算法，使得总体内存使用达到最优或次优；3）分析不同深度学习应用计算过程中的内存使用特征，设计多任务组合算法，使得内存利用率更高。

（5）基于硬件加速的内存数据库性能研究

随着数据库系统的内存逐渐增加，整张数据库表甚至整个数据库都能够存储在内存中。大规模的数据库系统将很多热点数据存储在内存中来提高读写性能，因此内存中的数据库操作也越来越多。而传统的数据库系统会分析大量的数据集，这些数据集通常是不规则的，而且没有索引，这导致在内存中数据库操作成本会变得很高昂，如何提高内存中数据库操作的性能成了一个很大的挑战。

硬件加速是指利用硬件模块来替代软件算法，从而发挥硬件固有的快速特性，目前常用FPGA器件进行硬件加速。FPGA 器件属于专用集成电路中的一种半定制电路，是可编程的逻辑列阵。为了提高内存中数据库操作的性能，一个可行的方案是，在采用外围电路方式下用FPGA芯片对数据库中的操作进行特殊的电路设计，从而达到程序并行、减小不必要开销的效果。

（6）实时关联大数据高时效处理系统

当前，5G网络和物联网设备的快速普及加速了数字经济的发展，促进了实时应用的丰富化进程。未来旨在围绕跨域关联高时效大数据流处理系统的基础性问题，拟开展实时关联大数据高时效处理系统关键核心技术的研究，包括：1）针对图流数据中动态边的时间信息存储难题，研究图流分层摘要图流存储结构，实现大数据图流时间范围查询。2）针对规模数据并行的高吞吐实时流处理技术，结合可实现超大规模并行计算单元的新型硬件（例如FPGA等），研究和设计和实现基于超大规模数据并行的高吞吐实时流处理系统。3）研究支持多样化语义的高时效大数据流处理平台。研究支持多样化语义的高时效流处理技术，构建多语义流处理系统模型，研发多语义流数据实时分析引擎。

（7）巨量图的计算与学习

图是表达事物之间复杂关联和知识的有效方式之一。关联的复杂性表达需求导致图的形式多样，如无标签的简单图、带标签的属性图以及时态图等。世界数字化需求同样引发图数据的规模飞速增长。因此，如何高效的对巨量多模态的图数据进行存储、处理和学习是亟待解决的科学问题。具体的探索如下：1）研究多模态图的模型；2）研究自适应而高效的多模态图划分方法；3）研究多模态图数据的存储模型和索引机制；4）研究适应新型体系结构下的并行分布式图处理任务的调度方案。5）研究图处理任务的分解方法以及适应大规模并行分布处理的大图执行计划生成方法；6）研究多模态图的挖掘，如时态图或多层图的稠密子图的挖掘，最短路径和可达性的查询等；7）研究知识图谱的智能问答技术，包括自然语言问题语义理解，面向智能问答的知识表示和问题生成等。

（8）针对多源异构大数据的高效数据管理

数据和内容是大数据网络的核心，针对大数据数据体量大、种类多、价值密度低等特点所导致的数据利用率低等问题，如何有效地对多源异构的大规模数据进行高效的存储管理，并准确获取我们所需的价值数据具有重要的研究意义，也是大数据网络信息管理的核心价值所在。有鉴于此，我们主要研究：（1）大数据环境下的知识管理，从知识生成、存储、组织、自治、导航、检索等层次出发，研究数据驱动的知识发现基本理论和关键技术。如：研究基于多源异构大数据的处理转化机制，构建海量知识库的高效抽取功能框架，并实现多知识库的统一映射转化技术；研究知识图谱的构建技术，研究针对元数据、知识实体及语义关联信息的自动抽取技术，将数据源中提取的实体、属性、关系加入并归纳到数据层，实现知识图谱自底向上的智能构建；研究知识图谱的弹性存储模型，支持大规模知识库的增量式更新，研究弹性存储模型以及访问策略等。（2）大数据的实时搜索，检索用户所需要的信息，理解数据空间和网络空间之间的交互，对其进行可靠管理、准确分析和全面理解，并提供个性化的服务，主要内容包括深层网络搜索、社交网络搜索、泛在网络信息搜索、数据流链路预测等。（3）大数据的智能推演技术，推理并挖掘出网络大数据背后隐含的关联信息，提供数据的智能化服务，主要内容包括构建基于嵌入式知识图谱的知识库表示模型，研究嵌入式知识图谱表示的关系推演算法，知识图网络匹配算法、知识图网络关联分析等技术。

（9）大规模异质时序网络表示学习方法研究

随着网络规模的不断发展和壮大，对网络的表示学习也成为网络研究的一大热点。良好的网络表示方法能有效提高网络后续任务的执行能力和执行效率。目前对网络的表示学习多基于静态网络展开，然而，真实的网络具有实时变化的特点，并且网络中的节点也并不是同一类型的。对网络的动态性和异质性建模必将提高网络的表示能力，因此，本研究将重点关注异质时序网络的表示学习。一是考虑如何更好地融合异质的网络信息；二是在大规模的的网络中，如何有效地模拟网络的动态性。本研究将充分考虑这两方面特点，重点突破大规模网络的表示学习。具体拟展开以下研究：1）时序网络的表示学习研究；2）大规模网络的表示学习；3）大规模网络中异质信息的处理。

（10）基于数据共享与增强的医疗大数据建模学习与解析

随着机器学习在医疗领域的应用，如何利用机器学习模型展开对医疗数据的分析，从而得到能够应用于临床的机器学习模型并发现相关医学知识成为研究热点。为此，我们提出了重点展开以下四个方面研究。1）研究基于联邦学习和区块链技术的多方医疗数据可信安全共享学习模式。2）面向微小特征提取的两段式深度学习模型构建。3）基于球面多维度影像数据增强的医疗图像拼接算法。4）基于特征主动式发现的医疗数据融合式深度学习模型构建。5）与此同时，我们将相关技术应用于新冠肺炎患者预后指标分析、新冠的中药治疗效果分析、儿童脓毒症的实验室预警指标探究、手术麻醉指标分析、术中低血压监测等领域。

（11）移动边缘计算下的大数据查询处理方法

随着各种智能终端所催生的5G时代到来，如何有效地利用、融合机器学习技术与数据管理技术，保障数据查询的效率和安全，从而实现更科学的大数据存储、分析和管理具有重大的研究价值和现实意义。基于以上需求，拟研究以下内容：（a）研究一种既可以保护隐私又能降低查询消耗的查询优化框架；（b）研究具有隐私保护的学习化安全索引。通过引入可信硬件SGX的隔离，密封和远程证明等特征，进

一步增加学习化索引的安全性；（c）拟提出轻量级的安全 R-树索引（SR*-tree）。从索引结构和查询协议方面提高查询效率，并在保证查询效率的同时尽可能减少索引的存储空间。

（12）代码大数据分析

随着GitHub、StackOverflow等平台上代码大数据的出现，通过人工智能、自然语言处理、数据挖掘等技术实现软件编程智能化成为可能。我们旨在通过自然语言处理和程序语言分析等技术架起自然语言与编程语言之间的桥梁，构建一系列智能化工具（例如代码摘要生成、代码片段搜索、代码补全、代码类型推断等等），从而辅佐程序员进行软件开发，提升软件开发的效率和质量。具体地，我们将聚焦代码表示学习这一核心问题，通过深度学习网络模型、图模型、类似BERT的预训练技术等实现程序的准确理解。基于此，我们将实现一套完整的开源软件NaturalCC，打造首个智能化编程的开源平台，从而方便研究者模型复现、公平比较和展示。

石宣化
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

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

赵　峰
教　授
研究方向：信息检索、数据挖掘与分布式计算等
Email：zhaof@hust.edu.cn

袁平鹏
教　授
研究方向：图数据库、图处理系统、分布式计算等
Email：ppyuan@hust.edu.cn

华强胜
研究员
研究方向：分布式计算理论与算法
Email：qshua@hust.edu.cn

陆　枫
副教授
研究方向：大数据(医疗)
Email：lufeng@hust.edu.cn

丁晓锋
副教授
研究方向：数据管理
Email：xfding@hust.edu.cn

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

张　腾
讲　师
研究方向：机器学习
Email：tengzhang@hust.edu.cn

万　瑶
讲　师
研究方向：代码大数据分析
Email：wanyao@hust.edu.cn

知识图谱与图神经网络的融合表示

（魏昊辉 https://blog.csdn.net/qq_42807120/article/details/115169493）

知识图谱（Knowledge Graph，KG）自从2012年被google提出，由于其出色的表达能力，以及在节点分类、连接预测等任务上的出色表现，已经逐渐成为AI相关应用的重要一支。与此同时图神经网络（Graph Neural Network，GNN），作为近些年深度学习方面发展最为迅速的领域，在信息通讯处理上展现出其得天独厚的优势。知识图谱作为一种特殊的结构化数据，其本质也是图相关结构，因此，如何将KG与GNN相结合，成为了近几年知识图谱领域里一大热点。

1. 知识图谱简介

KG的基础结构是传统三元组，即<h,r,t>，其中h代表头实体，r代表关系，t代表尾实体，头实体和尾实体作为两个节点，通过关系相连，因此，知识图谱以有向图的形式表示客观事物的关联性，经典的知识表示算法分为三大类：

（1）平移模型：将头尾实体看作空间内两个点，两点之间的矢量即为关系，头实体向量表示加上关系向量表示应等于尾实体向量表示以TransE为代表的平移模型都以此为基础。

（2）张量分解模型：张量分解模型将知识图谱看作一个大的三维张量，两维为实体表示，一维为关系表示，将每一个三元组对应的张量值都分解为实体表示和关系表示的矩阵形式，典型张量分解模型由Rescal。

（3）神经网络模型：利用CNN等神经网络模型对实体关系表示进行建模，如ConvE、ConvKB。

2. 图神经网络简介

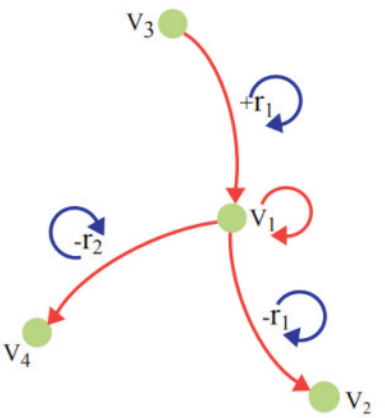
图由于其良好的信息交互能力，可以更好地对现实中的很多场景进行建模，如社会关系图、生物分子图等。现有的很多GNN都能对这些图模型进行拟合，如图卷积神经网络（Graph Convolutional Network，GCN）和图注意力神经网络（Graph Attention Network，GAT）。同时图又根据节点与边的种类分为同构图（Homogeneous）和异构图（Heterogeneous），异构图上的图神经网络也成为了最近人工智能领域的热点。

3. KG-GNN

传统的知识表达模型大多是独立学习知识库中每一个三元组，其实质更类似于NLP，这使得知识图谱中大量的邻居结构信息都被忽略，而图神经网络则充分利用了信息在节点之间的传播，因此将图神经网络的概念嵌入到知识图谱中成为有效对知识与数据进行关联的新想法。

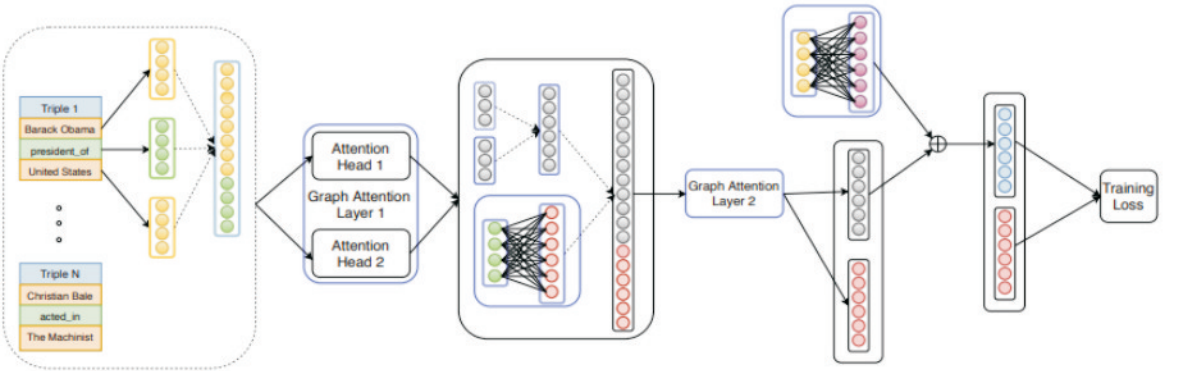
（1）TransGCN：这个模型是发表在K-Cap2019上的文章TransGCN:Coupling Transformation Assumptions with Graph Convolutional Networks for Link Prediction所提出的模型，这也是第一个将知识图谱与图神经网络相结合的模型，其基础思路很简单，类似于GCN，主要依靠信息的聚合与传递完成实体节点与关系的更新。

在TransGCN中，考虑到对于一个中心实体会存在两种方向的关系，如v3-r1->v1，v1-r2->v4因此在信息聚合时，将以中心实体为尾节点都逆向表示为v1-r1'->v3，以中心实体相邻关系及所连尾实体表示作为每个邻居的信息，以此进行聚合。



（2）Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs：发表于ACL2019）在GCN结合到KGE问题上并得到较好表现后，作者想到另一种高效的图神经网络，GAT。

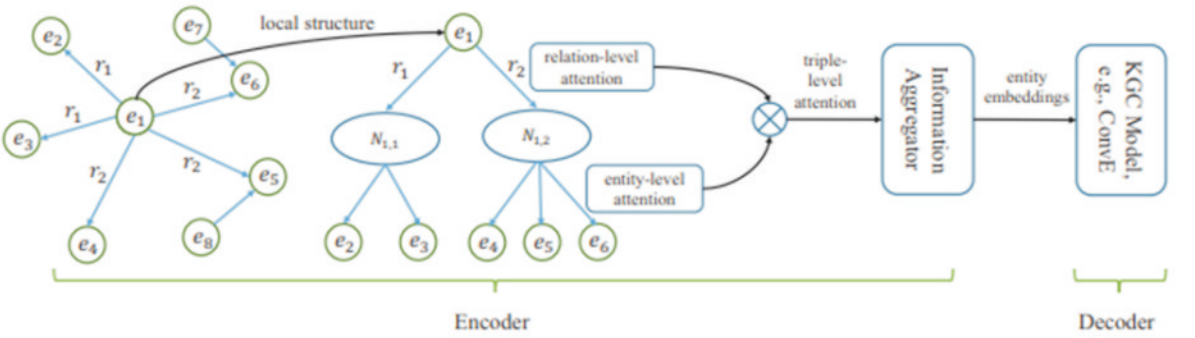
知识图谱以图结构进行考虑，最重要的环节是如何将relation的表示加入到信息的聚合中，该模型考虑直接将<h,r,t>看作h一个完整的邻居，并在此基础上将整个网络放入图注意力神经网络。



在文章最后，作者表明GAT加入知识图谱后，通过多次迭代可以学习到多阶邻居信息，高效获得全局信息。

（3）RGHAT：由Fuzhen Zhuang团队在AAAI-20上发表的Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion提出，思想也是将知识图谱与GAT相结合，不过该模型提出，将关系与实体作为整体进行信息的聚合会导致关系重要性不明确的问题，因此该模型通过两层结构，对关系与实体分别进行注意力机制的处理。

模型首先在头实体及邻居关系进行注意力机制，获得一阶注意力，然后再对h，r，t进行注意力机制，获得二阶注意力，最终，每个邻居的注意力即为一二阶注意力之积。



零知识证明在区块链中的应用

（庹舒月 https://blog.csdn.net/weixin_40252347/article/details/115173055?spm=1001.2014.3001.5502）

一、引言

在国家重视和政策扶持下，区块链技术的迅猛发展，区块链作为一个共享数据库，存储于其中的数据或信息拥有“不可篡改”“全程留痕”“可追溯”“公开透明”“去中心化”等特征。基于这些特征，区块链技术奠定了坚实的“信任”基础，创造了可靠的“合作”机制，拥有广阔的运用前景。零知识证明由于能在区块链应用中增强隐私性和安全性的潜力，也得到了广泛关注。

本文首先简要地介绍了区块链以及零知识证明的相关理论知识，接着对于当前区块链中零知识证明的应用进行阐述，包括zk-SNARK算法和ZeroCash区块链实现。最后，对于零知识证明在区块链中的应用进行总结和未来展望。

二、相关理论

1、区块链

从本质上讲，区块链（Blockchain）是一种按时间顺序将数据块连接起来的数据结构，并以密码学的方式保证其不可篡改。在数据方面看来，区块链是一种近乎不可被篡改的分布式数据，而且这里的分布式特点不仅仅体现在数据的分布式存储，而且表现为数据的分布式记录。在技术方面看来，区块链技术是多种技术共同整合的结果，并不是一种新型的单一技术。这些技术通过一种新的形式结构整合在一起，构成了一种新的数据结构用于数据的记录和存储。

区块链具有以下特性：（1）去中心化，区块链能够实现完全去中心化，不需要存在中心节点，采用分布式的数据存储结构，且所有的节点均保存着相同的区块信息。（2）不可篡改性，每个区块的块头部分均存储着Merkle树根节点的Hash值，其Merkle树的叶子节点则由每一条存储在区块链中的交易信息的相应Hash值构成，这样便可以通过验证Merkle树根节点的Hash值确保交易数据的准确性，实现不可篡改。（3）不可伪造性，在区块链不仅存储着交易数据的Hash值，而且其中包含交易双方的签名。而对于每一个使用者而言签名都是独立且唯一的，具有不可伪造性。（4）可验证性，区块链的可验证性一方面是指数据来源的可验证，可以验证电子货币在每一笔交易中的输入和输出。例如，在比特币中所有交易的输出均会作为下一笔交易的输入，而交易的输入均是前一笔交易的输出，即交易的可追溯性。除了保障数据来源的可验证外，还有确保交易金额的可验证，保障每一笔交易时的输入资金都是真实可靠的。（5）匿名性，区块链中的匿名性主要采用一种伪匿名的方式。区块链利用假名技术来将账号和真实身份的关系切断。例如，区块链中用户账号便是通过对用户公钥进行Hash运算得到的固定长度的Hash值。

2、零知识证明

零知识证明（Zero-Knowledge Proof）是由S.Goldwasser、S.Micali以及C.Rackoff在20世纪80年代初首次提出的。它指的是证明者能够在不向验证者提供任何有用的信息的情况下，使验证者相信某个论断是正确的。证明者向验证

者证明并使其相信自己知道或拥有某一消息，但证明过程不能向验证者泄漏任何关于被证明消息的信息。最终，他们能够证明通过证明者到核实者之间的一些交互，他们可以从根本上减少两者之间需要传递的知识数量。他们主要关注的是信息泄漏，也就是验证者在验证一个声明是否有效的过程中会了解到多少信息。

例如对于现在登录网站而言，在Web服务器上存储了客户的密码的哈希值，为了验证客户实际上知道密码，目前大部分网站采用的方式是服务器对客户输入的密码进行哈希计算，并与已存结果对比，但是这种方式的弊病在于服务器在计算时就可以知道客户的原始密码，一旦服务器被攻击，用户的密码也就泄露了。如果能够实现零知识证明，那么就可以在不知道客户密码的前提下，进行客户登录的验证，即使服务器被攻击，由于并未存储客户明文密码，用户的账户还是安全的。

零知识证明可以分为交互式和非交互式两种。对于交互式而言，零知识证明协议的基础是交互式的。它要求验证者不断对证明者所拥有的“知识”进行一系列提问。证明者通过回答一系列问题，让验证者相信证明者的确知道这些“知识”。然而，这种简单的方法并不能使人相信证明者和验证者都是真实的，两者可以提前串通，以便证明者可以在不知道答案的情况下依然通过验证。

三、区块链中零知识证明的应用

1、zk-SNARK

zk-SNARK是“zero knowledge Succinct Non-interactive Argument of Knowledge”的缩写，是非交互式零知识证明中的一种，一个将实际的零知识证明类问题转为计算机程序问题的理论。作为目前密码学及其应用领域中最令人激动的成果之一，它使得人们可以对形如“给定一个公开的谓词F，和一个公开输入x，我知道某个秘密输入w，使得$F(x,w)$为真”的语句进行证明和验证。其实现过程大致可以分为以下几步：

1）将要验证的程序拆解成一个个逻辑上的验证步骤，将这些逻辑上的步骤拆解成由加减乘除构成的算数电路。

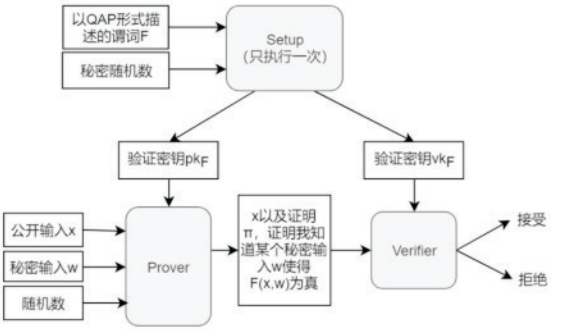
2）通过一系列的变换将需要验证的程序转换成验证多项式乘积是相等的，如证明t（x）h（x）=w（x）v（x）。

3）为了使得证明更加简洁，验证者预先随机选择几个检查点s，检查在这几个点上的等式是否成立。

4）通过同态编码/加密的方式使得验证者在计算等式时不知道实际的输入数值，但是仍能进行验证。

5）在等式左右两边可以同时乘上一个不为0的保密的数值k，那么在验证（t（s）h（s）k）等于（w（s）v（s）k）时，就无法知道具体的t（s）、h（s）、w（s）、v（s），因此可以使得信息得到保护。

zk-SNARKs协议由三个算法组成：Setup、Prover、Verify。zk-SNARKs的典型流程如图所示。



zk-SNARKs方案典型流程

2、ZeroCash

目前区块链中一些著名公链，例如BTC和

ETH的交易，在交易成功后，我们去区块链浏览器或调用对应的RPC接口查看对应的交易记录，可以获得交易发送者地址、接收者地址和交易金额等数据。这些地址仅仅只是一串字母和数字，从而起到一定的匿名效果，但是这仅仅只是一种伪匿名，通过一系列的相关交易和对应的显示社会关系还是可能会暴露用户的身份，从而ZeroCash区块链便应运而生了。

ZeroCash是基于零知识证明来进行隐私保护的区块链，采用了zk-SNARKs的非交互式的零知识证明来证明某些特定的资产归自己所有，目的是彻底解决交易被追踪从而暴露用户隐私的问题。ZCash拥有一个匿名交易系统，它支持多种交易类型，其中一种就是能够隐藏交易双方地址和交易数值的交易。这是完全的隐藏，而不是说数据还存放在ZCash的节点数据库中，而不向外显示出的隐藏。而是连节点都不知道交易的内容。Zcash的强大隐私保证源于Zcash中的屏蔽事务可以在区块链上完全加密，但仍然可以通过使用zk-SNARK证明在网络共识规则下验证为有效。

在ZCASH中，摒弃了之前的UTXO方式，而是使用了一种基于UTXO，被称为NOTE（支票）的新方式代替。NOTE代表了当前账户对资产的支配权，与UTXO不同，账户余额的存储方式不再是“未消费的交易输出”，而是“未被作废的支票（NOTE）”；一个NOTE是由所有者公钥PK、所拥有金额V、和唯一区分支票的序列号r组成，表示为NOTE=(PK, v, r)。

Zcash交易分为两类：透明地址交易和隐藏地址交易。两个透明地址之间的交易则与比特币交易没有区别，透明地址交易的输入、输出直接是公开可见的NOTE信息；对于隐藏地址交易，隐蔽地址之间的交易也会出现在公有区块链上，所以大家都知道有一笔隐蔽交易发生了，手续费也会支付给矿工，但交易的地址、资金的数额以及备注字段都被加密过，是不可见的。

四、总结

在数字经济全球化背景下，区块链的出现首次从技术上解决了基于信任的中心化模型带来的安全问题，并使得通过互联网进行价值转移成为可能，然而区块链的公开透明特性也给人们的隐私保护带来了巨大的挑战，因此如何实现隐私保护成为了阻碍区块链进一步发展的难题。将零知识证明应用于区块链已然为解决隐私安全问题做出了重大贡献。本文重点围绕零知识证明和区块链技术展开，阐述了零知识证明与区块链的相关理论知识，并且对零知识证明在区块链中的应用做了说明。未来区块链的隐私保护仍然任重而道远，如何实现快速高效、可信的零知识证明算法以及如何实现能够抵抗量子计算的零知识证明算法，都是需要进一步解决的问题。而且区块链+零知识证明的组合让用户重新掌握他们的个人数据成为了可能，可以在无需透露个人信息的情况下达成金融协议、创建在线账户或验证身份等。

参考文献

[1] Sasson E B , Chiesa A , Garman C , et al. Zerocash: Decentralized Anonymous Payments from Bitcoin[C]// 2014 IEEE Symposium on Security and Privacy (SP). IEEE, 2014.

[2] Bensasson E , Chiesa A , Tromer E , et al. Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture[J]. Sasson, 2014.

[3] 李康, 孙毅, 张珺,等. 零知识证明应用到区块链中的技术挑战[J]. 大数据, 2018, 004(001):P.57-65.

[4] Alshaikhli I F T , Makarin R H , Mohd Bakri S K , et al. Zero-knowledge-proof[J]. Cryptography Past Present & Future, 2011.

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Consulted, 1:2012, 2008.

关于不完整数据集的聚类

（朱玉莲 https://www.yuque.com/yulian-foak9/lac5tg/mmglgy）

在对数据进行数据挖掘和分析时，很多时候都是基于现有的数据，而已有的数据在采集和收集的过程中可能存在流程不规范、记录不准确的情况，导致数据出现错误和缺失的情况。即使是不完整的数据集，也是有进行数据分析的价值的，前段时间收集查阅了一些资料，本文对关于不完整数据集的聚类做一个总结。

1 数据缺失类型及定义

根据缺失的原因可以将数据缺失类型分为以下三类：

完全随机缺失：个数据元组在某个属性上是否有缺失值既不依赖于已知数据也不依赖于缺失值本身，数据的缺失完全是由随机因素导致的。

随机缺失：缺失值和缺失数据本身没有关系，却和已知的数据有一定的相关性，所以可以通过已知的数据得到数据缺失的方式。

非随机缺失：数据缺失的方式是非随机的，数据的缺失和缺失数据本身有一定的关系。

2 缺省数据的预处理

对于缺省数据的处理主要有三种方式：一是剔除，将缺省数据删除，最后得到一个完整的数据集；二是填补，使用零填充和均值填充和KNN插值等方法；三是不进行处理，采用支持不完整数据集聚类的算法。

3 不完整数据集的聚类算法

主要介绍三种支持不完整数据集的聚类算法。

3.1 基于模糊聚类的不完整数据聚类算法

将数据集分为完整的数据据元组集合和不完整的数据元组集合。

在模糊聚类中，将归属矩阵U和聚类中心点v作为迭代过程中的优化求解变量，同样，将缺失的属性值也作为迭代过程中需要求解的变量，最后将依旧能够得到聚类的结果，优化问题使用拉格朗日乘子法来求解。即求

$$\min_{(U,v)}\{J_m(U,v,X_m\}=\sum_{k=1}^{n}\min\{\sum_{i=1}^{c}(U_{ik})^m(d_{ik})^2\}$$

在 $\sum_{i=1}^{c}U_{ik}=1$ 约束下的解。d_{ik}表示样本记录和cluster之间的距离。

归属矩阵的求解公式

$$U_{ik}^{(r+1)}=D_{ik}^{\frac{1}{(1-m)}}\Big/\sum_{j=1}^{c}D_{ij}^{\frac{1}{(1-m)}}$$

聚类中心的求解公式

$$v_{ij}^{(r+1)}=(\sum_{k=1}^{n}(U_{ik}^{(r+1)})^m x_{ij})\Big/(\sum_{k=1}^{n}(U_{ik}^{(r+1)})^m)$$

缺失属性值的求解公式

$$x_{ij}^{(r+1)}=(\sum_{i=1}^{c}(U_{ik}^{(r+1)})v_{ij}^{(r+1)})\Big/(\sum_{i=1}^{c}(U_{ik}^{(r+1)})^m)$$

通过以上拉格朗日乘子法推导出来的公式，采用迭代求解的过程去计算出上述三个求解变量，最后根据归属矩阵和聚类中心可以将数据归属到不同的簇中去。

3.2 基于密度的不完整数据聚类算法

对于常见的基于密度的聚类算法，算法的

核心思想是用一个点的领域内其邻居点的个数来确定和衡量这个点的空间密度。但是由于不完整数据集的数据点是不能够确定位置的，所以点之间的距离无法计算，也无法得到空间密度。所以需要解决对于聚类中心的刻画，为了使簇内相似度大，同时簇间相似度小，聚类中心点的密度较大，周围的数据都与它更相似，而聚类中心点之间的相似度更大。

定义当X_i和y_i中一个存在数据缺失时，定义相似度度量（即距离）为，

$d_{ij}=\frac{s}{I_i}\sum_{k=1}^{s}(x_{ik}-x_{jk})^2 I_{ik}$，S为数据$x_i$的属性个数，

$$I_{ik}=\begin{cases}0, x_{ik}\text{缺失}\\ 1, x_{ik}\text{不缺失}\end{cases}，\quad I_i=\sum_{j=1}^{s}I_{ik}$$

定义点的密度为：$\rho_i=\sum_{j\in I_S, i\in I_S}e^{-(\frac{d_{ij}}{d_c})^2}$，$d_c$为分割距离。

定义点与其他疑似聚类中心的距离：

$$\delta_i=\begin{cases}\min_{q_j, j<i}\{d_{q_i q_j}\}, i\geq 2\\ \max_{j\geq 2}\{\delta_{q_j}\}, i=2\end{cases}$$

，当x_i具有最大的点密度时，δ_i表示集合S 中与点xi距离最大的点与ix 的距离；其他情况时，δ_i表示所有点密度比x_i大的点，和x_i距离最小的那个点与xi之间的距离。

根据ρ_i和δ_i，可以定义

$$\gamma_i=\frac{\rho_i}{\max\{\rho_1...\rho_N\}}\frac{\delta_i}{\max\{\delta_1...\delta_N\}}$$

来找到点密度较大，且离其他疑似聚类中心较远的点来作为聚类中心。

只要确定了聚类中心，就可以将不同的数据记录归属到簇中去。

3.3 基于信息理论的不完整数据聚类算法

基于信息理论的聚类算法，就是根据数据出现的频率情况，计算出该数据出现在不同簇中不确定度的，然后将该条数据划分到不确定最小的簇中去。

但是对于不完整数据集，由于缺省值的存在，对数据的频数统计会不准确，最后导致聚类的不准确。所以需要对不完整数据进行频数估计。一种简单的思路就是假定缺省的数据可能的数值分布情况与现有数据的分布情况是一致的，那么在统计一个属性的某个数值出现频数时，加上缺省数据中可能出现该值的频数估计。即

$$g(x_{ir})=f(x_{ir})+f(x_{ip+1})\times f(x_{ir})/\sum_{u=1}^{p}f(x_{iu})。$$

在知道不完整数据的频数估计后，再利用信息理论的公式去计算$I(C_k|x)=I(C_k)-I(C_k,x)$得到将记录x划分到第k个cluster中的不确定度。

4 总结

目前我对于不完整数据的聚类的解决思路有：（1）使不完整数据集变完整；（2）定义不完整数据集的相似性度量矩阵；（3）支持空值聚类的算法。

言 论（转载）

科研需要注重理论和实践的结合，既要看有质量的论文，也要关注开源社区。要做务实的研究，避免提出脱离现实、天马行空的假设。

（胡嘉海 https://m.weibo.cn/7232522311/4612164109207569）

取乎其上，得乎其中；取乎其中，得乎其下；取乎其下，则无所得矣。

（王洪磊 https://t.bilibili.com/497574714320044522）

不戚戚于“贫贱”，不汲汲于“富贵”。平心静气，踏实肯干，必能长风破浪。

（张 伟 https://mp.weixin.qq.com/s/PJZ0xTxYIanl9kXMZS6QUA）

居安思危，思则有备，有备无患

（杨文辉 https://mp.weixin.qq.com/s/yA9kjA-BTNkVnAUAzd0uvw）

Zookeeper的Leader选举

（李少锋 https://www.cnblogs.com/leesf456/p/6107600.html）

一、前言

Zookeeper是一个开源的分布式集群协调服务，具备高性能、高可用、且具有严格的顺序访问控制能力（主要是写操作的严格顺序性）的特点。对于集群启动而言，很重要的一部分就是Leader选举，接下来就开始深入学习Zookeeper的Leader选举。

二、Leader选举

2.1 Leader选举概述

Leader选举是保证分布式数据一致性的关键所在。当Zookeeper集群中的一台服务器出现以下两种情况之一时，需要进入Leader选举：（1）服务器初始化启动，（2）服务器运行期间无法和Leader保持连接。下面就两种情况进行分析讲解。

2.1.1 服务器启动时期的Leader选举

若进行Leader选举，则至少需要两台机器，这里选取3台机器组成的服务器集群为例。在集群初始化阶段，当有一台服务器Server1启动时，其单独无法进行和完成Leader选举，当第二台服务器Server2启动时，此时两台机器可以相互通信，每台机器都试图找到Leader，于是进入Leader选举过程。选举过程如下：

（1）每个Server发出一个投票。由于是初始情况，Server1和Server2都会将自己作为Leader服务器来进行投票，每次投票会包含所推举的服务器的myid和ZXID，使用(myid, ZXID)来表示，此时Server1的投票为(1, 0)，Server2的投票为(2, 0)，然后各自将这个投票发给集群中其他机器。

（2）接受来自各个服务器的投票。集群的每个服务器收到投票后，首先判断该投票的有效性，如检查是否是本轮投票、是否来自LOOKING状态的服务器。

（3）处理投票。针对每一个投票，服务器都需要将别人的投票和自己的投票进行PK，PK规则如下：

· 优先检查ZXID。ZXID比较大的服务器优先作为Leader。

· 如果ZXID相同，那么就比较myid。myid较大的服务器作为Leader服务器。

对于Server1而言，它的投票是(1, 0)，接收Server2的投票为(2, 0)，首先会比较两者的ZXID，均为0，再比较myid，此时Server2的myid最大，于是更新自己的投票为(2, 0)，然后重新投票，对于Server2而言，其无须更新自己的投票，只是再次向集群中所有机器发出上一次投票信息即可。

（4）统计投票。每次投票后，服务器都会统计投票信息，判断是否已经有过半机器接受到相同的投票信息，对于Server1、Server2而言，都统计出集群中已经有两台机器接受了(2, 0)的投票信息，此时便认为已经选出了Leader。

（5）改变服务器状态。一旦确定了Leader，每个服务器就会更新自己的状态，如果是Follower，那么就变更为FOLLOWING，如果是Leader，就变更为LEADING。

2.1.2 服务器运行时期的Leader选举

在Zookeeper运行期间，Leader与非Leader服务器各司其职，即便当有非Leader服务器宕机或新加入，此时也不会影响Leader，但是一

旦Leader服务器挂了，那么整个集群将暂停对外服务，进入新一轮Leader选举，其过程和启动时期的Leader选举过程基本一致。假设正在运行的有Server1、Server2、Server3三台服务器，当前Leader是Server2，若某一时刻Leader挂了，此时便开始Leader选举。选举过程如下：

（1）**变更状态。**Leader挂后，余下的非Observer服务器都会讲自己的服务器状态变更为LOOKING，然后开始进入Leader选举过程。

（2）**每个Server会发出一个投票。**在运行期间，每个服务器上的ZXID可能不同，此时假定Server1的ZXID为123，Server3的ZXID为122；在第一轮投票中，Server1和Server3都会投自己，产生投票(1, 123)，(3, 122)，然后各自将投票发送给集群中所有机器。

（3）**接收来自各个服务器的投票。**与启动时过程相同。

（4）**处理投票。**与启动时过程相同，此时，Server1将会成为Leader。

（5）**统计投票。**与启动时过程相同。

（6）**改变服务器的状态。**与启动时过程相同。

2.2 Leader选举算法分析

在3.4.0后的Zookeeper的版本只保留了TCP版本的FastLeaderElection选举算法。当一台机器进入Leader选举时，当前集群可能会处于以下两种状态

· 集群中已经存在Leader。

· 集群中不存在Leader。

对于集群中已经存在Leader而言，此种情况一般都是某台机器启动得较晚，在其启动之前，集群已经在正常工作，对这种情况，该机器试图去选举Leader时，会被告知当前服务器的Leader信息，对于该机器而言，仅仅需要和Leader机器建立起连接，并进行状态同步即可。而在集群中不存在Leader情况下则会相对复杂，其步骤如下

（1）**第一次投票。**无论哪种导致进行Leader选举，集群的所有机器都处于试图选举出一个Leader的状态，即LOOKING状态，LOOKING机器会向所有其他机器发送消息，该消息称为投票。投票中包含了SID（服务器的唯一标识）和ZXID（事务ID），（SID, ZXID）形式来标识一次投票信息。假定Zookeeper由5台机器组成，SID分别为1、2、3、4、5，ZXID分别为9、9、9、8、8，并且此时SID为2的机器是Leader机器，某一时刻，1、2所在机器出现故障，因此集群开始进行Leader选举。在第一次投票时，每台机器都会将自己作为投票对象，于是SID为3、4、5的机器投票情况分别为(3, 9)，(4, 8)，(5, 8)。

（2）**变更投票。**每台机器发出投票后，也会收到其他机器的投票，每台机器会根据一定规则来处理收到的其他机器的投票，并以此来决定是否需要变更自己的投票，这个规则也是整个Leader选举算法的核心所在，其中术语描述如下：

· **vote_sid：**接收到的投票中所推举Leader服务器的SID。

· **vote_zxid：**接收到的投票中所推举Leader服务器的ZXID。

· **self_sid：**当前服务器自己的SID。

· **self_zxid：**当前服务器自己的ZXID。

每次对收到的投票的处理，都是对(vote_sid, vote_zxid)和(self_sid, self_zxid)对比的过程。

规则一：如果vote_zxid大于self_zxid，就认可当前收到的投票，并再次将该投票发送出去。

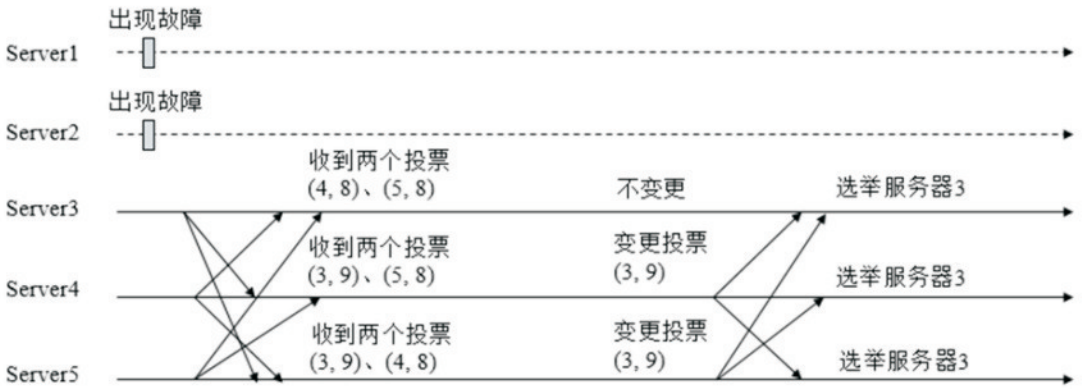
规则二：如果vote_zxid小于self_zxid，那么坚持自己的投票，不做任何变更。

规则三：如果vote_zxid等于self_zxid，那么就对比两者的SID，如果vote_sid大于self_sid，那么就认可当前收到的投票，并再次

将该投票发送出去。

规则四：如果vote_zxid等于self_zxid，并且vote_sid小于self_sid，那么坚持自己的投票，不做任何变更。

结合上面规则，给出下面的集群变更过程。



（3）确定Leader。经过第二轮投票后，集群中的每台机器都会再次接收到其他机器的投票，然后开始统计投票，如果一台机器收到了超过半数的相同投票，那么这个投票对应的SID机器即为Leader。此时Server3将成为Leader。

由上面规则可知，通常那台服务器上的数据越新（ZXID会越大），其成为Leader的可能性越大，也就越能够保证数据的恢复。如果ZXID相同，则SID越大机会越大。

2.3 Leader选举实现细节

2.3.1 服务器状态

服务器具有四种状态，分别是LOOKING、FOLLOWING、LEADING、OBSERVING。

LOOKING：寻找Leader状态。当服务器处于该状态时，它会认为当前集群中没有Leader，因此需要进入Leader选举状态。

FOLLOWING：跟随者状态。表明当前服务器角色是Follower。

LEADING：领导者状态。表明当前服务器角色是Leader。

OBSERVING：观察者状态。表明当前服务器角色是Observer。

2.3.2 投票数据结构

每个投票中包含了两个最基本的信息，所推举服务器的SID和ZXID，投票（Vote）在Zookeeper中包含字段如下

id：被推举的Leader的SID。

zxid：被推举的Leader事务ID。

electionEpoch：逻辑时钟，用来判断多个投票是否在同一轮选举周期中，该值在服务端是一个自增序列，每次进入新一轮的投票后，都会对该值进行加1操作。

peerEpoch：被推举的Leader的epoch。

state：当前服务器的状态。

2.3.3 QuorumCnxManager：网络I/O

每台服务器在启动的过程中，会启动一个QuorumPeerManager，负责各台服务器之间的底层Leader选举过程中的网络通信。

（1）消息队列。QuorumCnxManager内部维护了一系列的队列，用来保存接收到的、待发送的消息以及消息的发送器，除接收队列以外，其他队列都按照SID分组形成队列集合，如一个集群中除了自身还有3台机器，那么就会为这3台机器分别创建一个发送队列，互不干扰。

· **recvQueue**：消息接收队列，用于存放那

些从其他服务器接收到的消息。

· **queueSendMap**：消息发送队列，用于保存那些待发送的消息，按照SID进行分组。

· **senderWorkerMap**：发送器集合，每个SenderWorker消息发送器，都对应一台远程Zookeeper服务器，负责消息的发送，也按照SID进行分组。

· **lastMessageSent**：**最近**发送过的消息，为每个SID保留最近发送过的一个消息。

（2）建立连接。为了能够相互投票，Zookeeper集群中的所有机器都需要两两建立起网络连接。QuorumCnxManager在启动时会创建一个ServerSocket来监听Leader选举的通信端口(默认为3888)。开启监听后，Zookeeper能够不断地接收到来自其他服务器的创建连接请求，在接收到其他服务器的TCP连接请求时，会进行处理。为了避免两台机器之间重复地创建TCP连接，Zookeeper只允许SID大的服务器主动和其他机器建立连接，否则断开连接。在接收到创建连接请求后，服务器通过对比自己和远程服务器的SID值来判断是否接收连接请求，如果当前服务器发现自己的SID更大，那么会断开当前连接，然后自己主动和远程服务器建立连接。一旦连接建立，就会根据远程服务器的SID来创建相应的消息发送器SendWorker和消息接收器RecvWorker，并启动。

（3）消息接收与发送。消息接收：由消息接收器RecvWorker负责，由于Zookeeper为每个远程服务器都分配一个单独的RecvWorker，因此，每个RecvWorker只需要不断地从这个TCP连接中读取消息，并将其保存到recvQueue队列中。消息发送：由于Zookeeper为每个远程服务器都分配一个单独的SendWorker，因此，每个SendWorker只需要不断地从对应的消息发送队列中获取出一个消息发送即可，同时将这个消息放入lastMessageSent中。在SendWorker中，一旦Zookeeper发现针对当前服务器的消息发送队列为空，那么此时需要从lastMessageSent中取出一个最近发送过的消息来进行再次发送，这是为了解决接收方在消息接收前或者接收到消息后服务器挂了，导致消息尚未被正确处理。同时，Zookeeper能够保证接收方在处理消息时，会对重复消息进行正确的处理。

2.3.4 FastLeaderElection：选举算法核心

· **外部投票**：特指其他服务器发来的投票。

· **内部投票**：服务器自身当前的投票。

· **选举轮次**：Zookeeper服务器Leader选举的轮次，即logicalclock。

· **Pk**：对内部投票和外部投票进行对比来确定是否需要变更内部投票。

（1）选票管理

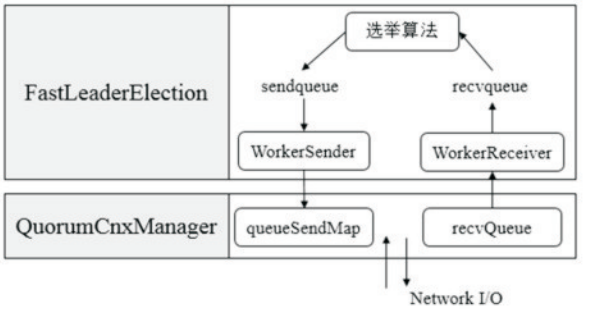
· **sendqueue**：选票发送队列，用于保存待发送的选票。

· **recvqueue**：选票接收队列，用于保存接收到的外部投票。

· **WorkerReceiver**：选票接收器。其会不断地从QuorumCnxManager中获取其他服务器发来的选举消息，并将其转换成一个选票，然后保存到recvqueue中，在选票接收过程中，如果发现该外部选票的选举轮次小于当前服务器的，那么忽略该外部投票，同时立即发送自己的内部投票。

· **WorkerSender**：选票发送器，不断地从sendqueue中获取待发送的选票，并将其传递到底层QuorumCnxManager中。

（2）算法核心



上图展示了FastLeaderElection模块是如何与底层网络I/O进行交互的。Leader选举的基本流程如下：

1. 自增选举轮次。Zookeeper规定所有有效的投票都必须在同一轮次中，在开始新一轮投票时，会首先对logicalclock进行自增操作。

2. 初始化选票。在开始进行新一轮投票之前，每个服务器都会初始化自身的选票，并且在初始化阶段，每台服务器都会将自己推举为Leader。

3. 发送初始化选票。完成选票的初始化后，服务器就会发起第一次投票。Zookeeper会将刚刚初始化好的选票放入sendqueue中，由发送器WorkerSender负责发送出去。

4. 接收外部投票。每台服务器会不断地从recvqueue队列中获取外部选票。如果服务器发现无法获取到任何外部投票，那么就会立即确认自己是否和集群中其他服务器保持着有效的连接，如果没有连接，则马上建立连接，如果已经建立了连接，则再次发送自己当前的内部投票。

5. 判断选举轮次。在发送完初始化选票之后，接着开始处理外部投票。在处理外部投票时，会根据选举轮次来进行不同的处理。

· 外部投票的选举轮次大于内部投票。若服务器自身的选举轮次落后于该外部投票对应服务器的选举轮次，那么就会立即更新自己的选举轮次(logicalclock)，并且清空所有已经收到的投票，然后使用初始化的投票来进行PK以确定是否变更内部投票。最终再将内部投票发送出去。

· 外部投票的选举轮次小于内部投票。若服务器接收的外选票的选举轮次落后于自身的选举轮次，那么Zookeeper就会直接忽略该外部投票，不做任何处理，并返回步骤4。

· 外部投票的选举轮次等于内部投票。此时可以开始进行选票PK。

6. 选票PK。在进行选票PK时，符合任意一个条件就需要变更投票。

若外部投票中推举的Leader服务器的选举轮次大于内部投票，那么需要变更投票。

· 若选举轮次一致，那么就对比两者的ZXID，若外部投票的ZXID大，那么需要变更投票。

· 若两者的ZXID一致，那么就对比两者的SID，若外部投票的SID大，那么就需要变更投票。

7. 变更投票。经过PK后，若确定了外部投票优于内部投票，那么就变更投票，即使用外部投票的选票信息来覆盖内部投票，变更完成后，再次将这个变更后的内部投票发送出去。

8. 选票归档。无论是否变更了投票，都会将刚刚收到的那份外部投票放入选票集合recvset中进行归档。recvset用于记录当前服务器在本轮次的Leader选举中收到的所有外部投票（按照服务队的SID区别，如{(1, vote1), (2, vote2)...}）。

9. 统计投票。完成选票归档后，就可以开始统计投票，统计投票是为了统计集群中是否已经有过半的服务器认可了当前的内部投票，如果确定已经有过半服务器认可了该投票，则终止投票。否则返回步骤4。

10. 更新服务器状态。若已经确定可以终止投票，那么就开始更新服务器状态，服务器首选判断当前被过半服务器认可的投票所对应的Leader服务器是否是自己，若是自己，则将自己的服务器状态更新为LEADING，若不是，则根据具体情况来确定自己是FOLLOWING或是OBSERVING。

以上10个步骤就是FastLeaderElection的核心，其中步骤4-9会经过几轮循环，直到有Leader选举产生。

“华科–华为数据中心架构联合创新实验室课题交流研讨会”顺利召开

毛伏兵

2021年3月3日，实验室在梧桐雨问道厅举行华科–华为数据中心架构联合创新实验室课题交流研讨会，本次研讨会主要参会人员有华中科技大学计算机学院吴涛书记，实验室主任金海教授，科学技术发展研究院院长廖小飞教授，计算机学院副院长石宣化教授，实验室蒋文斌教授，邵志远教授，华强胜研究员，刘海坤副教授，郑龙副教授，张宇副教授，顾琳副教授，肖江副教授。华为技术有限公司计算产品线研发管理部部长熊彦，武汉研究所所长张继超，中央研究院数据中心技术实验室主任邹斌，计算产品线架构设计部部长秦佩峰，计算产品线技术规划部部长惠涛，计算产品线技术开发部部长刘林超，计算产品线基础软件首席专家，计算产品线芯片技术首席规划师周轶刚，中央研究院数据中心技术实验室架构师廖健行。

会议由石宣化教授主持，实验室主任金海教授代表实验室致欢迎词，他向与会的领导、专家、学者表示欢迎，并指出要做有意思、有意义的工作，要做到工业界领先，学术届领先，要推动国内数据架构发展，共同为国家工业发展建设贡献力量。熊彦部长代表华为公司致开幕词，他衷心地表示华为很高兴能与优秀团队合作，希望通过大家的共同努力来提高产品质量，并创造更多新的可能性。

实验室四位老师代表和公司代表分别向在座的专家、领导作了汇报。首先，专家孙宏伟和邹斌分别介绍了华为产品线和中央研究院的工作计划、研究进展以及遇到的困难和可能的解决思路。报告指出在未来图规模很大，实时性需求高，稀疏度高，庞大的交通调度问题结合人工智能技术，处理起来会非常复杂。实验室代表刘海坤、郑龙、张宇、邵志远老师，针对各自的研究课题汇报了研究背景、研究目标和研究计划。在自由研讨环节，金海教授和廖小飞教授分别进行了点评。

吴涛书记做闭幕致辞，他指出华中科技大学和华为公司有着深厚的感情和深入的交流，二者的合作是学校和企业之间的强强联手。针对此次研讨会吴书记做了三点总结。1）非常聚焦：公司的报告、想法、需求和问题等思路特别清晰，在团队努力下后续工作的方向感会比较强；2）非常务实：聚焦问题进行讨论，体现了科学家的专研精神和华为勇于挑战的精神，这些会激励我们团队去解决问题；3）彼此能长远深入合作的愿景：针对现有问题要务实，研究前沿问题要能落地，彼此合作肩负使命。

最后，吴涛书记感谢金海老师团队，希望在与华为公司的合作过程中，能做领先、扎实和务实的工作；感谢华为公司的信任与支持，希望通过与华为各位专家的持续交流合作，大家在今后的工作中奋发努力，取得更好的成绩。

毛伏兵
讲　师
研究方向：系统软件和体系结构、集成电路物理设计、机器学习
Email：fbmao@hust.edu.cn

实验室学术委员会第十二次会议顺利召开

陶　莉

2021年3月21日，服务计算技术与系统教育部重点实验室第十二次学术委员会会议召开。实验室学术委员会成员中国人民解放军军事科学院梅宏院士、清华大学郑纬民院士、中国工程院陈左宁院士、中国科学院计算所孙凝晖院士、国防科技大学王怀民院士、上海交通大学过敏意教授、中科院信工所孟丹研究员、北京航空航天大学吕卫锋教授、西安电子科技大学杨宗凯教授、西安交通大学郑庆华教授等参会指导工作。实验室主任金海教授，副主任吴松教授、廖小飞教授以及实验室老师代表二十余人参加了会议。

本次会议由学术委员会委员中国人民解放军军事科学院梅宏院士主持，金海教授作了实验室2020年度工作汇报，就实验室2020年度整体情况、代表性科研成果、队伍建设与人才培养、开放交流与运行管理等情况做了详细汇报。郑龙副教授作了题为“图计算：现状、挑战与实践”的学术报告。

会上，学术委员会的专家们对实验室的工作情况和学术报告内容进行了充分的交流和讨论，充分肯定了实验室这一年以来的工作，认为实验室队伍建设和人才培养效果明显，各项工作取得了可喜的进展。同时，专家们对实验室学术发展、科研模式、成果转化等提出了宝贵的意见和建议，对实验室提出了更高的要求，希望实验室能进一步凝练学术主线、突出学术特色，争取再进一步迈上新台阶。

陶　莉
负责事务：微信公众号、论文英文查重等
Email：2018612060@hust.edu.cn

DELTA: A Security Assessment Framework for Software-Defined Networks

任昆仑　推荐

“DELTA: A Security Assessment Framework for Software-Defined Networks”是安全四大顶会之一NDSS在2017年收录的一篇论文。该文提出了自动化和标准化的SDN漏洞识别过程，开发了一个安全评估框架DELTA，在不同的测试环境中重新验证已发布的SDN攻击，并且，使用支持协议的fuzzing模块来增强DELTA，以自动发现新的漏洞。最终DELTA在不同的SDN控制器环境中成功重现了20个已知的攻击场景，发现了7个新的SDN应用误导攻击。

SDN(Soft-Defined Network)的提出是为了解决传统网络中控制平面和数据平面的紧耦合从而导致的难以部署网络新功能的问题。SDN强调控制平面和数据平面的解耦，逻辑上的控制平面部署在高性能的商业机器上，从而使得网络管理员能够在一个集中的网络视图上管理复杂的流量。SDN的关键特性是高层次的网络抽象和可编程性。SDN不再像传统网络那样使用静态网络策略，它抽象了网络底层的复杂性，对最终用户隐藏了网络拓扑和网络设备，对整个网络有着全局的视图。

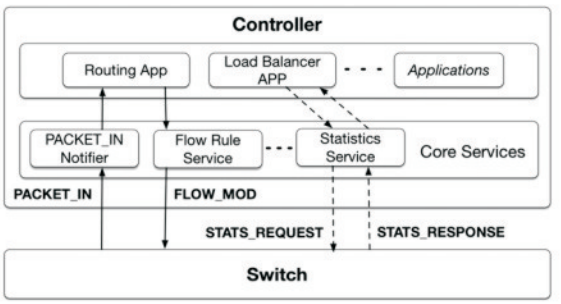


图1 SDN/OpenFlow控制流的示例

文章指出了目前针对比较著名的20种攻击，并根据其针对的不同SDN组件进行了分类和详细介绍，这里不再赘述。不过需要指出的是，有些攻击高度依赖于特定控制器的体系结构，该文在这里考虑了三种著名的控制器：ONOS、OpenDaylight和Floodlight，用它们代表当今最流行和广泛使用的SDN控制器。

对于已知的攻击场景，虽然很容易重现，但如果测试每个用例需要不同的测试环境，那么测试成本会很昂贵。所以测试框架需要有灵活配置的能力。DELTA的目的在于：1）覆盖尽可能多的攻击场景；2）高度自动化，尽量减少进行测试所需的人力和时间；3）很各种SDN组件可互操作。此外，DLETA还旨在能够容易地扩展到新的测试用例和帮助识别新的攻击场景。为了识别新的攻击场景，DELTA使用了模糊测试的概念，采用了黑盒模糊测试来生成随机化的测试向量。

DELTA的具体框架是由一个集中式代理管理器和多个代理组成。代理根据其位置分为三种不同类型：应用程序、通道和主机。这些代理位于SDN控制流的中间并实现攻击场景。

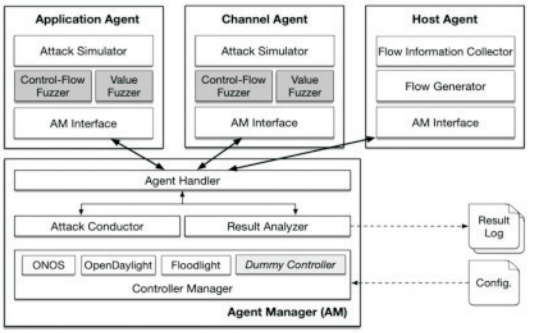


图2 DELTA的总体架构

下面是对各部分的简要介绍：

（1）代理管理器（Agent Manager）：代理管理器（AM）承担所有代理的控制器角色，它由四个模块组成：控制器管理器、攻击指挥器、代理处理器和结果分析器。AM和SDN的组件不耦合，它独立地执行两个功能：1）远程控制其他代理，重现已知的攻击场景或者发现针对目标网络的位置攻击场景；2）从每个代理检索执行结果。

首先控制器管理器读取一个包含目标控制器信息的配置文件，为了测试可能会使用一个虚拟的控制器。然后攻击指挥器启动已知的攻击场景，这个场景在之前预定义。复现攻击场景时，它根据预定义信息通过代理处理器来控制每个代理。最后，如果每个代理完成了一次攻击，结果将被发送到结果分析器，它将处理结果报告给操作员。攻击指挥器还控制每个代理中的模糊测试模块以发现未知的攻击案例。

（2）应用代理（Application Agent）：应用代理是一个运行在控制器内部的SDN应用程序，它在AM的监视下启动攻击。SDN应用程序可以直接参与SDN控制流，所以DELTA用它来拦截、伪造和更改SDN控制流和输入变量以适用攻击场景。

（3）信道代理（Channel Agent）：通道代理嗅探并修改在控制平面（这里就是控制器）和数据平面之间通过控制信道传递的控制消息。由于通信通常是未加密的，因此信道代理可以通过拦截消息来操作控制消息。

（4）主机代理（Host Agent）：主机代理作为一个主机或多个主机参与目标SDN网络。能过产生网络流量到任何可到达的目标（如交换机或主机），这样一个远程可控的主机发起一些有主机发起的攻击是有作用的。但与其他代理不同，宿主代理没有已知的攻击场景，但是可以使用它来生成小的或打的数据流，这些流可以在攻击场景中使用。

（5）模糊测试模块（Fuzzing Modules）：管理员可以选择部署DELTA中的模糊测试功能来使AM为应用和信道代理激活模糊测试功能。如果没有对模糊测试功能的指引，它们将一直运行，直到手动终止。操作者可以选择性地地为特定情况的一个随机化边界提供窄模糊测试提供输入。DELTA提供了两个模糊测试模块图的随机函数：1）Control-Flow Fuzzer；2）Value Fuzzer，它们都位于每个代理中。Control-Flow Fuzzer将SDN控制流操作随机化，Value Fuzzer将每个函数的输入值随机化。这些模块可以串联运行，也可以独立运行。

当一个随机化过程完成时，测试结果将被发送到AM中的结果分析器，然后分析结果来验证攻击场景的有效性。

该文通过几个流行的控制器对其提出的DELTA框架进行了评估，重现了20个已知攻击并发现了7个新的攻击场景，具体细节读者可在论文中查看。

这篇文章提出了一个系统的方法，来自动探索在SDN组件之间的关键数据流交换，以搜索已知和潜在未知的漏洞。该文提出的DELTA框架是第一个以SDN为重点的安全评估工具。此外，该文还提出了一种可推广的黑盒模糊测试算法，并将其集中到DELTA中，这种模糊测试算法是操作人员能够对组件接口的数据输入逻辑进行深入测试。

任昆仑
2020级博士研究生
研究方向：网络安全
Email：ren_kunlun@qq.com

Alleviating Irregularity in Graph Analytics Acceleration: a Hardware/ Software Co-Design Approach

陈绍鹏　推荐

“Alleviating Irregularity in Graph Analytics Acceleration: a Hardware/Software Co-Design Approach”是计算机体系结构领域国际顶级会议MICRO 2019录用的一篇文章，该会议于2019年10月在美国哥伦布市（Columbus）举行。文章针对图计算在并行过程中存在的负载、访存、更新不规则等一系列问题，结合传统架构（CPU/GPU）与现有定制加速架构的不足，基于软硬件协同思想，提出一种通过感知数据依赖并实现动态调度来缓和不规则影响的图计算加速架构GraphDynS。

图结构本身是具有高度可并行性的。然而，由于大规模图的迭代计算过程存在着大量且复杂的数据依赖行为，现有的图计算加速架构仍面临着三种不规则性挑战：（1）负载不规则，节点度数的不规则导致并行计算时分配到每个处理单元的负载不均衡；（2）遍历不规则，图遍历过程中的数据依赖导致大量的随机访存和数据冲突；（3）更新不规则，点更新和活跃点集更新过程中的全局遍历带来了大量的冗余计算和访存。针对这三种不规则性，文章对传统的以点为中心的编程模型（VCPM, Vertex-Centric Programming Model）进行数据通路去耦（datapath decoupling）优化，将任务分配与处理分离，并提出了数据感知的动态调度策略来缓解上述三种不规则性带来的影响。

如图1(a)所示，传统的Vertex-centric编程模型分为Scatter和Apply两个阶段：Scatter阶段遍历活

Algorithm 1: ***Vertex-Centric Programming Model***

```
                    ◁ Scatter Phase
for each active vertex u do
    offset ← OffsetArray[u.vid];
    while EdgeArray[offset].src_vid == u.vid do
        e(u,v) ← EdgeArray[offset++];
        edgeProResult ← Process_Edge(u.prop, e.weight);
        v.tProp ← Reduce(v.tProp, edgeProResult);
    end
end
                    ◁ Apply Phase
for each vertex v do
    applyRes ← Apply(v.prop, v.tProp, v.cProp);
    if v.prop!=applyRes then
        v.prop ← applyRes;
        activate vertex with v.vid and v.prop;
    end
end
```

（a）

Algorithm 2: ***Optimized Programming Model***

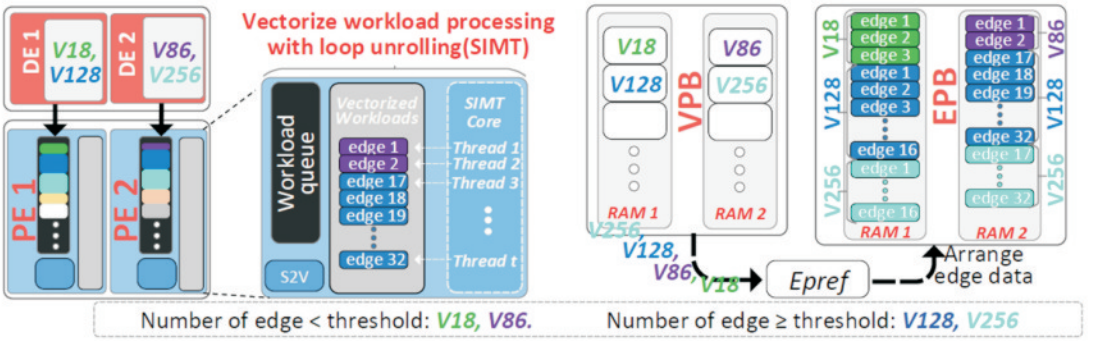
```
                    ◁ Dispatching Stage of Scatter Phase
for each active vertex u do
    dispatch(u.prop, u.offset, u.edgeCnt) to PE;
end
                    ◁ Processing Stage of Scatter Phase
for e(u, v) ← EdgeArray[u.offset : u.offset + u.edgeCnt]
  do
    edgeProResult ← Process_Edge(u.prop, e.weight);
    v.tProp ← Reduce(v.tProp, edgeProResult);
end
                    ◁ Dispathching Stage of Apply Phase
for each vertex list do
    dispatch(start id vListStartID and size vListSize of vertex
     list) to PE;
end
                    ◁ Processing Stage of Apply Phase
for vid ← vListStartID : vListStartID + vListSize do
    edgeCnt ← OffsetArray[vid+1] - OffsetArray[vid];
    applyRes ← Apply(v_vid.prop, v_vid.tProp, v_vid.cProp);
    if v_vid.prop != applyRes then
        v_vid.prop ← applyRes;
        activate v_vid with v_vid.prop, OffsetArray[vid],
         edgeCnt;
    end
end
```

（b）

图 1. 传统Vertex-centric编程模型(a)与Dispatching/Processing优化编程模型(b)

跃节点为源节点对应的所有边，并归约更新目的节点；Apply阶段比较更新前后的节点属性并更新活跃点集。由于传统的编程模型并不能有效地获取负载规模和精确地预取边数据，文章提出了Dispatching/Processing优化编程模型。如图1(b)所示，文章将编程模型的每个阶段分为Dispatching和Processing两部分：Dispatching 部分解耦数据通路，分配任务到不同处理单元；Processing部分获取负载规模并对边数据精确预取。

如图2(a)所示，文章提出的数据感知的动态调度策略主要分为三部分，对应解决上述三类不规则性问题。（1）负载调度。通过Dispatcher均衡调度负载到每个PE(Processing Element)，然后通过循环体展开技术将PE进行向量化负载处理；（2）遍历访问数据调度。对顶点和邻边精确预取，然后通过三级流水实现紧凑的原子更新流水线，以减少reduce的原子操作停顿；（3）数据更新调度。为避免对不必要节点的更新，在reduce过程中采用bitmap标记需要更新的候选节点，以保证只有候选节点会被更新处理，并采用activate-store机制，合并访问待激活节点数据。



（a） （b） （c）

图 2. 数据感知的动态调度

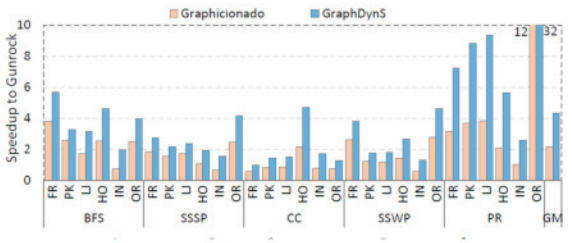


图3. 加速比对比图（相较于Gunrock）

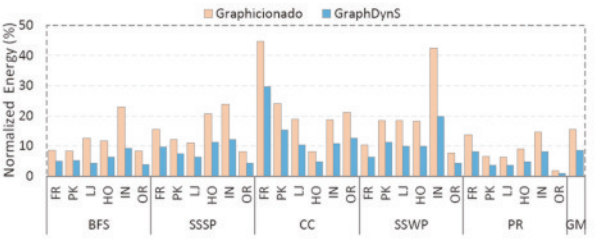


图4. 能耗对比图（相较于Gunrock）

文章实验采用不同规模数据集与算法（SSSP、CC、BFS、SSWP、PageRank等）将GraphDynS与当前主流的高度定制化的图加速器Graphicionado和基于GPU的大规模图计算系统Gunrock的性能功耗等参数指标进行了比较。如图3所示，GraphDynS相对于Graphicionado取得了1.9倍的性能提升，相对于运行于GPU V100的Gunrock取得了4.2 倍的性能提升。如图4所示，GraphDynS相较于Gunrock平均减少了91.4%的能耗，相较于Graphicionado能耗则平均减少了45%。

陈绍鹏
2015级博士研究生
研究方向：图计算，高层次综合
Email：shpchen@hust.edu.cn

Deep Leakage from Gradients

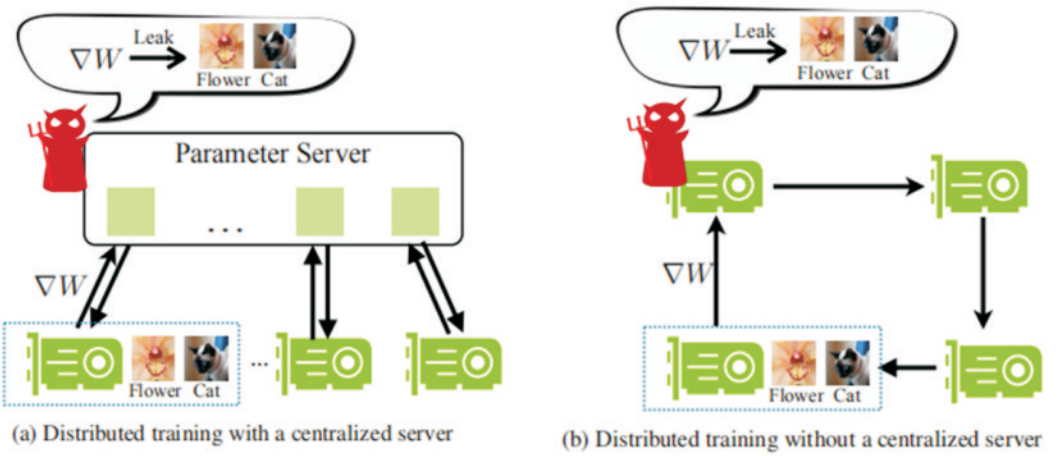
付佳韵　推荐

“Deep Leakage from Gradients”是机器学习领域顶级会议Annual Conference on Neural Information Processing Systems 2019收录的一篇论文，该论文介绍了一种简单有效的从公开共享梯度中恢复隐私训练数据的方法，命名为Deep Leakage from Gradients，揭示了梯度信息中蕴含的重大隐私泄露风险和基于梯度交换的分布式机器学习框架（collaborative learning）中的隐私安全隐患，并探讨了针对此种攻击可能的防御方法。

联邦学习框架（collaborative learning or federated learning）是一种分布式机器学习框架，被广泛应用于大规模机器学习任务。联邦学习训练过程中私有训练数据不会离开本地，多个客户端之间只进行自身梯度信息的通信，通过公开共享的梯度达到联合训练的目的。梯度信息的交换一般被认为是安全的，认为梯度不会导致训练数据的泄露，因此联邦学习被认为是可以有效保护隐私的机器学习框架，被大量应用于隐私数据的训练任务，例如医疗健康，金融保险等隐私数据的学习。

然而梯度真的不会泄露训练数据的信息吗？已有工作已经证明可以从梯度信息中推断出训练数据的某种属性或特征，那么是否有可能从梯度信息中完全暴露出训练数据呢？论文中表示这是完全有可能做到的。

为了表明如何从梯度中完全暴露训练数据信息，论文提出了Deep Leakage from Gradients (DLG) 攻击模型。首先，定义了Deep Leakage from Gradients攻击的威胁模型。在两种分布式联合学习中威胁模型假设的攻击者不同，在有集中参数服务器的框架下，假设攻击者是诚实但好奇的server，希望从上传得到的各个梯度更新$\triangle W$中暴露出training data；在没有中心参数服务器的peer to peer框架中，假设某一个client 是诚实但好奇的攻击者，可以得到邻居



(a) Distributed training with a centralized server　(b) Distributed training without a centralized server

图1 SDN/OpenFlow控制流的示例

client传递的△W，并希望从中暴露出training data。

攻击方法的整体框架如下，图1：

在分布式联合学习框架中，普通客户端（Participant）会先从server下载得到当前的global model(W)，并根据自身私有的数据x和

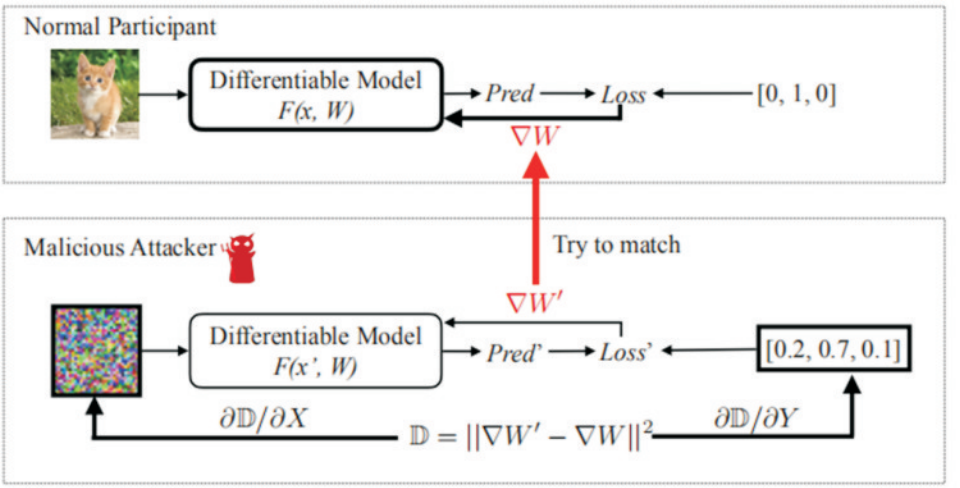


图1 Deep Leakage from Gradients (DLG)框架

对应标签y进行损失函数loss的计算，反向传播得到模型梯度△W。而恶意的攻击者可以得到global model(W)和模型梯度△W的先验知识。攻击者假设模拟数据x'和标签y'为需要优化的变量，经过同样的步骤将x'输入到global model W，计算预测值与标签y'的损失loss'，得到x'与y'对应的模型梯度△W'。攻击者希望模拟数据计算得到的梯度更新能与真实客户端的更新尽量相似，因此将△W与△W'的差距作为实际损失函数以优化x'和y'。多轮优化完成后，模拟数据x'和标签y'将非常接近真实训练数据，达到攻击者窃取训练数据的目的。

下面是算法流程，算法1：

（1）初始时对模拟数据x', y'进行初始化

（2）对x'和y'进行n轮优化，每一次优化过程为：

a. 根据x', y'和当时的global model参数Wt，计算出当前梯度△W'

b. 计算△W'与真实梯度更新△W的欧氏距离 ‖△W'−△W‖，作为损失函数

c. 根据损失函数，分别对x'和y'进行更新优化

（3）返回得到的x'与y'

在Mnist, CIFAR100等数据集上进行实验，结果如图2所示。当优化轮数到达500轮后，模拟数据已经非常接近于Ground Truth的训练数据，这说明论文中的攻击方法是非常有效的。

此外，论文还讨论了对于Deep Leakage from Gradients攻击可能的防御方法，论文中实验了不同程度加噪，梯度裁剪，fp16 convertion 等防御方法对于Deep Leakage from Gradients攻击的防御效果，综合来看基于梯度裁剪的方法对于梯度泄露攻击更有效。

更多的细节和算法在论文中有详细说明，由于空间限制不进行展开，感兴趣的读者

Algorithm 1 Deep Leakage from Gradients.

Input: $F(\mathbf{x}; W)$: Differentiable machine learning model; W: parameter weights; ∇W: gradients calculated by training data

Output: private training data $\mathbf{x}, \mathbf{y}$

procedure DLG(F, W, ∇W)
  $\mathbf{x}'_1 \leftarrow \mathcal{N}(0,1)$, $\mathbf{y}'_1 \leftarrow \mathcal{N}(0,1)$ ▷ Initialize dummy inputs and labels.
  for $i \leftarrow 1$ to n **do**
    $\nabla W'_i \leftarrow \partial \ell(F(\mathbf{x}'_i, W_t), \mathbf{y}'_i)/\partial W_t$ ▷ Compute dummy gradients.
    $\mathbb{D}_i \leftarrow \|\nabla W'_i - \nabla W\|^2$
    $\mathbf{x}'_{i+1} \leftarrow \mathbf{x}'_i - \eta \nabla_{\mathbf{x}'_i} \mathbb{D}_i$, $\mathbf{y}'_{i+1} \leftarrow \mathbf{y}'_i - \eta \nabla_{\mathbf{y}'_i} \mathbb{D}_i$ ▷ Update data to match gradients.
  end for
  return $\mathbf{x}'_{n+1}, \mathbf{y}'_{n+1}$
end procedure

Alg1 DLG算法流程

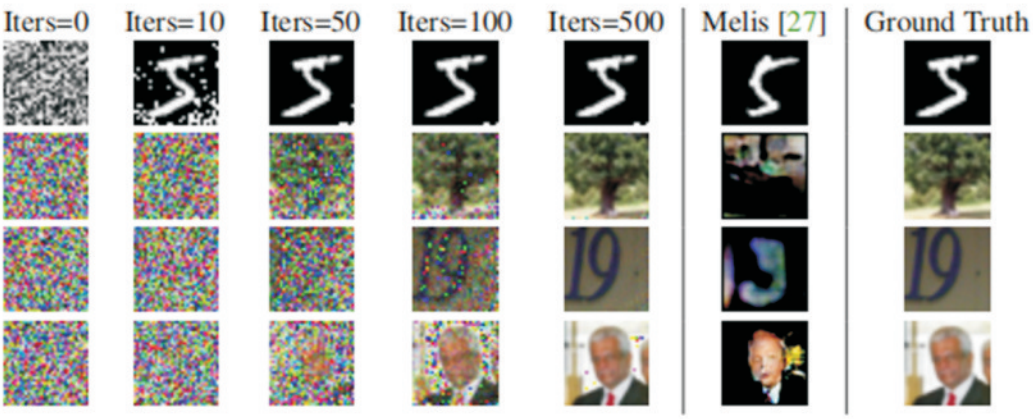


图2 在MNIST, CIFAR-100, SVHN 和 LFW 数据集上的效果展示

可以去论文中查看。总结而言，本文提出了一种新型攻击方法，可以在没有额外先验信息的情况下，从公开共享的梯度信息中恢复出局部客户端的训练数据。其主要贡献在于：（1）说明了从公开共享的梯度中得到私有训练数据是可能的，并且Deep Leakage from Gradients是第一个实现这种可能的算法。（2）Deep Leakage from Gradients只需要得到梯度信息，就可以恢复像素级别准确度的图片和token级别的语句，比起之前需要额外信息且恢复粒度更粗的方法有较大的提升。（3）为了阻止潜在的数据隐私泄露，分析了提出的攻击在不同设定下的攻击难度，并且对可能的防御方法进行了讨论。这篇论文揭示了梯度信息存在的巨大隐私泄露隐患，希望引起对梯度以及梯度共享的机器学习框架的安全性和隐私性的更多思考。

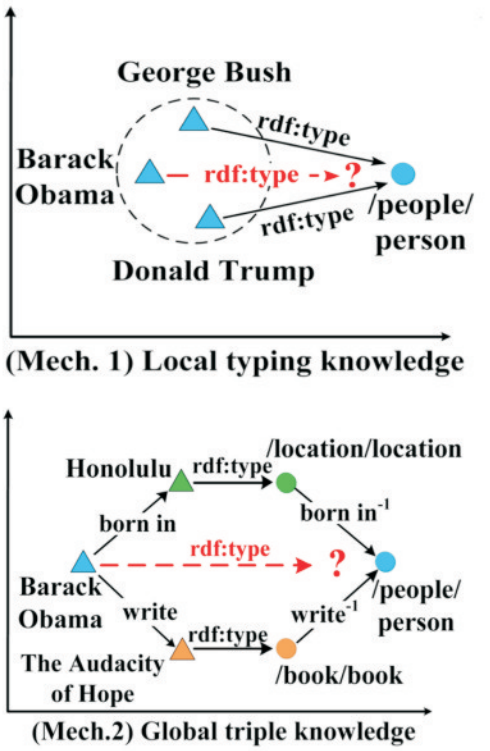
付佳韵
2020级博士研究生
研究方向：人工智能安全，联邦学习
Email：fujiayun1997@hust.edu.cn

Connecting Embeddings for Knowledge Graph Entity Typing

张照博 推荐

“Connecting Embeddings for Knowledge Graph Entity Typing”是自然语言处理顶级国际会议ACL-2020录用的一篇文章。本文主要研究知识图谱中实体的类型补全问题（Knowledge Graph Entity Typing）。这是Knowledge Graph Completion (KGC)的一个子问题。KGC主要面向知识图谱中实体和关系的缺失进行补全，对于实体的类型进行补全的工作比较少见。比如说，FB15k这一知识图谱中，有10%的实体属于/music/artist类型，而没有属于/people/person类型。知识图谱中实体类型的不完全性导致一些知识图谱驱动的任务中的某些类型相关算法产生了严重不足甚至直接不可用。

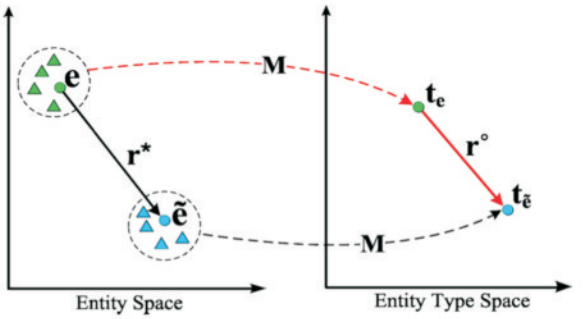
本文提出一种新的方法---ConnectE，该方法通过联合利用现有实体类型中的局部类型化知识和知识图谱中的全局三元组知识进行类型补全。具体来说，有两种不同的知识驱动的实体类型推理的有效机制。第一种是使用图一（Mech.1）中的局部类型知识，可以从嵌入空间中与实体接近的其他实体中找到实体的缺失实体类型。第二种是一个（头部或尾部）实体的缺失实体类型可以通过它们的关系从其他（尾部或头部）实体的类型中推断出来，使用图一（Mech.2）中的全局三元组知识。然后，使用它们的联合模型来推断缺失的实体类型，这样可以同时利用知识图谱中本身存在的实体类型和一些三元组知识的推断。通过在两个真实数据集（Freebase和YAGO）上的多项实验，验证了ConnectE对于改进知识图谱中实体的类型补全的有效性。



图一：类型补全的两种方式

ConnectE的整体架构如图2所示。ConnectE主要分为两个部分，第一部分是将实体从实体空间映射到对应的类型空间中（E2T: Mapping Entities to Types），实现映射 $f_{proj}(e) \simeq t_e, \forall e \in \xi$ 其中$e \in R^k$ 是实体向量，$t \in R^l$ 是实体类型向量。整体的得分函数表示为：

$S_{e2t}(e,t) = \| M \cdot e - t \|_{l2}^2$，其中，M是最终得到的映射矩阵。通过最小化整个知识图谱的上述得分函数，可以得到 $f_{proj}(e) = M \cdot e \simeq t_e$。



图二：ConnectE 的架构

第二部分是使用全局的知识进行建模（TRT: Encoding Triples in KGs）。基于实体三元组(head entity, relationship, tail entity)中头尾节点的实体也需要服从相同的关系这一假设，对于知识图谱中的头实体–关系–尾实有：(head type, relationship, tail type)，即 $t_h + r^o \simeq t_t$，所以可以通过训练头尾实体对应的类型的来表示实体类型。另外，因为类型的数量比实体更少，其表示需要的维度也相对较少，所以在实体空间和实体类型空间中，关系有两种不同的表示：$r^o \in R^l$，$r^* \in R^l$。对于类型三元组，得分函数表示为 $S_{trt}(t_h, r^o, t_t) = \| t_h + r^o - t_t \|_{l2}^2$。上述两部分组合后可以得到整体的一个的得分函数：

$$\mathbf{S}_{e2t+trt}(e,t_e) = \lambda \cdot \mathbf{S}_{e2t}(e,t_e) + (1-\lambda)\cdot\{\frac{1}{|P|}\sum_{t_e\in P}\mathbf{S}_{trt}(t_e,r,t_{\tilde{e}}) \quad + \frac{1}{|Q|}\sum_{t_{\tilde{e}}\in Q}\mathbf{S}_{trt}(t_{\tilde{e}},r,t_e)\}$$

DATASET	FB15kET				YAGO43kET			
METRICS	MRR	HITS@1	HITS@3	HITS@10	MRR	HITS@1	HITS@3	HITS@10
RESCAL (Nickel et al., 2011)	0.19	9.71	19.58	37.58	0.08	4.24	8.31	15.31
RES.-ET (Moon et al., 2017)	0.24	12.17	27.92	50.72	0.09	4.32	9.62	19.40
HOLE (Nickel et al., 2016)	0.22	13.29	23.35	38.16	0.16	9.02	17.28	29.25
HOLE-ET (Moon et al., 2017)	0.42	29.40	48.04	66.73	0.18	10.28	20.13	34.90
TransE (Bordes et al., 2013)	0.45	31.51	51.45	73.93	0.21	12.63	23.24	38.93
TransE-ET (Moon et al., 2017)	0.46	33.56	52.96	71.16	0.18	9.19	19.41	35.58
ETE (Moon et al., 2017)	0.50	38.51	55.33	71.93	0.23	13.73	26.28	42.18
ConnectE-(E2T+0)	0.57 +- .00	45.54 +- .28	62.31 +- .29	78.12 +- .12	0.24 +- .01	13.54 +- .12	26.20 +- .18	44.51 +- .09
ConnectE-(E2T+TRT)(disc.)	0.59 +- .01	48.54 +- .71	63.66 +- .39	78.27 +- .16	0.27 +- .01	15.1 +- .15	29.14 +- .13	47.08 +- .09
ConnectE-(E2T+TRT)(full)	**0.59 +- .00**	**49.55 +- .62**	**64.32 +- .37**	**79.92 +- .14**	**0.28 +- .01**	**16.01 +- .12**	**30.85 +- .13**	**47.92 +- .07**

表一：不同模型的类型补全性能

回到类型补全中，利用上述得分函数在整个知识图谱中进行训练后得到相关的类型t_e、实体e、关系r^o，r^*表示以及转换矩阵M，类型补全可以通过如下两种方式进行：

$$\hat{t}_e = \arg\min_{t\in\mathcal{T}} \mathbf{S}_{e2t}(e,t)$$

$$\hat{t}_e = \arg\min_{t\in\mathcal{T}} \mathbf{S}_{e2t+trt}(e,t_e)$$

在模型的优化方面，ConnectE使用的是经典的翻译模型的优化方案，对E2T，TRT，E2T+TRT三个得分函数在整个知识图谱的实例上进行优化：

$$\mathbf{J}_1 = \sum_{\mathcal{D}}\sum_{\mathcal{D}'}[\gamma_1 + \mathbf{S}(e,r,\tilde{e}) - \mathbf{S}(e',r,\tilde{e}')]_+,$$

$$\mathbf{J}_2 = \sum_{\mathcal{H}}\sum_{\mathcal{H}'}[\gamma_2 + \mathbf{S}_{e2t}(e,t_e) - \mathbf{S}_{e2t}(e',t'_e)]_+,$$

$$\mathbf{J}_3 = \sum_{\mathcal{Z}}\sum_{\mathcal{Z}'}[\gamma_3 + \mathbf{S}_{trt}(t_e,r,t_{\tilde{e}}) - \mathbf{S}_{trt}(t'_e,r,t'_{\tilde{e}})]_+$$

$$\begin{aligned}\mathcal{D}' :=& \{(e',r,\tilde{e})|(e,r,\tilde{e})\in\mathcal{D}, e'\in\mathcal{E}, e'\neq e\}\\ &\cup\{(e,r,\tilde{e}')|(e,r,\tilde{e})\in\mathcal{D}, \tilde{e}'\in\mathcal{E}, \tilde{e}'\neq\tilde{e}\},\\ \mathcal{H}' :=& \{(e',t_e)|(e,t_e)\in\mathcal{H}, e'\in\mathcal{E}, e'\neq e\}\\ &\cup\{(e,t'_e)|(e,t_e)\in\mathcal{H}, t'_e\in\mathcal{T}, t'_e\neq t_e\},\\ \mathcal{Z}' :=& \{(t'_e,r,t_{\tilde{e}})|(t_e,r,t_{\tilde{e}})\in\mathcal{Z}, t'_e\in\mathcal{T}, t'_e\neq t_e\}\\ &\cup\{(t_e,r,t'_{\tilde{e}})|(t_e,r,t_{\tilde{e}})\in\mathcal{Z}, t'_{\tilde{e}}\in\mathcal{T}, t'_{\tilde{e}}\neq t_{\tilde{e}}\}\end{aligned}$$

在FB15k和Yago43k上面进行类型补全的预测实验，得到了较好的结果。对Baselines的全面超越说明ConnectE确实有良好的捕获局部和全局的类型知识的能力。

张照博
2018级博士研究生
研究方向：NLP和知识图谱
Email：zhang_zb@hust.edu.cn

A Decentralized Blockchain with High Throughput and Fast Confirmation

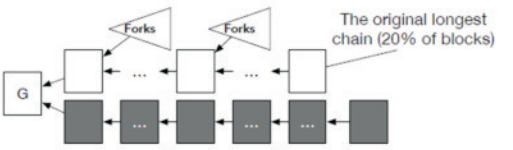
岳镜涛　推荐

“A Decentralized Blockchain with High Throughput and Fast Confirmation”是计算机体系结构领域的顶级会议ATC 2020录用的一篇文章。本文提出了Conflux，一种具有高吞吐量、快速确认和去中心化的区块链系统。

近年来，区块链已经发展成为一种在互联网范围内为安全、去中心化的分布式数据库提供动力的新技术。比特币等经典区块链系统使用中本聪共识。它将交易组织到一个有序的区块列表中，每个区块包含多个交易和指向前一个区块的链接。参与者(矿工)解决工作量证明(PoW)难题，竞争产生下一个区块的权力。为了防止攻击者重写以前的交易，诚实的参与者同意将最长的区块链作为正确的历史记录。每个新区块都追加在最长链的末端，以使最长链更长，并更难被重写。

一个理想的区块链具有以下四个理想属性：安全性、去中心化、高吞吐量和快速确认。然而，目前常用的中本聪共识因其缓慢的区块产生速度而产生性能瓶颈。例如，比特币每10分钟生成一个1MB的块，因此每秒只能处理7笔交易。用户通常必须等待一个小时(即，6个区块)才能获得对交易的不可更改性的高置信度。

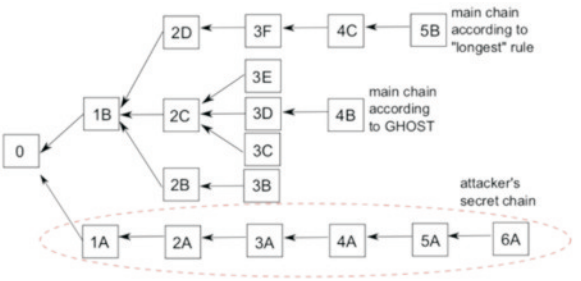
因为产生区块是并发的，并且区块在网络中的传播需要一定的时间，因此如果只是简单提高中本聪共识中的区块产生速度，那么就容易出现图一所示的情况：一个节点在产生新区块之后，其它节点需要等待一段时间才能接收到新区块，在此期间，某个没有收到新区块的节点可能会产生另一个冲突的区块。这种情况在区块链中被称为分叉。传统区块链使用最长链原则来解决分叉。在所有分支当中，最长的那条会成为共识链，只有在共识链上的区块才是有效的。然而频繁的分叉会导致大量诚实节点的算力被浪费在短的分支上，从而缩短共识链的长度，导致攻击者只需要很少的算力便能成功发动攻击。



(a) An attacker with more than 20% of the network computation power can revert the longest chain.

图一　分叉示例

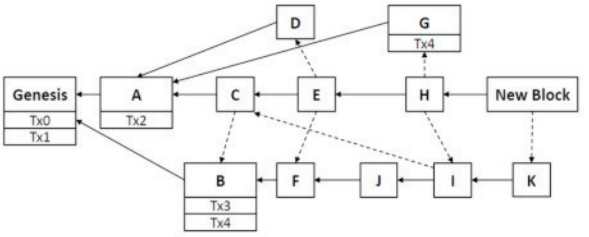
Sompolinsky等人意识到了这个问题，并在15年提出了GHOST共识算法，其核心是最重子树原则，如图二所示。假设在某个时刻，区块链发生了大量分叉。如果使用最长链原则，那么最下面那条链，也就是攻击者产生的那条链成为了最长链，意味着攻击者对系统的攻击成功。如果使用最重子树原则，在每次分叉的时候选择拥有最重子树的分叉节点。比如说，针对1A和1B两条分叉，可以看到1A后面是一条链有5个区块，1B后面有一棵树，里面有11个区块，所以选择1B作为主链。以此类推，选出的主链就是0、1B、2C、3D和4B，主链上的区块都是诚实矿工创建的，攻击者的攻击不会成功。



图二 GHOST协议示例

最重子树原则可以在频繁发生分叉的情况下提供更好的安全保证，但是无法避免分叉造成的区块浪费。比如根据最重子树原则，不在共识链上的区块，例如2B和4C，都会被丢弃，很明显，这仍然会导致很多区块被浪费。因此，性能优化的目标就很明确了，就是要想办法把这些被浪费的区块全部利用起来。

如图三所示，在Conflux中，区块链原本的链式结构改成了有向无环图（DAG）。每个区块是图中的一个顶点，而一个区块指向另一个区块的指针则是一条有方向的边。DAG中的边被分为两类，一类是父边，用实线表示，另外一类是参考边，用虚线表示。每个区块有且只有一条父边，而参考边可以有多条，也可以没有。如果只考虑父边，那么这里的图就退化成了一颗树，论文中称其为树图。实际上，这里所谓的树图就是一条带分叉的区块链。参考边体现的是时间上的先后关系。比如说，E到D有一条参考边，说明D在E之前产生。



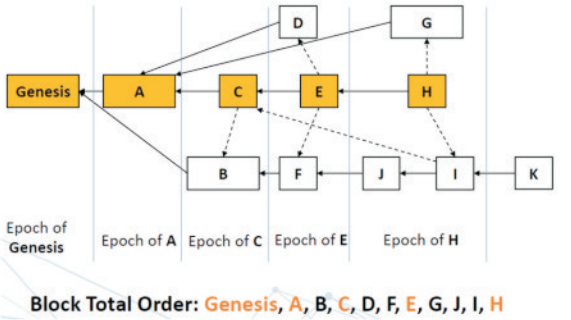
图三 Conflux中的DAG结构

当产生一个新区块时，需要使用前面介绍的GHOST算法算法计算出共识链（本文实际上使用的是本文提出的改进GHOST算法），论文中称其为pivot链。图三中的pivot链是Genesis、A、C、E、H，由于最后一个区块是H，于是新区块将自己的父边指向H。同时，新区块还发现先于自己产生的K没有入边，于是新区块产生一条指向K的参考边。如果有多个K这种没有入边的区块，那么就会产生多条参考边。

如前所述，在GHOST共识算法选出一条共识链后，哪些不在共识链上的区块会被丢弃，这样就浪费了很多产生区块的计算资源和传播区块的网络资源。为了避免浪费，这篇论文提出了一种排序算法，即按照一定的顺序遍历DAG中所有的区块，根据遍历算法访问区块的先后顺序，就可以将DAG转换成一条链。这样就可以将DAG中的所有区块利用起来。

图四显示了排序算法。首先计算共识链pivot链（图四中黄色区块）。接下来，根据pivot链将整个图划分成一个个epoch，每个pivot 链区块对应一个epoch。然后，将那些不在pivot 链上的区块，也就是图四中的白色区块分配到每个epoch。比如说E是第一个存在到达D的路径的pivot链区块，因此D和E属于同一个epoch。最后对所有区块进行排序，具体方法是首先根据epoch的先后顺序进行排序，对于每个epoch内的区块，使用拓扑顺序和区块id进行排序。实际上每个epoch内具体使用哪种排序方法并不重要，只要所有节点都使用相同的确定性算法即可。

排序得到的结果可以看做是一条链。对这条链的操作和标准的区块链相同，如果前面的区块和后面的区块当中记录的数据发生了冲突，那么就以前面的区块为准。比如说如果B中的一笔交易和E中的一笔交易发生了冲突，那么E中的那笔交易就会被无效化。

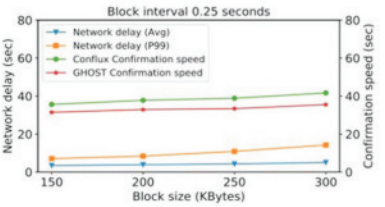


图四 Conflux排序算法

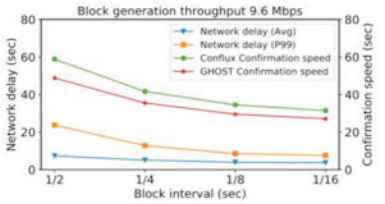
论文还有一些其它工作，例如针对GHOST存在的活性攻击漏洞改进了GHOST算法，以及使用延迟执行减少重复执行。由于篇幅关系，这里就不做过多介绍。

在实验部分，本文对Conflux的吞吐量、确认速度和可扩展性进行系统评估。作者在多达800台Amazon EC2虚拟机上部署了Conflux，并通过插入延迟来模拟城际延迟，同时设置每个节点的网络带宽为20Mbps。Conflux中的节点通过gossip网络连接，每个节点随机选择10个节点连接，从而组成P2P网络。

作者首先评估了区块大小和区块产生间隔对网络传播延迟和区块确认速度的影响，如图五所示。根据实验，作者选择了300KB的区块大小和0.25秒的区块间隔。



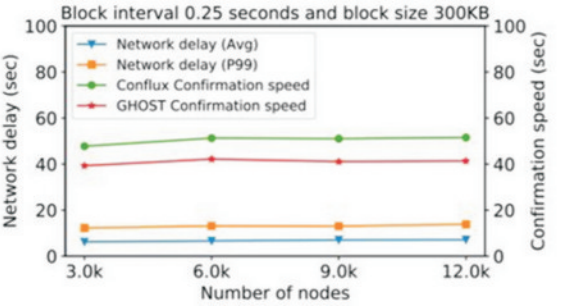
(a) Latency vs. block sizes.



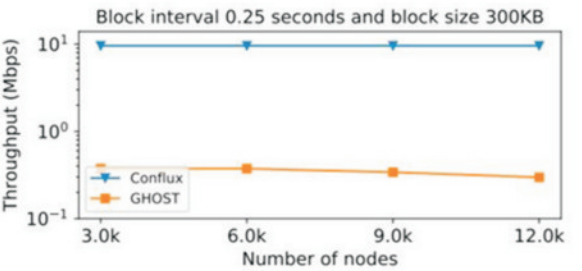
(b) Latency vs. block intervals.

图五 区块大小和区块间隔对传播延迟和确认延迟的影响

接下来作者在大规模网络（3000-12000个节点）中测试了Conflux的吞吐量和延迟。如图六和图七所示，可以看到随着节点数量的增加，延迟和吞吐量基本保持不变。在20Mbps的带宽下，Conflux的交易吞吐量达到了9.6Mbps，大约为每秒数千笔交易。



图六 节点数量增加对区块传播延迟和确认延迟的影响



图七 节点数量增加对吞吐量的影响

总的来说，本文实现了一种新的区块链系统，并在GHOST的基础上提出了一种新的区块链共识算法，它可以实现高吞吐量和快速确认。本文提出的系统的性能只受网络带宽和节点处理能力的限制，在20兆的网络下，可以实现9.6M的有效吞吐量。

岳镜涛
2019级博士研究生
研究方向：大数据
Email：d201980947@hust.edu.cn

Experience: Aging or Glitching? Why Does Android Stop Responding and What Can We Do About it?

黄均浩　推荐

“Experience: Aging or Glitching? Why Does Android Stop Responding and What Can We Do About It?”是计算机网络领域顶级会议MobiCom'20上收录的一篇文章。文章分析了导致安卓系统中出现应用不响应（Application Not Responding，ANR）与系统不响应（System Not Responding，SNR）的原因，并提出了一种降低ANR与SNR发生频率的方法。

响应时间是影响系统的用户体验的一个重要因素，几乎每个安卓用户都经历过应用不响应（Application Not Responding，ANR）和系统不响应（System Not Responding，SNR）。ANR是指前端的应用对用户输入或者系统指令5秒内不做出响应，或者后端应用对系统指令10秒内不做出响应。在这种情况下，系统会询问用户是继续等待应用做出响应，还是终止当前应用。SNR是指重要的系统线程在一分钟内不对指令做出响应，会导致系统强制重启。ANR与SNR极大地影响了安卓系统的用户体验。当前安卓社区对ANR与SNR的特点、分布情况以及出现的根本原因不清晰，且没有较好的解决方法。

本文通过对现有安卓系统进行修改，并在安卓社区里面邀请活跃用户进行安装，收集了超过3000部不同安卓版本、不同硬件配置的手机的有关ANR与SNR的日志信息，并使用机器学习的方法对日志信息进行分析，分析得出安卓系统的ANR与SNR的分布特点、与安卓版本以及硬件配置之间的关系。

经过研究发现，导致ANR与SNR的原因主要有以下四点：安卓的写放大迁移（Write Amplification Mitigation，WAM）的低效（35%）、系统服务之间的锁争用（21%）、内存不够（18%）、应用特定缺陷（26%）。内存不充足和系统服务之间的锁争用是经典的操作系统问题，且当前没有较好的解决方法，而应用特定缺陷也由各个应用决定，也没有较好解决方法。通过深入研究，导致WAM低效的是安卓系统中有缺陷的设计，并且能够通过简单的修复，从根本上解决。

Model	CPU	Memory	Storage	Android Version
1	1.8 GHz	3 GB	32 GB	7.0
2	2 GHz	4 GB	64 GB	7.0
3	2 GHz	4 GB	64 GB	7.0
4	1.8 GHz	6 GB	64 GB	9.0
5	1.8 GHz	6 GB	64 GB	7.0
6	2.2 GHz	4 GB	64 GB	8.0
7	2.2 GHz	4 GB	64 GB	9.0
8	2.2 GHz	6 GB	64 GB	7.0
9	2.2 GHz	6 GB	64 GB	8.0
10	2.2 GHz	6 GB	64 GB	7.0
11	2.3 GHz	6 GB	64 GB	8.0
12	2.8 GHz	6 GB	128 GB	8.0
13	2.8 GHz	8 GB	128 GB	9.0
14	2.84 GHz	8 GB	128 GB	9.0
15	2.84 GHz	8 GB	128 GB	9.0

表1：参与测试的手机的硬件以及系统配置

测试中采用了不同硬件以及系统配置的手机，参与测试的手机的系统以及硬件配置如表1所示。测试中发现，ANR与SNR是普遍发生在每种型号的手机上的，且ANR与SNR的发生频率与手机型号有较强相关性的。而且ANR和SNR的发生频率并没有随着系统版本的迭代以及硬件配置的提高，发生显著降低。每种型号的单个手机发生ANR与SNR的平均次数，如图1所示。本文还对ANR与SNR发生的时间上是否相关进行了研究，发现二者发生的时间间隔较大，二者的发生不存在因果关系。ANR与SNR发生时间之间的间隔如图2所示。

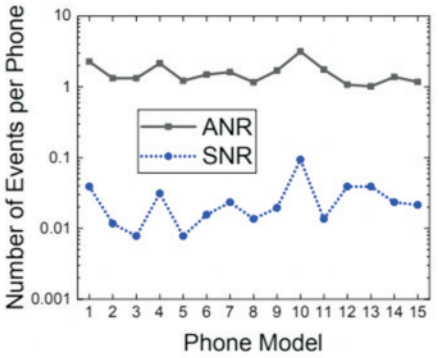


图1 每种型号的单个手机发生ANR与SNR平均发生次数

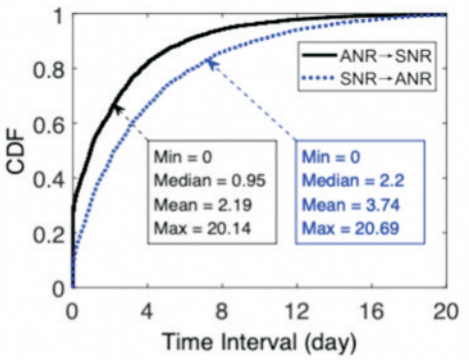


图2 ANR与SNR发生时间间隔

有关使用机器学习的方法，对日志进行分析，并得出主要原因的部分，不展开叙述，详见论文。

安卓的WAM实现有两种机制，一种是实时的WAM，一种是成批WAM，实时的WAM会在不同应用对同一个page上文件进行删除与写入发生冲突，而成批进行WAM，进行频率过低（一天最多一次），且不能被打断，倘若在进行批处理WAM时，用户使用手机，会发现相应时间较长，一旦被中断，只能重新开始。本文对现有的成批WAM进行了改进，提出了一种数据驱动型的WAM。即设置一个阈值，倘若系统中删除的文件的总量达到该阈值，即进行批处理WAM。读写速度与批处理时间WAM持续时间之间的关系如图3所示，当阈值设置为6GB，能够实现二者的均衡。

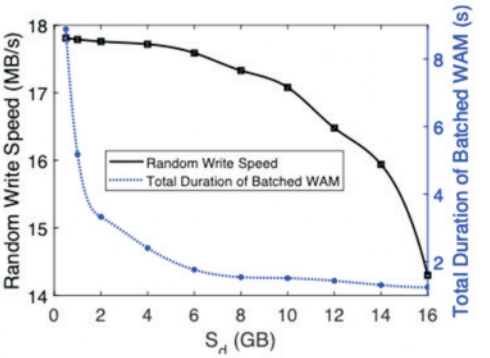


图3 批处理WAM时间和随机读写速度与阈值之间的关系

测试表明，这种方法能够将由WAM导致的ANR与SNR发生频率降低99%，从整体上把ANR与SNR发生频率分别降低32%与47%，而仅仅将数据传输效率降低了3%。

黄均浩
2020级博士研究生
研究方向：新型操作系统、虚拟化
Email：huang.jun.hao@foxmail.com

Livia: Data-Centric Computing Throughout the Memory Hierarchy

陈申彪　推荐

“Livia: Data-Centric Computing Throughout the Memory Hierarchy”是体系结构领域的国际顶级会议ASPLOS’20录用的一篇文章。该文提出了一种在存储器层次结构中运行以数据为中心的计算的内存服务模型，并设计和实现了适用于该内存服务模型的微系统架构Livia，大大提高了系统处理不规则工作负载(irregular workloads)的性能。

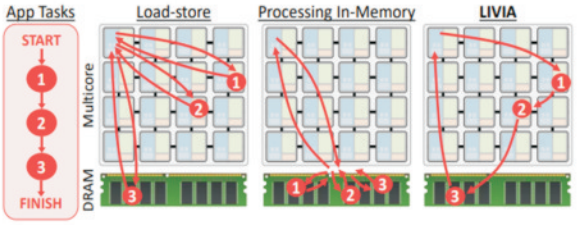


图1 Load-Store、PIM和Livia系统

文章首先分析了Load-Store系统和存内计算(PIM)系统处理数据不规则工作负载的性能表现。如图1所示，当load-Store系统选择在左上核心中执行任务时，需要将所需的数据都传递到该核心中，造成了大量不必要的数据移动；PIM 系统虽然拉近了任务与数据的距离，但是需要执行昂贵的DRAM访问。于是，本文提出了内存服务模型，旨在加速瓶颈在数据访问上的数据不规则的工作负载。内存服务将应用程序功能分解为一个任务链，Livia将这些任务分配在存储器层次结构中的最佳位置，获取本地数据并执行。如图1所示，Livia最大程度地消除了不必要的数据移动，并利用本地数据减少了不必要的DRAM访问，大大提高了系统的运行效率。

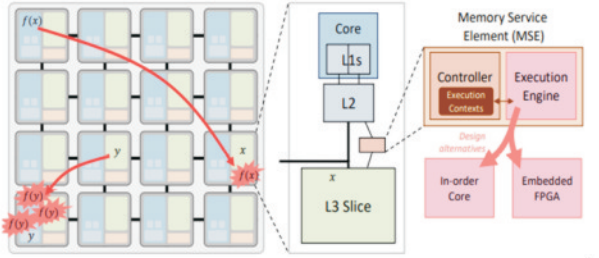


图2 Livia架构

文章接着详细介绍了Livia系统的设计与实现。图2是Livia的系统架构图。Livia将内存服务元素（MSE）添加到多核的每个tile中。MSE包含一个用于迁移任务和数据的控制器和一个可以高效地执行数据附近的任务执行引擎。

MSE控制器主要处理来自核心和其他MSE的调用消息，并将任务分配到层次结构中的最佳位置，以最大程度地减少数据移动。MSE通过遍历存储层次结构来确定最终执行任务的位置。在每个层次结构中，若数据存在，则在本地执行任务；若数据不存在，则将任务传递给下一个层次结构中，重复进行判断。最终使得任务执行在数据附近，尽量减少不必要的数据移动并充分利用了本地数据。图3是Livia 任务调度的三个例子，完整展示了对于不同位置的数据x、y和z，MSE控制器的进行任务调度的工作流程。

PaGraph: Scaling GNN Training on Large Graphs via Computation-aware Caching and Partitioning

何海恒 推荐

“PaGraph: Scaling GNN Training on Large Graphs via Computation-aware Caching”是云计算领域顶级会议SoCC 2020年录用的文章，该会议于2020年9月线上举行。

图神经网络已经广泛用于探索和处理图结构数据。对于大规模图数据集，现有的图神经网络框架利用 GPU 等加速器并结合图采样和批处理方式来有效地训练图神经网络。然而，现有的技术面临着数据加载时间长、可拓展性差等问题。本文提出 PaGraph, 在单机多 GPU 机器上采用图缓存策略和图分割策略来有效提升图神经网络训练性能。

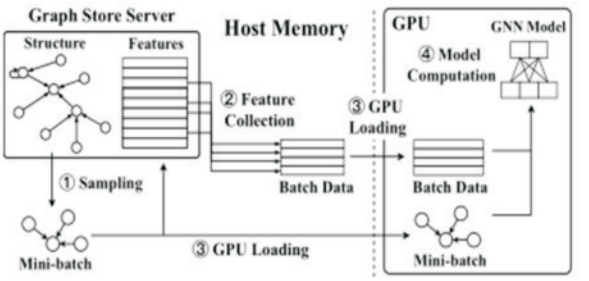


图1 基于采样的GNN模型

接上页

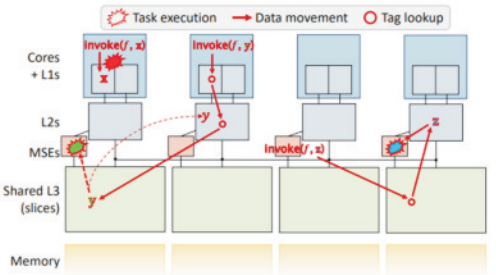


图3 MSE控制器的任务调度实例

MSE执行引擎是Livia在存储层次结构中运行任务的组件。执行引擎可以用两种方式来具体实现，分别为in-order cores和embedded FPGAs。前者易于实现，后者性能更好。作者为了完整评价内存服务模型和Livia系统在处理不规则的工作负载的性能表现，分别实现了Livia SW和Livia FPGA两个架构模拟器。

文章最后，作者分别测试了在Load-Store系统、PIM系统，Hybird PIM系统，Livia SW系统和Livia FPGA系统中处理各种不规则工作负载时的性能表现，包括Lock-free-lookup AVL tree，Linked lists，Graph analytics: PageRank和Producer-consumer queue。实验结果表明，相较于其他的系统，两个Livia系统在处理不规则工作负载时的性能表现都更加出色。

陈申彪
2020级博士研究生
研究方向：数据流、存内计算
Email：d202081083@hust.edu.cn

作者对于基于采样的GNN训练模型进行优化，通过以往的实验观察发现，基于采样的GNN 训练有以下的问题：1）数据加载效率低下：数据加载比计算花费的时间长得多；2）冗余顶点访问模式：由于图中的一个顶点可能与多个训练顶点相连,某些顶点被多次加载；3）GPU资源未充分利用：无论是计算资源还是GPU内存，均未得到有效的利用。

针对以上问题，作者首先提出了一种缓存策略。由于图中一个顶点的出度越高，那么它成为别的顶点的邻居节点的概率就越高，也就更容易被采样，因此顶点的出度表明了它在整个周期中被采样的概率。缓存策略即预先对顶点的出度进行离线排序，并选取具有较高出度的顶点填充GPU缓存。针对缓存策略，为了避免高优先级训练计算的资源竞争，需要预先估计缓存分配的最大可用GPU内存量。基于采样的小批量训练使用的是数量几乎相同的数据样本作为输入，并且执行几乎相同的计算量来训练每个迭代的共享GNN模型。因此，通过对GPU内存使用情况的一次采样来决定缓存大小就足够了。

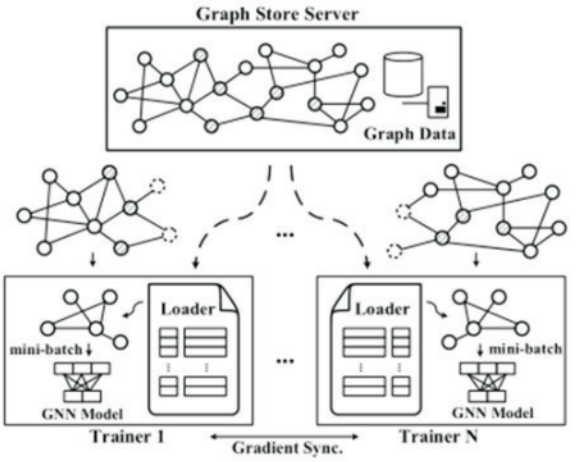


图2 PaGraph的结构

当在多个GPU上同时训练GNN模型时，因为单个图为多个GPU上的并行训练器提供数据访问位置，因此所有GPU缓存都将保持相似的顶点，这导致缓存命中率随着GPU数量的增加而不断降低。因此作者提出了一种数据并行的方法，每个GPU不需要访问公共的图，而是使用自己的数据分区（子图），通过训练计算得到局部梯度，然后交换梯度来同步更新模型副本。这样可以改进数据的局部性，并增加缓存顶点的总数，从而提高了缓存的命中率。而这就涉及到一种图划分算法，既要保证每个GPU的计算负载均衡，又要尽可能的比避免GPU之间跨分区的数据访问。为了保持不同训练器之间的计算平衡，需要保证不同分区的训练顶点个数相近。假设有K个分区，扫描所有的训练点，将其分配到其中一个分区中。对于扩展顶点，只引入必要的边，以满足训练过程中所需的信息。扩展的顶点可能包括训练点，这些扩展的训练点被视为镜像，不会被训练。这样，分区就相互独立了。

在实验部分，作者对GCN和GraphSage这两种经典的图神经网络进行性能评估，与当时性能最好的DGL相对比，在单GPU取得了4.8倍的性能提上，在多GPU上均取得了4.9倍（4GPU）性能提升。作者测量了所设计的缓存策略带来的优势，即对所有的实验来说，均大幅度降低了加载到GPU中的数据总量（降低80%），进而提高了整体性能。另一方面，对于图划分算法，实验结果表明PaGraph有着2.3倍的性能提升，而划分图只需要离线进行即可。

何海恒
2020级博士研究生
研究方向：gnn加速
Email：d202081100@hust.edu.cn

DeepMutation++: A Mutation Testing Framework for Deep Learning Systems

王 健 推荐

“DeepMutation++: A Mutation Testing Framework for Deep Learning Systems”是软件工程领域的顶级会议IEEE/ACM International Conference on Automated Software Engineering 2019收录的一篇论文，该论文介绍了一种专门用于深度神经网络（DNNs）的变异测试框架DeepMutation++，它有利于DNN的质量评估，支持前馈神经网络（FNNs）和循环神经网络（RNNs）。它不仅能够对DNN模型的鲁棒性进行整体静态分析，还可以通过动态分析来识别顺序输入（例如音频输入）中的脆弱部分。

在过去的十几年里，深度学习（DL）在许多领域取得了巨大的成功，然而随着最近一系列与DL相关的事故的发生，DL系统的稳健性和安全性成为一个大问题。目前，DL系统的性能主要是根据模型在测试集上的精度来衡量的，测试数据的质量，对于DL系统的鲁棒性和通用性测试至关重要。在传统软件中，变异测试是分析测试数据质量的一项重要技术。其关键思想是通过向原程序中注入错误来生成一组变异程序，测试数据区分原始程序和变异体的能力代表着测试数据的质量。

论文借鉴了传统软件的变异测试的思想，提出了DeepMutation++，一个针对FNN和RNN的变异测试框架。DeepMutation++设计了8个模型级变异算子，用于直接变异DNN模型，将潜在的故障注入到模型中。论文进一步提出了9个专门用于RNN的变异算子，基于RNN的特点，DeepMutation++既支持静态变异，对测试数据进行整体分析，又支持动态变异，在运行时检测测试数据的脆弱片段。因此，通过分析变异测试结果，可以评估DNN模型的鲁棒性和测试数据的质量。

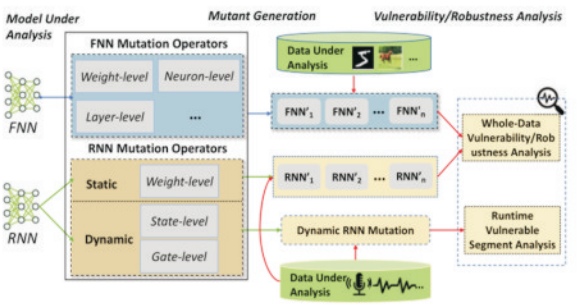


图1 DeepMutation++框架的工作流程

图1显示了DeepMutation++框架的简化工作流程。给定一个DNN模型和一组待分析的测试数据，DeepMutation++首先利用提供的变异算子生成一组高质量的DNN变异体，并设置由用户指定的阈值。在产生一定数量的变异体后，分析原DNN模型与产生的DNN变异体的行为差异，为鲁棒性分析提供输入。此外，DeepMutation++特别为RNN提供了一种动态模式，分析测试输入中脆弱的输入段。最后，输出分析报告，同时表明测试数据的质量和DNN模型的鲁棒性。

论文主要研究两种典型的RNN，LSTM和GRU。除了基本权值外，RNN通常由门单元和存储单元组成，不同的门将模型权值分离成多个部分。考虑到RNN在变异测试中的独特特点和使用案例，DeepMutation++从静态和动态两个层次支持RNN变异算子。

表1 RNN变异算子

Level		Operator	Description
Static	Weight	Weight Gaussian Fuzzing (WGF)	fuzz weights
		Weight Precision Reduction (WPR)	reduce weights' precision
Dynamic	State	State Clear (SC)	clear the state to *0*
		State Reset (SR)	reset state to previous state
		State Gaussian Fuzzing (SGF)	fuzz state value
		State Precision Reduction (SPR)	reduce state value's precision
	Gate	Gate Clear (SC)	clear the gate value to *0*
		Gate Gaussian fuzzing (GGF)	fuzz gate value
		Gate Precision Reduction (GPR)	reduce gate value's precision

表1总结了论文提出的RNN变异算子。静态级变异算子用于执行离线变异模型的生成。生成的变异体存储在用户指定的位置，之后可以执行后阶段变异分析。但是，静态生成的变异体只允许对测试输入的质量进行整体分析。为了实现测试输入的段级分析，论文提出了动态变异算子，可以改变RNN动态段的内部运行状态(即内存状态、门控制状态)，生成变异体。

为了分析模型对输入的鲁棒性，DeepMutation++将原始模型正确分类的所有测试数据放到每个变异体中进行预测，对预测结果进行分析。参考用于传统的变异测试评价指标的killing score，论文提出了一组适用于模型变异测试的评价标准：

（1）一个DNN 模型m及其变异体m'，定义：如果模型针对输入t的输出不一致，即m（t）≠m'（t），则t被m'杀死。给定一组变异DNN模型数据集M，论文将killing score定义为：

$$KS_1(t,m,M)=\frac{|\{m'|m'\in M\wedge m(t)\neq m'(t)\}|}{|M|}$$

（2）给定输入t的第i个分段t_i，RNN模型m，以及通过使用动态变异算子、针对t_i进行变异生成的变异体m'。给定一组变异RNN模型M，定义段级killing score为：

$$KS_2(t_i,m,M)=\frac{\sum_{m'\in M}\|prob(t_i,m)-prob(t_i,m')\|_p}{|M|}$$

对于模型输入，KS_1的值越大，模型针对输入的鲁棒性就越差。对于输入的一部分，KS_2的值越大，模型针对该部分的健壮性就越差。

此外，论文还讨论了针对FNN的检测与更进一步的鲁棒性分析进行了讨论，论文中实验了不同程度的实验，验证了提出的killing score的有效性，对常见的几种神经网络进行了鲁棒性评估。更多的细节和实验结果在论文中有详细说明，感兴趣的读者可以去论文中查看。该篇论文是对DeepMutation: Mutation Testing of Deep Learning Systems中提到的工作的改进与优化，建议读者结合前期工作进行阅读。

总结而言，论文提出了DeepMutation++，一个用于深度学习系统变异测试的框架。DeepMutation++支持为FNN和RNN静态生成高质量的变异体，并动态改变RNN的运行时状态，在分析DNN鲁棒性、模型输入脆弱性和相应的脆弱片段中有着显著的效果。其主要贡献在于：（1）验证了DeepMutation++框架在两种典型场景下对DNN鲁棒性分析和测试数据脆弱段检测的有效性.（2）报告了框架在DNN变异体生成中的效率以及稳健性分析结果。（3）论文发现，基于变异测试定义的指标可能是DNN鲁棒性的一个重要指标，与DNN抗攻击的鲁棒性有很强的相关性。

王　健
2020级博士研究生
研究方向：模型安全
Email：hust_jianw@qq.com

Smart Contract Vulnerability Detection Using Graph Neural Networks

祝 语 推荐

“Smart Contract Vulnerability Detection Using Graph Neural Networks”是人工智能领域的国际顶级会议International Joint Conferences on Artificial Intelligence Organization (IJCAI) 在2020年收录的一篇论文。该文构造了一个合约图来表示智能合约功能的句法和语义结构并对图进行归一化。随后，在此基础上提出了一种无度图卷积神经网络（DR-GCN）和一种新颖的时间消息传播网络（TMP），从归一化合约图中学习并运用于漏洞检测。

智能合约是1994年，由密码学家Szabo提出的，其实质是对现实世界中的合约执行电子化的量化交易协议，一旦部署之后就能实现自我执行。但在当时，因为可信执行环境的缺乏，这项技术始终没有被运用到实践中，而区块链的诞生为这个技术的大规模运用提供了基础。区块链的特殊结构使得想要更改存储在区块链中的数据变得极为困难，如图1所示。

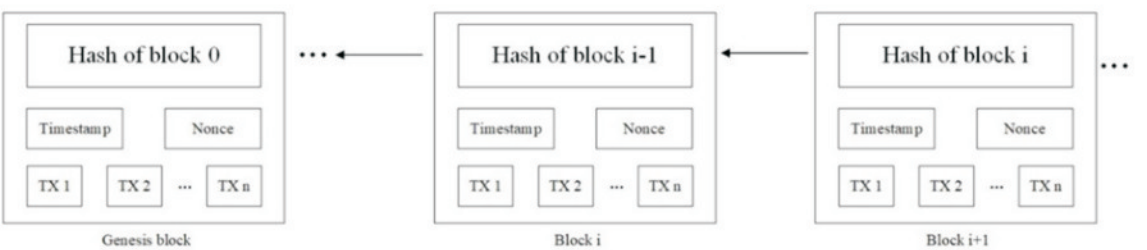


图1 区块链结构

智能合约的编写与传统的编程存在一定的差异，导致其本身也存在很多特有漏洞，这进一步增加了智能合约开发的难度。又因为智能合约本身通常与金融联系在一起，漏洞攻击者可以从中获得巨大收益，这意味着智能合约中的漏洞可能导致更严重的后果，这也使得其更容易遭受恶意攻击以及漏洞利用。由于区块链的特性，智能合约一旦部署在链上就不能修改，在发现漏洞后无法使用传统方法对其进行升级或者打补丁，因而在部署之前进行充分且详尽的智能合约安全性审计是有意义且必要的。

本文是智能合约领域首个将深度学习运用于漏洞检测的工具，其将合约漏洞检测任务建模为分类任务，随后设计了一种新型图神经网络来进行训练，最后根据输出的标签来判断当前合约是否存在相关漏洞。

首先作者为每个智能合约设定一个标签 $\hat{y}$，如式(1)所示。

$$\hat{y}=\begin{cases}0, & \text{合约不存在漏洞}\\ 1, & \text{合约存在漏洞}\end{cases} \tag{1}$$

随后将智能合约形式化为合约图，并且为智能合约中不同的程序元素分配了不同类型的节点。一个合约图中包含三种类型的节点：主要节点、次要节点、后备节点。其中主要节点

表示对自定义或内置函数的调用，次要节点则用于建模关键变量，后备节点则是来激发攻击合同的后备功能。

作者为了描述了被测试中的合约功能可能遍历的路径，构造了边以对节点之间的关系建模。这个过程中，作者将边建模为(V_s,V_e,o,t)这样的一个四元组。其中V_s和V_e分别标识边的起始节点和结束节点，o表示时间顺序，t表示边缘类型。为了捕获节点之间丰富的语义依赖性，作者构造了四种类型的边，如图2所示。

Symbol	Semantic Fact	Type
AH	assert{X}	Control-flow edges
RG	require{X}	
IR	revert	
IT	throw	
IF	if{X}	
GB	if{...} else {X}	
GN	if{...} then {X}	
WH	while{X} do{...}	
FR	for{X} do{...}	
AG	assign{X}	Data-flow edges
AC	access{X}	
FW	natural sequential relationships	Forward edge
FB	interactions with fallback function	Fallback edge

图2 边的语义信息

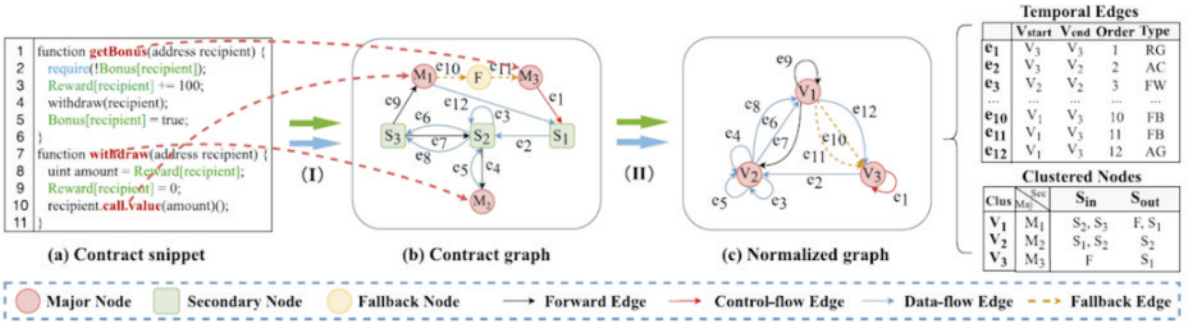


图3 图的生成及其标准化

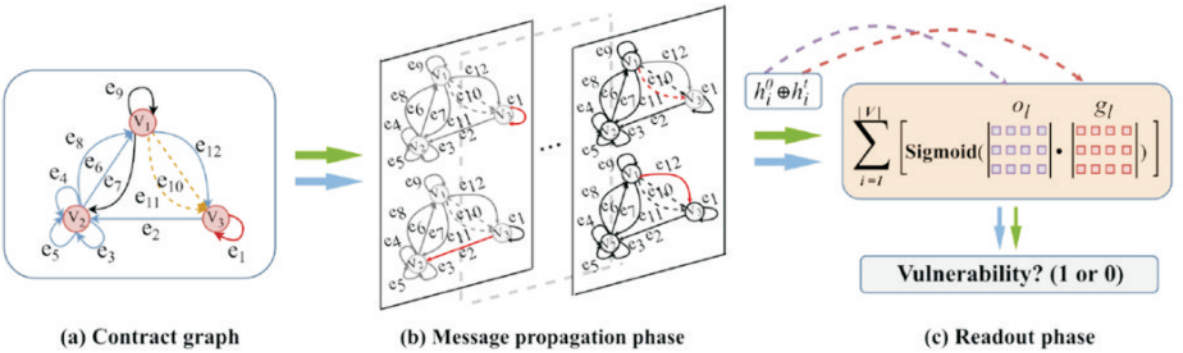


图4 TMP网络

至此已经完成了合约图的构建，在实际过程中，作者发现不同的合约源代码会产生不同的图，从而阻碍了图神经网络的训练。因此，作者考虑对于合约生成的图进行标准化。这个标准化过程主要包含两种方法：其一是节点消除，其二是节点特征整合，限于篇幅，这块的细节感兴趣的可以查阅原文。

整个智能合约图的生成及标准化如图3所示。

本文的另一个贡献是提出了一个新颖的图神经网络—TMP网络。TMP网络由消息传播阶段与消息读出阶段组成，如图4所示。在消息传播阶段，TMP遵循其时间顺序依次沿边传递信息。最后在消息读出阶段，为该图计算一个标

签值。形式上，我们将智能合约的图表示为$G=\{V,E\}$，其中V表示所有的主要节点，E表示所有的边构成的集合。$E=\{e_1,e_2,..,e_n\}$，其中e_k表示第k个时间边缘。

消息传播阶段：消息沿边传递，每个时间步沿一个边。在第0个时间步时，利用V_i的特征初始化每个节点V_i的隐藏状态h_i^0。在第k个时间步时，消息流过第k个时间边缘e_k并更新V_{ek}的隐藏状态，即e_k的末端节点。特别地，消息m_k的计算遵循以下公式；

$$\begin{aligned} x_k &= h_{sk} \oplus t_k \\ m_k &= W_k x_k + b_k \end{aligned} \tag{2}$$

矩阵Wk和偏置向量b_k是网络参数。原始消息x_k包含来自e_k的起始节点和边缘e_k本身的信息，然后使用W_k和b_k将其转换为向量嵌入。收到消息后，e_k的末端节点通过汇总来自传入消息及其先前状态的信息来更新其隐藏状态h_{ek}。形式上，h_{ek}的更新过程遵循以下公式：

$$\begin{aligned} \hat{h}_{ek} &= tanh\left(Um_k + Zh_{ek} + b_1\right) \\ h'_{ek} &= soft\,max\left(R\hat{h}_{ek} + b_2\right) \end{aligned} \tag{3}$$

其中U，Z，R是矩阵，而b_1和b_2是偏差值。

消息读出阶段：在连续遍历G中的所有边之后，TMP通过读取所有节点的最终隐藏状态来计算G的标签。令h_i^T表示第i个节点的最终隐藏状态，我们可以通过以下公式生成预测标签值。

$$\hat{y} = \sum_{i=1}^{|V|} f(h_i^T) \tag{4}$$

考虑到最终隐藏状态h_i^T和原始隐藏状态h_i^0之间的差异在漏洞检测任务中是有帮助的，因而我们考虑将该差异纳入到标签值的计算中去，标签值的计算公式如下所示。

$$\begin{aligned} s_i &= h_i^T \oplus h_i^0 \\ g_i &= softmax(W_g^{(2)}(tanh(b_g^{(1)} + W_g^{(1)} s_i)) + b_g^{(2)}) \\ o_i &= softmax(W_o^{(2)}(tanh(b_o^{(1)} + W_o^{(1)} s_i)) + b_o^{(2)}) \\ \hat{y} &= \sum_{i=1}^{|V|} Sigmoid(o_i \odot g_i) \end{aligned} \tag{5}$$

利用上述流程对网络进行训练，然后，将训练好的模型用于对归一化图进行检测并产生漏洞检测标签值，从而判断该合约是否存在相关漏洞。

在该实验中，作者采用现实世界以太坊上真实运行的合约来进行训练，通过人工对其标记，从而形成一个训练集。

在本文中，作者提出一种用于智能合约的全自动漏洞分析器。与现有方法相比，该方案显式地对智能合约的后备机制进行建模，考虑程序元素之间的丰富依赖关系，并探索使用新颖的图神经网络进行漏洞检测的可能性。实验结果表明，作者提出的方法优于现有的检测技术。这证明深度学习方法在智能合约漏洞检测任务中也存在着巨大的潜力和可能。

祝　语

2020级博士研究生

研究方向：区块链安全、智能合约安全

Email：zy_hust@hust.edu.cn

开拓视野、合作交流会让科研事半功倍

吴月明

不知不觉，今年是我在实验室度过的第五个年头。从最开始的迷茫、无措到现在的平静、坚定，一路走来，有挫败也有成长，有沮丧也有欢乐，有压力也有动力。这一次，我想分享一些关于论文发表的粗浅体会，希望对大家有所帮助。

想要发表一篇好的论文，我们需要具备以下几个条件：1）创新的想法；2）扎实的实验；3）规范的写作。

刚开始踏入科研领域，作为初学者常常不知道如何下手。很感谢邹德清老师对我的指导，他告诉我量变产生质变，创新的想法是建立在大量阅读基础上的。首先要有一个初步方向，需要调研其相关联的论文，精读甚至重现以加深对文章的理解，为自己打下牢固的基础，进而对此观点做改进，或者做技术上的优势融合。其次，要开拓视野，不局限于自己方向的文章，还可以去阅读其他交叉方向的论文。比如我是做安全方向的，我会去阅读人工智能、数据挖掘、软件工程方面的论文。博三时，当我读完一篇社交网络方向的论文后，突然发现它好像可以应用在恶意软件分析领域。如果把应用的函数看作是单个人，函数和函数之间的调用关系看作是人与人之间的交流关系，那么应用的函数调用图就可以看作是一个社交网络，然后采用社交网络方面的技术来对恶意软件进行恶意行为分析。所以当视野开拓后，看到和想到的会更加全面和立体，也更容易产生好的idea。

有了创新的想法，还需要化理论为行动，需要扎扎实实地去做实验。在这个过程中，如果我们化身bug侠，不要羞于向别人寻求帮助，导师、师兄师姐或者其他同学都是很好的请教和交流的对象。很可能我们遇到的bug别人之前也遇到过，这样一问就解决了，大大地节省了时间，提高了效率。同样的，idea实现后，可能实验测试的效果却不理想，这个时候就需要有耐心，一步一步调整思路。和导师多交流，和其他合作的老师多交流，头脑风暴总比自己一个人闷头单干强。当获得初步理想的实验结果之后，还需要对实现的idea进行不同角度的测试和分析，例如可扩展性和鲁棒性。总而言之，扎实的实验才是我们文章的坚厚基石。

最后是写作，规范的写作能大大提高文章的录用率。建议大家在写作前多看顶会文章，仔细分析它们的整体框架，你会发现这些文章的框架大致上都一样。我们可以套用框架，并逐步填充完善自己的内容，也可以参照它们的风格来一点点修改自己的内容。当“模仿”文章写多了之后，有了积累，慢慢就会形成自己的风格，后面就可以按照自己的风格来写作了。另外是写作语句，大家在创作之初都会遇到一个问题：为什么表达同样的意思我们的语句读起来那么别扭？其根本原因就是写的太少，积累的太少。这里没有捷径可走，只有多看多练多写，建议大家去看《经济学人》这样的“权威”杂志，取他人之长补自己之短，尝试着将它们的经典语句应用在自己的写作上。

以上就是我这几年在学习和科研上的些许分享，希望能对大家有所帮助，也祝愿各位同学能精益求精，再接再厉，在牛年冲“顶”成功！

吴月明
2016级博士研究生
研究方向：恶意软件检测，漏洞检测
Email：wuyueming@hust.edu.cn