RH
RQ

Bringing great research ideas into open source communities

Miroslav Bureš

"Research is an adventure": Putting theory to the test at the university and in the field

+

Unikernel Linux moves forward

Generative AI: What does it mean for open source?

"Open source opens doors": mentoring students for success at UMass-Lowell

Red Hat
Research Quarterly
Volume 5:2 | August 2023 | ISSN 2691-5251

MOC ALLIANCE

MAKING THE CLOUD LESS, WELL, CLOUDY

The Mass Open Cloud Alliance (MOC Alliance) is a collaboration of industry, the open-source community, and research IT staff and system researchers from academic institutions across the Northeast that is creating a production cloud for researchers. Of course, a collaboration is only as good as its collaborators.

Follow the MOC Alliance as they create the world's first open cloud.

@mass-open-cloud

www.massopen.cloud

contact@massopen.cloud

Housed at:

Boston University Rafik B. Hariri Institute for
Computing and Computational Science & Engineering

BOSTON
UNIVERSITY

RESEARCH
QUARTERLY

VOLUME 5:2

Table of Contents

Departments

Features

ABOUT RED HAT Red Hat is the world's leading provider of open source software solutions, using a community-powered approach to provide reliable and high-performing cloud, Linux®, middleware, storage, and virtualization technologies. Red Hat also offers award-winning support, training, and consulting services. As a connective hub in a global network of enterprises, partners, and open source communities, Red Hat helps create relevant, innovative technologies that liberate resources for growth and prepare customers for the future of IT.

NORTH AMERICA
1 888 REDHAT1

EUROPE, MIDDLE EAST,
AND AFRICA
00800 7334 2835
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
info-latam@redhat.com

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

Red Hat

RESEARCH QUARTERLY

VOLUME 5:2

From the director

About the Author

Hugh Brock is the Research Director for Red Hat, coordinating Red Hat research and collaboration with universities, governments, and industry worldwide. A Red Hatter since 2002, Hugh brings intimate knowledge of the complex relationship between upstream projects and shippable products to the task of finding research to bring into the open source world.

The uncertainty principle

by Hugh Brock

One of the funny things about research is you never know what you're going to get. In fact, the uncertainty of research is not just unavoidable—it's desirable. Scientific breakthroughs like penicillin and even X-rays were the result of attentive scientists noticing something interesting while pursuing something else, then applying the same rigor to the new path that they would have for their original thesis. If this teaches us anything beyond the virtue of attention to detail in research, surely it is that in planning and especially funding it, we must pay as much attention to the researcher doing the work and the field they work in as to the specific question they propose to answer.

What launched me down this line of thinking is our long-running UKL project, with the original aim of producing a Linux-based unikernel. A unikernel is a software application that is built into a single binary along with the kernel and operating system that will support it, and that runs in the same privileged space as that kernel. Although there are, of course, security concerns with running an application this way, major performance gains can be realized through bypassing parts of the kernel not needed for the particular application running in this mode. Rich Jones' piece in this issue on the research to date describes the successful results in detail, as well as the latest twists and turns that may make building the application along with the kernel unnecessary. Instead, we may have actually discovered a way to write a user-space application that can effectively act as a device driver through controlled privilege escalation. This is not at all what we were looking for, but it may turn out to be substantially more useful. We've had similar happy accidents applying machine learning to compiler optimization, as well as in tuning dynamic systems for energy efficiency.

Although I'm sure Czech Technical University (CTU) researcher Miroslav Bureš understands the value of uncertainty in research as well as anyone, his work in testing unreliable systems is mostly devoted to reducing uncertainty as much as possible. Red Hat research scientist Martin Ukrop interviews him in this issue to talk about how he designs experiments to simulate unreliable connections in large IoT installations, providing a testbed for system designers. His work, grounded in the long-time collaboration between Red Hat Czech and CTU, is finding applications with NATO troops as well as commercial systems.

And one last word on uncertainty. We thought it was past time to take a good look at the Generative AI systems that have taken the world by storm over the last year, to understand both what they are and what they mean for open source development. In this issue, our AI leader in Red Hat Research, Sanjay Arora, collaborates with leading Red Hat software licensing expert Richard Fontana to paint a comprehensive picture of what generative AI is, what it can actually be used for, whether it can be trusted (not in most cases, in my view), and whether it makes sense to talk about models being "open" or not. I did not know exactly where this article would go when we conceived it, and it went in a very different and much better direction from what I was expecting. Writing, as it turns out, is a lot like research.

Red Hat Collaboratory at Boston University seeks proposals for 2024

The Red Hat Collaboratory at Boston University has launched its annual Request for Proposals (RFP). Proposal submissions are due October 2, 2023, and awards will be announced by December 12, 2023. Awarded projects will have a start date of January 1, 2024.

The funding program enables collaborative research between Red Hat engineers and Boston University faculty and students, focusing on the hybrid cloud space on platforms ranging from edge devices to cloud datacenters. A fundamental goal of the collaboration is to develop techniques and best practices to integrate the rigor of academic research with the power of open source innovation. This is the third RFP cycle under the expanded partnership between Boston University and Red Hat. The Collaboratory funded $2.2M in funding to 19 projects through the 2023 RFP and $2.3M to 16 projects through the 2022 RFP.

RFP GUIDELINES

Projects must be open source and should generally focus on problems of distributed, operating, security, or network systems whose solution shows promise for advancing their field and impacting industry. Software developed during this research must be made available under an open source license, and all results will be publicly available.

Proposers are encouraged to review research areas of focus and existing strategic projects, available through the RFP page, in advance of developing a proposal to ensure alignment with the Collaboratory's research objectives. Details on projects awarded funding in the 2022 and 2023 calls and recent Red Hat Research Interest Group agendas are available on the Red Hat Research website (research.redhat.com).

The Collaboratory will fund projects at three levels: large (< $500K per year), small (< $175K), and speculative (< $100K). Large-scale projects are expected to be highly visible and engage a community of faculty, students, and Red Hat engineers. Small-scale projects will be more limited in scope and should include collaboration among a smaller group of faculty, students, and Red Hat engineers. Speculative projects may include fundamental systems research, work relevant to the Mass Open Cloud Alliance, and high-risk projects. Speculative projects do not need to have the committed involvement of a Red Hat engineer, but initiating university-industry collaboration through these projects remains a priority.

Projects have a timeline of one to two years, depending on scale, and may be eligible for renewal annually after the award period ends. We have streamlined application requirements for renewal proposals this year. The Collaboratory aims to encourage multi-year projects while ensuring that they continue to progress.

Teams of BU faculty and Red Hat technical experts will review applications, making selections based on feasibility, potential visibility, potential impact, novelty, contribution to diversity in Collaboratory research, and relevance to the Collaboratory and its infrastructure, all within the context of the proposed budget.

ⓘ

LEARN MORE

Researchers and engineers with questions or interest in the program should contact the Collaboratory team at prop-rhcollab-l@bu.edu as soon as possible.

NVIDIA

THE UNIVERSAL AI SYSTEM FOR HIGHER EDUCATION AND RESEARCH

NVIDIA DGX A100

Higher education and research institutions are the pioneers of innovation, entrusted to train future academics, faculty, and researchers on emerging technologies like AI, data analytics, scientific simulation, and visualization. These technologies require powerful compute infrastructure, enabling the fastest time to scientific exploration and insights. NVIDIA® DGX™ A100 unifies all workloads with top performance, simplifies infrastructure deployment, delivers cost savings, and equips the next generation with a powerful, state-of-the art GPU infrastructure.

Learn More About **DGX** @ nvda.ws/dgx-pod
Learn More About **DGX on OpenShift** @ nvda.ws/dgx-openshift

Hybrid cloud, edge, and security research featured at DevConf.CZ

After more than three years of strictly virtual meetings, DevConf.CZ has finally returned to in-person events. The Brno-based hybrid gathering is an annual, free, Red Hat sponsored community conference for developers, admins, DevOps engineers, testers, documentation writers, and other contributors to open source technologies.

Presentations highlighted progress made via industry-university collaboration in areas critical to future technology development, including hybrid cloud, edge computing, and security. More information on these projects, including contacts, opportunities to contribute, and access to slides, publications, repositories, and other artifacts, can be found on the Red Hat Research website. Recordings of most presentations at DevConf.CZ can be found on YouTube in the DevConf.CZ 2023 playlist.

HYBRID CLOUD

Optimizing Java on the EU processor platform, Christos Kotselidis (University of Manchester, UK; KTM Innovation) and Karm Michal Babacek (Red Hat)

This talk derives from the recently launched Accelerated EuRopean clOud (AERO) project. The University of Manchester and Red Hat are two of 12 consortium members participating in this EU-funded effort toward European sovereignty in the chip design and computer infrastructure industry. The presentation demonstrated two key components of the software stack regarding the Java ecosystem (video):

- Quarkus from Red Hat, a Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM's native-image (slides)
- TornadoVM from the University of Manchester, a JVM plugin for accelerating Java programs on GPUs and FPGAs (slides)

Design and deployment of FaaS apps at the edge-cloud, Yiannis Georgiou (CTO, Ryax Technologies)

PHYSICS is an EU-supported research project with the goal of unlocking the Function-as-a-Service (FaaS) paradigm for cloud service providers and application developers. Georgiou's talk focused on a design and development environment from the PHYSICS project aiming to ease application evolution to the new FaaS model. It uses the Node-RED open source tool as the primary function and workflow runtime to enable a more user-friendly and abstract function- and workflow-creation process for complex FaaS applications. To this end, it provides an extendable, pattern-enriched palette of ready-made, reusable functionalities such as workload parallelization, data collection at the edge, and function orchestration creation, among others. The environment embeds seamless DevOps processes for generating the deployable artifacts of the FaaS platform (Openwhisk). Annotation mechanisms are also available for the developer to dictate diverse execution options toward the deployment stacks, including sizing and locality considerations, as well as abilities for dynamic FaaS applications to continuously leverage the edge-cloud continuum. (video)

AERO
An open source cloud software ecosystem for the EPI hardware
Christ dis, Technical Coordinator
Associ ssor, The University of Manchester
Chief E KTM Innovation
Brno, 1 23
SUPERSONIC
SUBATOMIC
JAVA
QUARKUS

Karm Michal Babacek (left) and Christos Kotselidis present the AERO project at DevConf.CZ.

Some of the 1,100 in-person attendees who returned to the beautiful Faculty of Information Technology building at the Brno University of Technology for DevConf. CZ 2023.

Writing a K8s Operator for Knative functions, Luis Tomas Bolivar (Red Hat) and Jose Castillo Lema (Red Hat)

This well-attended workshop offered a hands-on opportunity to work with ideas and tools developed by PHYSICS project researchers for serverless and FaaS environments. One of the most relevant upstream projects for serverless is Knative, which recently added support for creating, building, and deploying functions on top of K8s clusters. Workshop participants implemented a K8s Operator, using the operatorsdk framework, to provide the functionality of the Knative CLI. (slides)

TRUST

Cybersecurity in the post-quantum era, Lukas Malina (Brno University of Technology, CZ)

This talk briefly introduced post-quantum cryptography and discussed the following issues: How do emerging quantum computers jeopardize current ICT security protocols (e.g., TLS, SSH, IPSec)? How long can we use existing asymmetric schemes such as RSA, ECDSA, or ECDH? Which quantum-resistant cryptography protocols are recommended by security authorities (e.g., NIST, NSA, BSI)? How shall we establish keys, encrypt data, and digitally sign messages after 2025? Are current security libraries ready for the post-quantum era? The talk reflects research done as a part of the Cybersecurity Excellence Hub in Estonia and South Moravia (CHESS), a 12-member consortium of universities, businesses, and government agencies addressing critical challenges in six areas: the internet of secure things, security certification, verification of trustworthy software, blockchain, post-quantum cryptography, and human-centric aspects of cybersecurity. (video)

Trust management in digital ecosystems, Dávid Halász (PhD Student, Masaryk University, CZ, Red Hat; Dávid spoke in place of Barbora Buhnova, who could not appear due to injury.)

Digitalization is leading us toward ecosystems where systems, processes, data, and things are not only interacting with each other but might start forming societies on their own (e.g., forming the Social Internet of Things). In these digital ecosystems, trust management on the level of system-to-system or thing-to-thing interaction becomes an essential ingredient to supervise the safe and secure progress of our digitalized future. This future-focused presentation, also stemming from CHESS, explored the essential elements that need to be considered for proper trust management in complex digital ecosystems and how trust-building can be leveraged to support people in safe interaction with other autonomous digital agents (e.g., self-driving cars, drones, and other robotic systems). (slides) (video)

User authentication in public open source repositories,
Agata Kružíková (PhD student, Masaryk University, CZ)

Kružíková maintained a booth where DevConf attendees could participate in her research project "Authentication in public open source repositories," which focuses on user behaviors.

UNIVERSITY CONNECTIONS

Teaching at the university: lessons learned.

Red Hat Panelists Alexandra Nikandrova (Technical Writer), Tomas Tomecek (Sr. Principal Software Engineer), Maria Svirikova (Interaction Designer), Sarka Jana Janderkova (Technical Writer), and Dávid Halász (Principal Software Engineer) shared their stories about introducing open source technologies and processes with university students. The discussion addressed challenges and opportunities that come with educating students on these topics, tips for effectively preparing for classes, the impact of this journey on the larger community and industry, and the benefits of open source technologies in shaping the future of education. The conversation offered valuable insights for anyone interested in open source technologies and how to share their knowledge. (video)

SUCCESS

DevConf.CZ 2023 witnessed an impressive turnout with over 1,100 in-person attendees and more than 5,000 views on live streams. The majority of attendees were from the Czech Republic, with participants from the United States, Germany, Austria, Slovakia, Serbia, Poland, and other countries. The conference featured an incredible lineup of speakers, with over 290 presenters delivering diverse sessions, including a keynote, talks, lightning talks, workshops, meetups, and activities.

Joel Savitz, a Red Hat software engineer based in Boston, described his experience:

> *" DevConf.CZ was an exciting conference: I met interesting developers from around the world, including people I'd only seen in meetings or emails. Hearing in-person talks on topics like secure supply chains and AI applications helped me understand projects and concepts much more quickly than if I were just reading documentation. Conferences like this help improve subsequent online interactions and strengthen open source communities.*

Planning for DevConf.CZ 2024 is underway—stay tuned for announcements later this year!

Funded by the European Union

*Projects **AERO** (Grant Agreement No.101092850), **CHESS** (Grant Agreement No.101087529), and **PHYSICS** (Grant Agreement No. 101017047) have been funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.*

Publication highlights

Red Hat Research collaborates with universities and government agencies to produce peer-reviewed publications that bring open source contributions along with them. These research artifacts illustrate the value that open industry-academia collaborations hold not just for participants, but for technological advancement across the field of computer engineering. This is a sampling of recent papers and conference presentations; to see more visit the publications page of the Red Hat Research website.

"**Afterlife: the post-research affect and effect of software**," Nicolas E. Gold (University College London, UK), Ian Lawson (Red Hat), Neil P. Oxtoby (University College London, UK). In (2023) *Research Ethics*.

"**An analysis of agile coaching competency among practitioners**," Leigh Griffin (Red Hat; South East Technological University, Westford, Ireland), Arjay Hinek (Red Hat). In (2023) *IFIP Advances in Information and Communication Technology* 668, pp. 30-37.

"**Co-developing hardware and software,**" Ulrich Drepper (Red Hat). Keynote presented at (2023) *35th Euromicro Conference on Real-Time Systems/ Operating Systems Platforms for Embedded Real-Time Applications* (Vienna, Austria).

"**Copy-on-pin: the missing piece for correct copy-on-write**," David Hildenbrand (Red Hat), M. Schulz (Technical University of Munich, Germany), Nadav Amit (VMWare). In (2023) *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (Vancouver, BC) 2, pp. 176-91.

"**Enabling VirtIO driver support on FPGAs**," Sahan Bandara (Boston University), Ahmed Sanaullah (Red Hat), Zaid Tahir (Boston University), Ulrich Drepper (Red Hat), Martin Herbordt (Boston University). In (2022) *IEEE/ACM International Workshop on Heterogeneous High-performance Reconfigurable Computing* (Dallas, TX), pp. 1–8.

"**An energy-efficient FaaS edge computing platform over IoT nodes: focus on consensus algorithm**," David Fernández Blanco (University of Lyon, France), Frederic Le Mouël (University of Lyon, France), Trista Lin (Stellantis), Julien Ponge (Red Hat). In (2023) *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing* (Tallinn, Estonia), pp. 661-670.

"**FAB: an FPGA-based accelerator for bootstrappable fully homomorphic encryption**," Rashmi Agrawal (Boston University), Leo DeCastro (MIT), Guowei Yang (Boston University), Chiraag Juvekar (Analog Devices), Rabia Yazicigil (Boston University), Anantha Chandrakasan (MIT), Vinod Vaikuntanathan (MIT), Ajay Joshi (Boston University). Forthcoming in (2023) *IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (Montreal, QC).

"**Hardware data re-organization engine for real-time systems**," Shahin Roozkhosh

(Boston University), Denis Hoornaert (Technical University of Munich, Germany), Renato Mancuso (Boston University), Manos Athanassoulis (Boston University). In (2022) *Proceedings of the WiP Session at the IEEE Real-Time Systems Symposium (RTSS@Work)* (Houston, TX).

"**On-the-fly data transformation in action,**" Ju Hyoung Mun (Boston University), Konstantinos Karatsenidis (Boston University), Tarikul Islam Papon (Boston University), Shahin Roozkhosh (Boston University), Denis Hoornaert (Technical University of Munich, Germany), Ahmed Sanaullah (Red Hat), Ulrich Drepper (Red Hat), Renato Mancuso (Boston University), Manos Athanassoulis (Boston University). Forthcoming in (2023) *Proceedings of the VLDB Endowment* 16:12.

"**Operating system noise in the Linux kernel**," Daniel Bristot de Oliveira (Red Hat), Daniel Casini (Sant'Anna School of Advanced Studies, Italy), Tommaso Cuinotta (Sant'Anna School of Advanced Studies, Italy). In (2023) *IEEE Transactions on Computers* 72:1, pp. 196-207.

"**Relational fabric: transparent data transformation**," Tarikul Islam Papon (Boston University), Ju Hyoung Mun (Boston University), Shahin Roozkhosh (Boston University), Denis Hoornaert (Technical University of Munich, Germany), Ahmed Sanaullah (Red Hat), Ulrich Drepper (Red Hat), Renato Mancuso (Boston University), Manos Athanassoulis (Boston University). 2023. In (2023) *Proceedings of the IEEE 39th International Conference on Data Engineering (ICDE)* (Anaheim, CA).

"**Relational memory: native in-memory accesses on rows and columns**," Shahin Roozkhosh (Boston University), Denis Hoornaert (Technical University of Munich, Germany), Ju Hyoung Mun (Boston University), Tarikul Islam Papon (Boston University), Ahmed Sanaullah (Red Hat), Ulrich Drepper (Red Hat), Renato Mancuso (Boston University), Manos Athanassoulis (Boston University). In (2023) *Proceedings of the 26th International Conference on Extending Database Technology (EDBT)* (Ioannina, Greece).

"**Scaling up performance of managed applications on NUMA systems**," Orion Papadakis (University of Manchester, UK), Andreas Andronikakis (University of Manchester, UK), Nikos Foutris (University of Manchester, UK), Michail Papadimitriou (University of Manchester, UK), Athansios Stratikopoulos (University of Manchester, UK), Foivos Zakkak (Red Hat), Polychronis Xekalakis (NVIDA), Christos Kotselidis (University of Manchester, UK). In (2023) *ACM SIGPLAN International Symposium on Memory Management* (Orlando, FL), pp. 1-14.

"**TeraHeap: reducing memory pressure in managed big data frameworks**," Iacovos G. Kolokasis (University of Crete, Greece), Giannos Evdorou (University of Crete, Greece), Shoaib Akram (Australian National University), Christos Kozanitis (ICS-FORTH, Greece), Anastasios Papagiannis (Isovalent, USA), Foivos Zakkak (Red Hat), Polyvios Pratikakis (University of Crete, Greece), Angelos Bilas (University of Crete, Greece). In (2023) *International Conference on Architectural Support for Programming Languages and Operating Systems* (Vancouver, BC) 3, pp. 694-709.

"**Unikernel Linux (UKL)**," Ali Raza (Boston University), Thomas Unger (Boston University), Matthew Boyd (MIT), Eric B. Munson (Boston University), Parul Sohal (Boston University), Ulrich Drepper (Red Hat), Richard Jones (Red Hat), Daniel Bristot de Oliveira (Red Hat), Larry Woodman (Red Hat), Renato Mancuso (Boston University), Jonathan Appavoo (Boston University), Orran Krieger (Boston University). In (2023) *Proceedings of the 18th European Conference on Computer Systems (EuroSys)* (Rome, Italy) .

"**Visualizing anti-patterns in microservices at runtime: a systematic mapping study**," Garrett Parker (Baylor University, Texas), Samuel Kim (Baylor University, Texas), Abdullah Al Maruf (Baylor University, Texas), Tomas Cerny (Baylor University, Texas), Karel Frajtak (Czech Technical University—Prague), Pavel Tisnovsky (Red Hat), Davide Taibi (University of Oulu, Finland). In (2023) *IEEE Access* 11, pp. 4434-442.

"**Software-shaped platforms**," Renato Mancuso (Boston University), Shahin Roozkhosh (Boston University), Denis Hoornaert (Technical University of Munich, Germany), Ju Hyoung Mun (Boston University), Tarikul Islam Papon (Boston University), Manos Athanassoulis (Boston University). In (2023) *Proceedings of Cyber-Physical Systems and Internet of Things Week* (San Antonio, TX), pp. 185–91.

Research is an adventure

Putting theory to the test at the university and in the field

*An interview with **Miroslav Bureš***

*conducted by **Martin Ukrop***

Don't tell engineering professor Miroslav Bureš that software testing can't be exciting. As the System Testing IntelLigent Lab (STILL) lead at Czech Technical University in Prague (CTU), Bureš's work bridges the gap between abstract mathematics and mission-critical healthcare and defense systems. His research focuses on system testing and test automation methods to give people new tools to detect relevant defects more quickly and cheaply than they can today. He is interested in using IoT technology and artificial intelligence in systems that can help first responders in general and military medicine. Using sensor networks, electronic devices, and augmented reality, these systems can make first responders' work faster, more effective, and less risky.

RHRQ asked Martin Ukrop, Senior Research Program Manager based in Brno, CZ, to speak with Bureš about his interest in testing, connecting university research with industry needs, and building systems to save lives. Ukrop is a security specialist who facilitates industry-academia cooperation for Red Hat Research. He also teaches security, programming, and algorithms courses at Masaryk University.

Martin Ukrop: Your research is heavily focused on advanced testing techniques. What got you interested in this?

Miroslav Bureš: The original motivation came from industry, not from research. When I finished my PhD at CTU ages ago, I went into industry as a project manager at a bank. After I successfully finished one project, a top manager at the bank came to me and said, "Hey, we have a project for you. I just fired the test manager from this project. Try to do something with it– you likely cannot make it even worse." I didn't know much about software testing, but we managed somehow, and it was great fun. I kind of made a name in the bank, then other managers came and said, "Ok, we would like to do this kind of testing. Can you do it?" During these projects, I came to see that people in industry should be given better tools and better methods, so I got the idea to come back to my alma mater and investigate these methods to try to develop something more. I soon realized that there are a lot of methods out there, and the necessary work was collecting them and making some qualified opinion on what would work best for which cases, under which conditions, and so on. After a few years, we founded STILL, and I switched to full-time research in this field, trying to produce new and more effective methods to test software and electronics for industry practitioners.

Martin Ukrop: You are now a full-time academic, but you still have quite a lot of relations with industry. Where do you see the worth of keeping a foot in both worlds?

Miroslav Bureš: I don't believe those are two worlds. It should be just one world: the best academic outputs made useful for people and the best from industry inspiring the research. Sometimes we do theoretical stuff that will later translate to particular methods we believe will

About the Author
Martin Ukrop
is a Senior Research Program Manager with Red Hat Research focusing on security research and facilitating industry–academia cooperation in EMEA. He received his PhD in Computer and Information Systems Security from Masaryk University, Czech Republic, focusing on human aspects in computer security. He remains an active teacher as well as a life-long learner.

be used by industry. But when I recall the last two successful PhD projects I've supervised, both originated by talking to people in industry.

For instance, for a project in state machine testing, we met an automotive industry partner and showed them, "This is what is available on the research side." And the industry folks told us, "That's nice, but it's not a 100% match—not even a 70% match." These available formal methods were time-consuming to translate to their practical usage—they need to be tailored to their specific requirements. Fair enough. So after a few discussions with other industry partners, we made the PhD project tightly inspired by these practical needs.

You also need to be able to prove things when you experiment. The closer you are to industry, the more realistic the systems and the more accurate the proof of your models. So it's natural to cooperate very closely with industry to do something truly useful.

Martin Ukrop: Do you sometimes have the reverse situation, where you have an interesting academic piece that turns out to be useful in industry, or do you tend to go from industry to academia?

Miroslav Bureš: Sometimes you find something that originated on the research side, and you realize five, 10, or 20 years later that you can apply these things. And there are some bits and pieces of mathematics that are more theoretical, but after some time, for instance at the level of applied AI, the industry changes. Things are done differently, and what was theoretical becomes practical.

IoT devices enable monitoring vital signs of soldiers, medics, or first responders in the field.

Martin Ukrop: I loved the way that you disagreed with me about academia and industry being separate worlds. However, up until now, we've been describing the advantages of cooperation between academia and industry. Are there times when this cooperation is less advantageous?

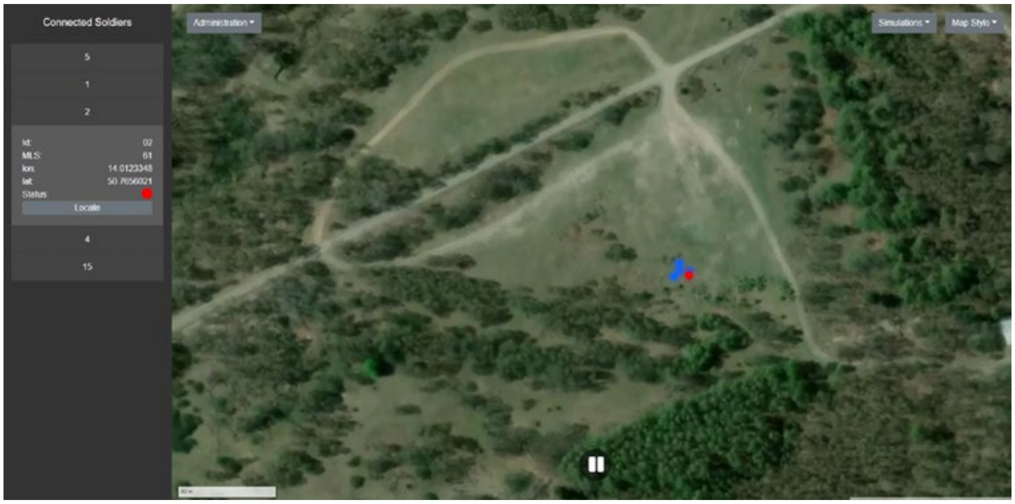
Miroslav Bureš: There is a concept called an industrial PhD, which I have seen several times successfully running in Scotland and Finland. I love this concept, but there are risks. Deadlines can be tough in industry. In research, you have long-term deadlines and long-term deliverables. In industry, deadlines are usually tighter, and there can be urgencies coming from daily operations: "We need to fix this," or "We need all hands on this release," and then you pause the research for a while. That can mean a one-month delay for the PhD student, and these delays might repeat.

In the opposite direction, students might be doing something very practical in the industrial context. It might be challenging to publish their work when most of the research community focuses on theory or primary research. High-level conferences or journals can be very competitive, so it takes time to focus and formulate your thoughts accurately. If you are working in industry, you're going to be interrupted by phone calls and questions—even a five-minute conversation can spoil your concentration.

It's a matter of KPIs (Key Performance Indicators), definitely. PhD students' KPIs are to publish good papers, present at conferences, and make a novel contribution to the research and industry communities. Industry tasks are usually more deadline-based and driven by daily operations. So it's about finding some balance.

Martin Ukrop: Red Hat is one of many companies you work with in research. How was the cooperation with Red Hat created?

Miroslav Bureš: We have a long tradition of cooperation since 2015.



5

1

2

Id: 02
MLS: 61
lon: 14.0123348
lat: 50.7656021
Status: ●

4

15

The location of a soldier with concerning vital signs can be visualized on a map.

Finding wounded soldiers in situations with poor visibility is facilitated by a wearable device.

One of the first major projects we did together (the PATRIOT project) was a common project funded by the Technical Agency of the Czech Republic to create a new test framework for IoT solutions. This was when the IoT market was starting, and the big hype was just beginning. So we asked the logical question: how to test these systems? Red Hat's part was to do a test automation framework based on an open development stack to automate integration tests in IoT systems. In the meantime, we did other work on the test automation framework Avocado. We created a new module for combinatorial interaction testing with an industrial use case study with Bestoun Ahmed at CTU (now Karlstad University) and Amador Pahim at Red Hat. It resulted in a good paper ("Toward an automated unified framework to run applications for combinatorial interaction testing") with Richard Kuhn from the US National Institute of Standards and Technology (NIST).

Martin Ukrop: Now Red Hat has a lab on the university grounds in space shared with the STILL lab, so Red Hat engineers are regularly at the university cooperating with faculty and students. Can you talk a little bit about how this works and how the university and the company benefit from this?

Miroslav Bureš: It is definitely an advantage that engineers can get directly in touch with the students and give them topics for theses inspired by systems from Red Hat, and it attracts students to the company. And the university will get the real data and sometimes funding, depending on the situation. This is an excellent model for how to partner—to treat the academic partner like a real partner, not a cheaper source of labor. Some other companies are trying to do this, but soon it turns out that it's not a sustainable model. And actually, the output of this more cooperative partnering style is much better for both sides than just the "we want the students" approach.

Martin Ukrop: At Red Hat Research Day last September (2022) in Brno, you mentioned other partners you cooperate with in your research. Can you talk about them?

Miroslav Bureš: In that talk, "Testing the reliability of systems with unstable or low-quality network connectivity" (available on YouTube), I discussed one of our projects in military medicine. It's highly experimental. We are creating a sensor network technology to help Combat Lifesavers (CLS) and field doctors do their jobs more safely and effectively. In this project, we cooperate with the University of Defense, Faculty of Military Health Sciences, and the

DefSec Innovation Hub, which is, practically speaking, the Czech branch of NATO's Allied Command Transformation Innovation Hub. This hub encourages and investigates new technology for defense. Then we cooperate with other academic partners, for instance, the Faculty of Biomedical Engineering of the CTU. Those are the people doing the hardware part of the project.

Martin Ukrop: Can you give a couple of examples of the projects you are working on with them? What is the intersection of software testing and doctors or the army in the field?

Miroslav Bureš: I can give you an example of a project we are preparing that can be used by firefighters and CBRN (Chemical, Biological, Radiological, Nuclear) units in the army. This project integrates the bits from previous projects and brings some innovations. If you have a first responder going into the field, you need situational awareness. We started talking to people with autonomous drones—such a drone can start from a vehicle coming to the scene and recon it automatically, so the people in the vehicle can see on a tablet what's going on and prepare for the mission in advance. Otherwise, they must rely on just the textual description via some transmitter or phone.

There's also a second part to this, the body sensors. This technology is advancing rapidly: we already have smart textiles that are simply amazing. In 10 years, we will have really good gear that can collect data with almost medical accuracy so that you can monitor things related to a person's health. So you can learn how a firefighter feels during a mission or how seriously a soldier is wounded. Even civilians can be given the sensors to monitor them before first responders arrive. The possibilities are great.

Another component is augmented reality. Here, we are doing some experiments to give first responders or military units the capability to see the position and status of their members through smoke or heavy terrain. It gives the unit a much better chance to react correctly and have the situational awareness to respond.

Martin Ukrop: That sounds like a very interesting mix from formal methods underlying advanced testing techniques to drones and body sensors, and through that to saving human lives.

Miroslav Bureš: Yes, it starts with some basic mathematics for AI and the construction of the systems and ends with concrete gear. It's nice to be present for the whole lifecycle, end to end.

Martin Ukrop: Seeing the whole process, from the early design to the physical thing, is a very satisfying moment, especially in research, when one often does things that can take years and years until application and production.

Miroslav Bureš: Yes, this is not just hype—like AI hype or IoT hype. This is the actual capability to make dangerous work safer and more effective.

Martin Ukrop: Last question: where do you see yourself in the next 10 years? Are there any goals that you would like to achieve? What does the future hold?

A wearable body sensor provides critical health information and situational awareness for people in high-risk environments.

Miroslav Bureš: We focus on two streams. The first is a new subfield of system testing mathematics, which we call constrained path-based testing. We believe that this new style of system-testing mathematics will guide people through testing situations more effectively. The second thing is to go further with the medical and rescue missions research. We want to develop more capabilities and power for these systems. It's the future, from my view of things.

Ten years from now, honestly, I cannot say. Research is an adventure. When we find something more useful, we will go for it.

Martin Ukrop: Thank you for the interview. We wish you a very interesting and engaging adventure in the upcoming 10 years! RH RQ

ⓘ

See also Miroslav Bureš's article "Testing critical IoT systems to mitigate network disruptions" in RHRQ February 2023 for more on this research.

MUNI FI **Masaryk University Faculty of Informatics**

Your research,

projects

and education partner.

RH
RQ
Bringing great research ideas into open source communities
Barbora Buhnová
+
Constant-time cryptography
The elastic bare metal cloud
Linux-based unikernels
Machine learning & accessibility
Red Hat Research Quarterly

RH
RQ
Bringing great research ideas into open source communities
Machine learning meets big data
Finding bugs in parallel programs
+
Mental models
Václav Matyáš
A thread model:
Red Hat Research Quarterly

FI.MUNI.CZ

Clouds that compete can't connect.
Says who?

A AWS

B Azure

C Google Cloud

[✓ D All of the above]

/Keep your options open
redhat.com/options

Feature

About the Author

Richard Jones has been using Linux since the early 1990s, joining Red Hat in 2007. Richard is now a Senior Principal Software Engineer in Red Hat's R&D Platform team.

Unikernel Linux (UKL) moves forward

by Richard Jones

RHRQ first looked at the Unikernel Linux (UKL) project—a joint effort involving professors, PhD students, and engineers at the Boston University-based Red Hat Collaboratory—almost two years ago (RHRQ 3:3, November 2021). This previous article covered the background of unikernels in detail, but in brief: an application links directly to a specialized kernel, a lightly modified version of Linux in this case, so that the resulting program can boot and run on its own. Unikernels have demonstrated significant advantages in boot time, security, resource utilization, and I/O performance. They enable those advantages by linking the application and kernel together in the same address space.

> Unikernels have demonstrated significant advantages in boot time, security, resource utilization, and I/O performance.

UKL's focus to date has been on minimizing changes both to the Linux kernel and to applications. By reusing Linux, we gain the advantages of Linux for free, especially wide driver support. We also studied the performance and latency characteristics of the final unikernels to see if making small, targeted changes could provide benefits.

The significant progress made by this project was detailed at the Eighteenth European Conference on Computer Systems (EuroSys '23), May 8–12, 2023, Rome, Italy, and published in the conference's proceedings. Here are some of the highlights.

PROJECT EVOLUTION

The Unikernel Linux (UKL) project started as an effort to exploit Linux's configurability to create a new unikernel in a fashion that would avoid forking the kernel. A unikernel taking this approach could support a wide range of Linux applications and hardware while becoming a standard part of the ongoing investment by the Linux community. Our experience has led us to a more general goal: creating a kernel that can be configured to span the spectrum between a general-purpose operating system, amenable to a large class of applications, and a highly optimized, possibly application- and hardware-specialized, unikernel.

Work to date has demonstrated that we can integrate unikernel techniques into a general-purpose operating system in a way that avoids forking it. It has also demonstrated performance gains. We think that most applications would run under these techniques at parity or slightly faster with no changes. With relatively little effort, targeted changes to the kernel can achieve significant gains.

A SPECTRUM OF CAPABILITIES

If we enable a base model UKL configuration (requiring 550 lines of code changes to

Linux) in the kernel, we're starting at the general purpose end of the spectrum. This simplest configuration of UKL supports most applications, albeit with only modest (5%) performance advantages.

Like many unikernels, UKL is a single application that is statically linked with the kernel and executed in supervisor mode. However, the base model of UKL preserves most of the capabilities of Linux, including a separate pageable application portion of the address space and a pinned kernel portion, distinct execution modes for application and kernel code, and the ability to run multiple processes. The main changes are that system calls are replaced by function calls and application code is linked with kernel code and executes in kernel mode.

As a result, this base model provides an avenue toward supporting all hardware and applications of the original kernel and the entire Linux ecosystem of tools for deployment, debugging, and performance tuning—which has been very useful in the course of this research. It also allows a developer to run "perf" directly inside the unikernel to collect performance information and feed that back into changes they make to the application to improve performance.

For more effort but with potentially more gain, a developer can move along the spectrum toward a specialized unikernel. A larger set of configuration options (1,250 lines of code changes total) may improve performance but will not work for all applications. Once an application is running, a developer can easily explore a number of configuration options that, while not safe for all applications, may be safe and offer performance advantages for their application.

One configuration bypasses the entry/exit code, which usually executes whenever control transitions between application and kernel through system calls, interrupts, and exceptions. Running the entry/exit code can get expensive for applications making many small kernel requests. The developer can also select between two UKL configurations that avoid stack switches, each appropriate for a different class of applications.

Knowledgeable developers can also (or alternatively) improve performance by modifying the application to call internal kernel routines and violating, in a controlled fashion, the standard assumptions and invariants of kernel versus application code. For example, they may be able to assert that only one thread is accessing a file descriptor and avoid costly locking operations.

To understand the implication of UKL's design for applications, we evaluated it with Redis, a widely used in-memory database. We saw two clear opportunities for performance improvement. First, we saw that we could shorten the execution path by bypassing the entry and exit code for read and write system calls and invoke the underlying functionality directly. We also observed that read and write calls eventually translate into `tcp_recvmsg` and `tcp_sendmsg`, respectively.

By taking advantage of these optimizations, researchers found that Redis throughput could be increased by up to 26% relative to standard Linux.

This led us to create a shortcut that enabled an application like Redis that always uses TCP to call the underlying routines directly. Only 10 lines of code were needed to implement this shortcut.

By taking advantage of these optimizations, researchers found that Redis throughput could be increased by up to 26% relative to standard Linux, whereas the UKL base model only improved throughput by 1.6%.

WHAT'S NEXT?

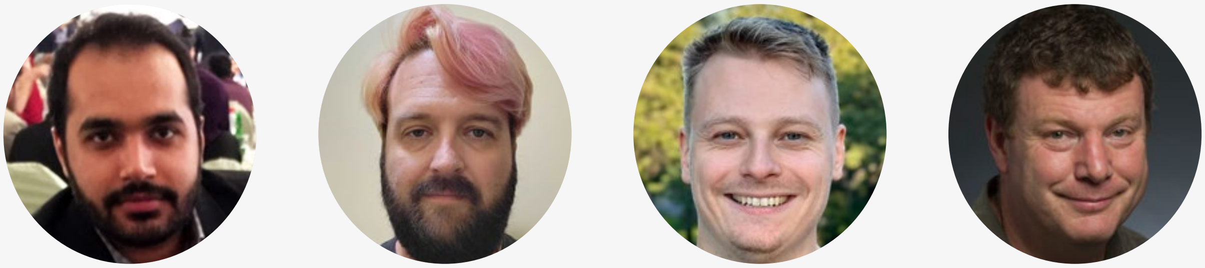
In addition to some cleanup work, such as rebasing to the latest kernel, glibc (which also requires code changes), and gcc, near-term work will focus on getting the project into the hands of more developers. The first step is adding the packages to the Fedora COPR service. The lengthy work of splitting up the Linux patches, authoring good commit messages, and checking that they pass Linux standards and tests is currently being done by Eric Munson at Boston University. After this is complete, we will submit them again to the Linux kernel community for comment.

The goal is, over time, to work with the community to add the changes to the Linux kernel as the current work is proven out and determined to be useful. In parallel with working with the kernel community, we need to demonstrate that the patches are useful for someone. To that end, we will work with other companies that have workloads requiring the highest performance and lowest latencies. We're currently looking for additional partners, both commercial and individuals, who would like to try out their applications with UKL. Most plain C/C++ applications with few dependencies that already work on Linux can be ported to UKL in an afternoon.

While we have been working on UKL since around 2018, other technologies occupying a similar space have come along, especially `io_uring` and eBPF. `io_uring` is interesting because it amortizes syscall overhead. eBPF is interesting because it's another way to run code in kernel space (albeit for a very limited definition of "code"). How do these approaches compare to UKL? We will be talking to developers who use these technologies to explore that question.

ABOUT THE RESEARCH TEAM

The Unikernel Linux project is the work of a sizeable team. Primary researchers at Boston University include PhD candidates Ali Raza, Eric Munson, Thomas Unger and Professor Orran Krieger, with additional support from PhD candidates Arlo Albelli, James Cadden, Matthew Boyd, and Parul Sohal, and Professors Renato Mancuso and Jonathan Appavoo. Red Hatters contributing to the project include Richard Jones, Ulrich Drepper, Larry Woodman, Daniel Bristot de Oliveira, Isaiah Stapleton, and Ryan Sullivan.



Boston University Researchers Ali Raza, Eric Munson, Thomas Unger, and Orran Krieger

To learn more, visit the Unikernel Linux project page on the Red Hat Research website. To see presentations and project artifacts, view the GitHub repository, or contact rjones@redhat.com.

intel AI
AI ON INTEL®
intel XEON® PLATINUM inside™
NOW BUILD THE AI YOU WANT ON THE CPU YOU KNOW.
Learn more at ai.intel.com

Feature

Generative AI and large language models: how did we get here, where are we going, and what does it mean for open source?

We may not have all the answers, but we're homing in on the essential questions about the future of AI and machine learning.

by Sanjay Arora and Richard Fontana

If you've somehow managed to escape the last nine months of breathless headlines and wild speculation about ChatGPT and what it means for humanity, you are lucky indeed. It's not as though machine learning, large language models (LLMs), and image generation are particularly new ideas, or even particularly revolutionary. However, the sudden availability of the ChatGPT "oracle" and the services competing with it have captured popular imagination on the scale of the moon landings, and not surprisingly. Even though an LLM is not much more than a linear regression over a big pile of data, it seems like a real intelligence. That makes it both interesting and scary for all kinds of reasons. When my retiree neighbors start asking me questions about AI, I think it indicates that a fundamental shift has happened.

There is no question that Generative AI—in short, a system that can generate text, images, or other media in response to prompts—is going to become both ubiquitous and required in academic research, in industry, and in teaching and learning. But its growing popularity raises important questions for software engineers and researchers, particularly those of us concerned with open source. How do large models come into being? Why now? Where are they likely to go next? Can a model be open source, freely modifiable, and redistributable by others? Does modifying an open source model, if such a thing exists, require the original data? All of it, or just some? If I use the output of a model in my code or my writing, is it still mine? If not, whose is it?

To answer these questions, we asked Red Hat Research's AI leader Sanjay Arora to help us understand how we got here and where, exactly, "here" is. We then asked Red Hat Legal's leading thinker on open source and licensing, Richard Fontana, to help us understand the relationship between the models behind Generative AI, open source licensing, and software development. Although they leave us with many unsettled questions, I believe they impart a solid understanding of what Generative AI really means for open source and IT, and what to look for in the future.

—Hugh Brock, Director, Red Hat Research

HOW WE GOT HERE

There has been impressive recent progress in machine learning models that generate realistic, high-quality text and images, including models like the GPT family for text generation and stable diffusion for image generation. While the public implementations of models like GPT have attracted plenty of hype and attention, tracing the historical development of ideas that led to them is instructive for understanding both their shortcomings and possible future developments.

Neural networks rose in prominence in 2012 after the image classification model AlexNet, a convolutional network, beat state-of-the-art benchmarks by a large margin on a standardized dataset called ImageNet.[1, 2] This model was trained using a combination of backpropagation and gradient descent,[3] both very old ideas. AlexNet was also one of the first implementations using a graphics processing unit (GPU) for speeding up forward and backward propagation. The main surprise here was that a multi-layered neural network could be trained by a first-order derivative-based local optimization technique like gradient descent, and that this could be done in a reasonable amount of time using GPUs.

This event led to an explosion of activity in multi-layered neural networks. Significant advances made training more stable—new activation functions, variants of gradient descent, various normalization and regularization strategies, architectural choices like skip connections, and so on—and enabled scaling to larger datasets using GPUs. The focus of deep learning practitioners switched from feature selection to devising better neural network architectures that would allow efficient end-to-end learning. The central idea was that given a labeled dataset of (input, output) pairs, one had to devise an architecture that could (a) operate on the inputs, (b) produce outputs of the right format (image, text, discrete labels, real-valued vectors), and (c) have enough weights or capacity to learn the mapping efficiently. This model could then be trained to learn the task of mapping the input to the output if such a mapping existed—in other words, if the outputs could be predicted from the inputs.

SELF-SUPERVISED LEARNING

The need for high-quality labeled data became a major bottleneck. Labeling data is generally a slow, laborious process that needs domain experts. Labeling tumors on radiological scans, which is very time-consuming, is a representative example. An elegant and very effective idea was resurrected to sidestep this requirement: use one part of the input data to predict another part of the input data that was masked or omitted from the actual input to the model. This is called self-supervised learning.

> The question is not *if* LLMs will give some software companies a competitive advantage in the marketplace, but when and in what areas.

[1] AlexNet. (2023, July 21). In Wikipedia. https://en.wikipedia.org/wiki/AlexNet

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90. https://doi.org/10.1145/3065386

[3] What is gradient descent? IBM.com. https://www.ibm.com/topics/gradient-descent

Examples include:

- Predicting which word occurs in a sentence based on neighboring words
- Predicting the next token in a sentence
- Splitting an image into a 3 x 3 grid and training a network to predict the relative ordering of any two patches from the image
- Converting an image to grayscale and predicting the RGB image from the grayscale image

These so-called auxiliary tasks don't need additional labeled data. Instead, the input data can be cleverly used to define a prediction task that a neural network can be trained to perform. The reason this is useful is particular to neural networks. Neural networks can be thought of as iterative maps of vectors to vectors. Here, a vector refers to a collection of numbers that can be added element-wise and multiplied by numbers (e.g., three-dimensional coordinate vectors in elementary geometry). This means that an input, such as an image or text, can be mapped to a vector that is an intermediate step for calculating the output.

These intermediate vectors, also called representations or embeddings, have a very interesting property: similar inputs get mapped to nearby vectors. Here, similar is a subjective notion. For example, we consider two images similar if they represent the same object, even if the angles, lighting, or background are different. Nearby, on the other hand, is a very concrete mathematical concept: two vectors are nearby if their difference vector has a short length. So we now have an explicit mathematical way of representing the subjective notion of similarity. To take advantage of this for self-supervised learning, you collect a large dataset, define a self-supervised task, and train a neural network on the task. You can then use this neural network to map each input object to its vector representation, which serves as a compressed numerical representation of the more complex input object.

Once a model has been trained on a self-supervised task, it has learned the underlying structure inherent in the inputs. For language, this includes the grammar and structure of sentences and paragraphs. For images, this structure is the joint probability distribution of pixels that describes realistic images. In other words, the representations learned by the model are now meaningful for other tasks. The most common way we harness these representations for specific tasks is through a process called fine-tuning. Fine-tuning involves taking the pretrained model (i.e., the model trained in a self-supervised way) and training it further on small supervised tasks. For example, a pretrained language model can be trained further on a small dataset of reviews and their sentiment—"positive" or "negative," say—to create a sentiment analysis model. The powerful representations learned by the pretrained network make training on small supervised datasets very effective. We call this semi-supervised training.

SCALING UP

As self-supervised training followed by fine-tuning demonstrated its effectiveness, another significant development occurred: the invention of a new architecture called the transformer. The overwhelming architectural choice for natural language (or sequential data, in general) used to be recurrent neural networks in their various avatars, like long short-term memory networks (LSTMs) or gated recurrent units (GRUs). They operated on the input tokens in a loop (i.e., sequentially), which is hard to parallelize. Transformers operate on all input tokens at once. To account for long-range dependencies and relationships between tokens, they use a mechanism called self-attention, which is a concrete way of scoring the relationship between any two tokens. Transformers impose fewer biases on the input data but can be scaled to much larger datasets given a fixed time budget. In recent years, transformers have been extended to other data modalities, especially computer vision (i.e., to images).

Combining transformers with semi-supervised training made it possible to train models on vast amounts of data. Based on past models, some scaling hypotheses were posited that estimated the amount of data, amount of compute, and model sizes required to achieve a certain model quality, where "quality" means test loss, or how accurately a model predicts on a hold-out dataset not used during training.[4] These efforts led to training large language models like the GPT series, LLaMa, and PaLM.

EFFICIENT FINE-TUNING

A few other innovations have been instrumental. The first is a technique

[4] Kaplan, J., McCandlish, S., Henighan, T., et al. (2020). Scaling laws for neural language models. ArXiv. https://doi.org/10.48550/arXiv.2001.08361

called LoRA (low-rank adaptation),[5] which addresses the challenge of fine-tuning large pretrained models. LoRA dramatically reduces the number of effective parameters that must be changed during fine-tuning. This makes it possible to fine-tune massive LLMs on small datasets and limited hardware in a reasonable amount of time.

Another major idea is instruction tuning. While one could take a pretrained network and fine-tune it on various tasks like question-answering or text summarization, this process would result in one fine-tuned network per task. Ideally, one would need just one network to perform all the tasks, with the task being passed as an additional input. For example, one could pass the task and the input together as "Summarize the following text: [text]" or "Answer the question: [question]." The downside of this approach is that the network can only perform tasks seen during training and is very sensitive to how the query is structured. Instruction tuning allows one to train a single network to perform multiple tasks by feeding the task description as text to the model.[6] As models got larger and had the capacity to perform multiple tasks, instruction tuning led not only to a model performing tasks seen during training but also to one that could generalize to new tasks outside the training set.

Yet another development is reinforcement learning from human feedback (RLHF). All language models are trained to predict probabilistic outputs; in other words, the output is a probability distribution over all possible tokens. The probabilities are then sampled to generate concrete tokens. This means that while the probabilities are fixed for a given input, the sampling process will generate a different output each time. Some of these outputs are qualitatively better than others. For a given query or input, a network's output is sampled several times to produce different outputs, which human testers then rank. Ideally, the network would produce outputs that are highly ranked. The sampled outputs and their human ranks are fed back to the network to encourage the network to produce highly ranked answers. This is done using the framework of reinforcement learning (on-policy, model-free methods like proximal policy optimization, for the experts), where the ranks act as rewards.

IMPLICATIONS FOR INDUSTRY

As generative models permeate industry and are adopted by professionals, their computational requirements and footprint will only rise. Even if training from scratch is limited to a few large companies, every institution using these models will have to invest in infrastructure for fine-tuning and inference. This infrastructure might be a third-party cloud service or on-premises hardware, but either will require a significant investment of time and resources.

There's interest within most if not all large companies in using these models, especially LLMs. While a lot can be done through simple fine-tuning, robust application of LLMs in an industrial setting requires evaluating and implementing a lot of tools to ensure correctness and verify produced results. General education and guidelines for the appropriate use of various techniques, while a significant investment, will also ensure that we use these tools effectively.

For the software industry specifically, LLMs provide ways to significantly improve software production, application performance, and customer service— in some cases, radically so. The question is not if LLMs will give some software companies a competitive advantage in the marketplace, but when and in what areas.

QUESTIONS REMAIN

Even with all these exciting developments, language models have several problems. Chief among them is that they often hallucinate outputs, meaning that the outputs can state "facts" that cannot be inferred from the training data. This has massive implications for using these models in any realistic applications. While one school of thought maintains that the only solution is using other methods (e.g., knowledge graphs, symbolic methods) in conjunction with language models, another school of thought believes these problems can be solved within the deep learning paradigm. Only time will be the judge of who is right. Another major problem with these models is their training cost, especially in terms of energy consumption. This problem can be

[5] Hu, E., Wallis, P., Allen-Zhu, Z., et al. (2022). LoRA: Low-rank adaptation of large language models. Proceedings of the International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2106.09685

[6] Bosma, M., & Wei, J. (2021, June 10). Introducing FLAN: More generalizable language models with instruction fine-tuning. Google Research. https://ai.googleblog.com/2021/10/introducing-flan-more-generalizable.html

approached at various levels, from more efficient hardware to better, more sample-efficient algorithms. There has also been promising progress in using more carefully curated data[7] to train much smaller models with performance similar to larger ones.

Aside from these practical questions around accuracy and cost, the very effectiveness of these models—the likelihood that people will actually use them en masse for things—means that we have to begin thinking about the legal and ethical issues relating to their use, particularly looking through an open source lens. Is it legal to use publicly available data for training? When, and to what extent? Should attribution or even compensation be required? Can models be considered copyrightable subject matter? If so, how should they be licensed? How should derivative, fine-tuned models be licensed? We examine some of these questions below.

DOES COPYRIGHT APPLY?

From a legal and ethical perspective, generative AI, including LLMs, has inspired many debates about copyright, licensing, and even the principles of open source. Right now, we are a long way from clear answers, and it's worth keeping in mind that a growing number of entities could lobby to influence the state of copyright law, and courts may hand down decisions that change the law in unexpected ways. While we wait for greater clarity, these are some critical questions to keep in mind. Note: most of what follows applies specifically to US law. The complexity of considering the treatment of this topic under a multiplicity of legal jurisdictions is generally beyond the scope of this article.

How will we determine limits on the use of training data?

Individual items of training data will in some cases be copyrighted, and those who assemble training data sets may in some cases have a copyright interest in the data set as a whole. Outside the US, some countries have additionally recognized sui generis database rights or other rights in non-creative data compilations. If training data is under copyright, the copyright owner can set limits or conditions on the freedom of others to make, distribute, or adapt copies. If you are training a model, you are necessarily making copies of the training data. Individual copyrightable data items (which could be, for example, some text, source code, or an image) might be covered by no license, or they may be covered by a license ranging from (a) one permitting essentially all uses with no conditions, to (b) relatively permissive licenses with limited conditions (open source, open content, and open data licenses are subsets of this category), to (c) relatively restrictive licenses (like proprietary software licenses).

This is not the end of the story, however. In the US, the fact-dependent doctrine of fair use allows copying of copyright-protected materials for certain purposes such as education, research, and journalism, while some other countries have begun to legislatively carve out exceptions to copyright law for activities like text and data mining and web scraping. These limits on copyright protection likely benefit activities involved in training models.

Can learning models themselves be copyrighted?

A deep learning model is specified by its architecture and parameters—its weights and biases. While courts have not yet addressed this issue, it seems unlikely that weights and biases can be subject to copyright protection, at least under current US law. In some circumstances, of course, a set of numbers may be copyrightable (for example, any digital encoding of some original, creative, and expressive content). But a model's parameters are not such an encoding.

Copyright only covers original and creative expression. Ideas, for example, are not copyrightable. In its landmark decision in *Feist Publications, Inc., v. Rural Telephone Service Co.* (1991), the US Supreme Court held that information by itself—like a collection of phone numbers—is not protectable under copyright.

What claim do creators have to their labor?

The court in *Feist* concluded that the effort to compile mere information, no matter how laborious, had no impact on copyright protection, explicitly rejecting the earlier "sweat-of-the-brow" doctrine. (There may be other countries where doctrines like "sweat-of-the-brow" are viable.) The court has been clear that copyright exists to promote knowledge and creative expression, not to reward labor or restrict the sharing of facts. This point has some resonance in the 2021 case of *Google LLC v Oracle America, Inc.*,

[7] Gunasekar, S., Zhang, Y., Aneja, J., & Mendes, C. (2023). Textbooks are all you need. ArXiv. https://doi.org/10.48550/arXiv.2306.11644

in which the Supreme Court held that Google's copying of the Java SE API, which included only those lines of code needed to allow programmers to create a new program, was a fair use of that material. In the trial phases of the case, Oracle placed some emphasis on the amount of effort and care that went into designing a complex API. An appeal to the value of labor may have emotional resonance for engineers who put in hours of work, but it should not properly have a bearing on the question of copyrightability. And, of course, if the weights and biases are not copyrightable, efforts to regulate use of the model parameters through purported copyright licensing should not be effective, though there may be alternative legal machinery for achieving such regulation.

How will open source licensing react to the rise of machine learning?

Open source licenses are primarily (though not entirely) low-friction forms of copyright licensing characterized by normative, customary limits on how restrictive the license conditions can be. One question that the open source and the machine learning practitioner communities are grappling with is whether existing open source licensing norms adequately address the issues introduced by generative AI. Open source licenses facilitate machine learning—above all because of the availability of powerful open source machine learning frameworks like PyTorch—but the open source licenses in use today were all developed before machine learning models became an issue of significant interest.

Several recent developments around AI have had an impact on broader ongoing debates over the proper meaning and scope of open source. Some organizations have been releasing machine learning artifacts, including model checkpoints, in public repositories on GitHub and HuggingFace, under restrictive licenses not compatible with the Open Source Definition (for example, licenses prohibiting commercial use or non-research use), yet describing such releases as "open source." The Open Source Initiative, which maintains the Open Source Definition, has raised concerns about "open-washing" in the industry.

At the same time, some machine learning practitioners have been promoting so-called Responsible AI Licenses (RAIL), which feature use restrictions aimed at preventing the use of AI for purposes regarded as unethical or at odds with certain social policy goals. These restrictions are particularly centered around the use of model weights, despite their dubious protectability under copyright. The various restrictions in the RAIL licenses prevent them from satisfying the OSD, but some RAIL advocates no doubt believe that the definition of open source itself should be changed or relaxed to accommodate such new regulatory models as applied to community-released model artifacts.

Another development arising out of the tension between open source licensing and machine learning stems from the widely-publicized tendency of certain generative models to replicate potentially copyrightable portions of training data, an issue that underlies a number of recent lawsuits brought against companies developing and commercializing generative AI technology. The specific area of source code generative tools came to public attention with the launch of GitHub Copilot. Some open source developers, particularly those using copyleft licenses like the GPL, have not only argued that such replications typically will not comply with the requirements of open source licenses but have also raised broader concerns about the use of their code in training data—even though any open source license should permit such use. These developers may find it appealing to add license prohibitions against use in machine learning, even though, as with RAIL, this would represent a departure from open source licensing norms. Some developers of generative AI programming assistant tools have responded to these concerns in various constructive ways, such as enabling authors to opt out of having their code used in training data and attempting to document the provenance and licensing of generated output.

Amid all this ambiguity, one thing that's clear is that we should be skeptical about making copyright licensing do more than it should. That's a point Luis Villa made in the podcast series "Was open source inevitable": "For years, we said the licenses were the only acceptable way to legislate behavior. . . . Maybe we wouldn't be having so many of these discussions today if we'd said that codes of conduct are also important and how we behave with each other as peers and friends and human beings." As the use of machine learning and generative AI expands, there's a risk that people will make assumptions about what is actually licensable and enter into agreements that are not enforceable. There may also be some activities—disclosing your training data, or at least information about your training data, for example—that become social expectations even if they are not legal requirements mandated by a license. The practice of publishing "model cards" and similar information seems to point in this direction.

Realizing the potential of Generative AI and LLMs described in the first half of this article will depend on open source communities, industry, and AI/ML researchers working together in the open. The more roadblocks we set up, the slower the progress.

About the Author
Sanjay Arora
works at Red Hat's AI Center of Excellence and is mainly interested in the application of machine learning to low-level systems.

About the Author
Richard Fontana
is Senior Commercial Counsel at Red Hat and founder of Red Hat's Technology and Open Source legal team. Before coming to Red Hat in 2008, Richard was counsel at the Software Freedom Law Center (SFLC) and served as one of the three principal authors of GPLv3. For several years, he was a board director for the Open Source Initiative and chaired its license review committee.

RESEARCH QUARTERLY

VOLUME 5:2

Feature

"Open source opens doors": mentoring students for success

Research- and leadership-focused support is getting results in the push to grow and diversify the engineering talent pool.

by Heidi Dempsey

About the Author

Heidi Picher Dempsey is the US Research Director for Red Hat. She seeks and cultivates research and open source projects with academic and commercial partners in operating systems, hybrid clouds, performance optimization, networking, security, and distributed system operations.

The technology industry has largely embraced the theory that diversity drives innovation, but in practice the talent pipeline continues to be leaky. Even when high school preparation is equal, students of color are more likely than white students to leave STEM majors, and half of women in tech leave the field by age 35 because they find the environment inhospitable. That's one reason Red Hat Research participates in the RAMP program (Research Academic and Mentoring Pathways to Success) at the University of Massachusetts at Lowell, an excellent model for how industry and academia can collaborate to accelerate change.

RAMP is a summer program for incoming UMass-Lowell students and select high school students interested in science and engineering. RAMP focuses on increasing the enrollment, retention, and success of groups underrepresented in engineering, but it's also the foundation of a network of faculty and mentors that connect students to people and opportunities that encourage them to keep going. From my experience, I know that participating in research with a mentor as an undergraduate can be transformative. Just discovering that it was possible sparked an interest that became my career at a time when my alma mater (MIT) was under scrutiny for being unfriendly to women in science.

I learned about the RAMP program while working with UMass-Lowell faculty on collaborative projects for Red Hat Research. The RAMP program, along with the SoarCS program for incoming computer science students, was just beginning, and both seemed like a perfect fit for our mandate to build a more robust talent pipeline for research through university relationships. The leader of the RAMP program, Kavitha Chandra, received her PhD from UMass-Lowell and is currently the Associate Dean for Undergraduate Affairs in the Francis College of Engineering. A professor of Electrical and Computer Engineering, she collaborates with industry partners like Red Hat and faculty from multiple disciplines to provide mentorship and leadership training for underrepresented students in engineering.

Several Red Hatters, alongside representatives from many other companies, have served as mentors in the RAMP program, including Rashid Khan, Senior Director of Networking Services,

Red Hat Senior Director of Networking Services and UMass-Lowell alumnus Rashid Khan

Kavitha Chandra, Associate Dean for Undergraduate Affairs in the Francis College of Engineering at UMass-Lowell

who also participates in research at UMass-Lowell and the Red Hat Collaboratory at Boston University. Khan got his BS in electrical engineering and mathematics from UMass-Lowell in 1996, and a Masters in Digital Signal Processing from Tufts University, and has stayed active as an alumnus through the university's Career and Co-op Center, coaching students on technical interviews, and by helping to create the UMass-Lowell open source lab.

EARLY ROLE MODELS

"What you see at a young age will inspire you toward what you want to be later in life," Professor Chandra said. Chandra watched her father work to attain his PhD and start his career while living in Alabama and raising a family and saw firsthand how vital mentorship is to success. Seeing him overcome the challenges he faced as a student and immigrant father of three young children with the benefit of consistent support motivated Chandra along her own academic path. It also spurred her desire to engage students from underrepresented minorities. "We were in the South in the seventies, and there were not many people who looked like us. My father's professor and his wife really made it their mission to help us out, and I saw that and was inspired by it."

Chandra took this lesson to building a mentorship program: get to students as young as you can. For many students, the path to a successful career as either an engineer or an academic is unfamiliar, making it easier to give up when doubts arise. And if students don't see people with similar backgrounds working in engineering fields in middle or high school, they may not even consider it a possibility. There's also a chain reaction of missed opportunities that can be limiting later. For example, even an excellent student will struggle for admission to competitive graduate programs without undergraduate research experience. "Once students start courses and internships, their lives are all about doing their jobs," Chandra said. "We need to reach them before that."

Khan also has an immigrant's perspective on the importance of mentorship, having moved to the US at the age of 19 to attend UMass-Lowell. "I have a soft spot in my heart for Lowell, not only because I went there, but because I know the value of that education for the many first-generation college students who go there," Khan said. Khan pointed out that this group of students brings in socioeconomic diversity as well as ethnic and gender diversity, which is critical to expanding access to education: "These are the students who can't afford an SAT course or an SAT retake, so while they may be smart and skilled they can miss opportunities just because of financial barriers."

Giving back is also a significant motive for Khan, who says Lowell gave him the confidence and education necessary to thrive in a challenging and profitable career. "It's not an elite private university, but the students and faculty are phenomenal. The friends I graduated with are CEOs, entrepreneurs, and leaders in banking, nuclear engineering, and so on. Lowell serves a lot of families who want to give their kids a great education but can't afford an elite college, and I want to contribute to that success story and help out those families."

BEYOND ACCESS

One philosophy behind the RAMP program is that access is necessary but not sufficient for increasing diversity. Chandra and Khan each emphasized the variety of barriers that students face. For example, young women often find that they are the only woman in a class of 30 or 40. Especially if they've been acculturated to certain gender-based roles, they may be hesitant to participate vocally. Chandra named not asking questions as one of the biggest barriers for students to overcome once they've gotten a seat at the table, especially if they want to pursue a leadership position. "You aren't questioning things, or, if you do, every question is prefaced with 'this question sounds stupid but…'" Chandra said. "Why do you have to use that? They don't see any man saying that kind of stuff. Just ask the question!"

Khan also noted that during his undergrad years, he watched the attrition of women from engineering programs in real time: "At the start of my program, there were a handful of women in electrical engineering, but by the time I graduated, there was only one. The side effect now is that, 25 years later, there are still not enough women in engineering. I realize now that if they had had a mentor to encourage them to persevere or an example of women in leadership, more of them might have stayed."

Kavitha Chandra (bottom row, center) and sociology professor Susan Tripathy (far left) pose with RAMP students.

Chandra and Khan both point to one-on-one mentorship and long-term relationships as the most critical factors in moving from access to inclusion. "Continuity is key. The only thing that works is that process of mentoring, where they're in your lab day in and day out, and you're talking to them about their struggles and helping them understand why they have to continue," Chandra said.

> You can't hire someone from an underrepresented background and consider the job done.

Chandra emphasized that the same is true in industry: "You can't hire someone from an underrepresented background and consider the job done. There have to be other elements, like connecting them with the right people who will recognize the barriers and the cultural issues they face and make them see that those obstacles are not their fault."

OPEN SOURCE RESEARCH

The most crucial goal of the RAMP program is to encourage students to be curious—so curious that they aren't just listening but are driven to ask questions and find ways to engage. Open source development and engineering research have proven an excellent way to provoke this response.

While discussing why open source is such a powerful tool for students, RAMP participants and students agreed on a crucial factor: open source opens doors. Khan explained, "All you need is a laptop and an internet connection, and it can open the whole world to you. I have had students contact me from remote parts of the world who are working on things that

Red Hat mentor Salvatore Daniele works with Chelmsford High School student Andrew Barber on a finite state machine programming project.

are next-next-gen. A young woman contacted me from a university in a developing country with a question about working on 100 gigabits a second, while we were working on 25 and 40 gigs. With proper guidance from teachers, she reached out to someone who could help her find her way through the open source community. This isn't possible with proprietary hardware or software."

Chandra added, "I'm dealing with students, especially young women, who are not looking at computer science as a field because they don't see themselves writing code alone in a corner. Open source gives them many ways to participate, to have an impact in a community, and to contribute."

And because those communities are inherently open, students have the opportunity to share their work and get full credit for their accomplishments. A student working for a defense contractor may not be able to publish their work at all, or they might have to conceal some of the details. By contrast, Khan said, "In open source, as soon as you work on something, it's public. You can publish a paper on it, make presentations about it, go to conferences about it, and your code is there forever. People will iterate on it, improve on it, but your name remains in the chain of contributors."

Those accomplishments have real consequences for students looking for work or applying to grad school, according to Chandra. "My goal is that students will be able to say something more than 'I did an internship.' Instead, they can say, 'I participated in this open source conference,' or 'I made a poster with the professional group I worked with'—they have real artifacts that help them stand out."

INDUSTRY ENGAGEMENT

When asked why connecting with engineers in industry is essential at an early stage, Chandra laughed. "I think from the students' perspective, university professors don't have real jobs. They look at the lives of professionals in industry as something real." Only a fraction of students will go on to academia, but at the same time, most students have a very limited notion of what an engineer does. Many have a stereotype that doesn't view engineering as a very interdisciplinary, inclusive field. They don't realize that engineers work in many different fields, including unexpected areas like the social sciences, education, and the humanities.

"A lot of students come in thinking that software engineers are stuck in a cubicle in front of a screen for 12 hours a day, six days a week," Khan confirmed. Working with industry shows them the practical implications, like creating the next generation of telecommunication, healthcare, defense, robotics, and automotive technology. "It surprises them to realize that software actually runs somewhere. I give them the example of social media, where you can flip through high-quality pictures one second at a time, and the next picture is always preloaded, and everything is cached for you. All of that is enabled by open source networking. When they get that and say, 'Oh yeah,' that is a beautiful moment."

Chandra said that when students understand what working as a developer or engineer really means, they get much more excited. "They want to know how future technologies and engineers can work in areas they didn't use to associate with computer science. If we can show them this vision, there will be a better, more diverse pipeline of students who want to come into this field. There's just so much need for more people with all different backgrounds and interests."

Khan reported that, from his perspective, the RAMP program has increased the ethnic and gender diversity of the student-to-employee pipeline. "Linux and Red Hat are not as widely known as a company like IBM outside academia. RAMP and our other student programs at UMass-Lowell help us introduce Red Hat and the concept of an open source software company, which is totally new to many of them." Students' work in industry also exposes them to a company's culture and what it's like to collaborate with some of the brightest minds in technology, whether that's at Red Hat or another company.

LAUNCHING THE NEXT GENERATION

Students in the 2023 RAMP program worked on projects related to the theme "Engineered systems and cybersecurity: from requirements to verification." Over the course of the six-week program, students complete 12 modules focused on the relevance of physics, math, computing, and human factors in engineering design, but most of their time is spent working on their projects and meeting with faculty and industry mentors. Our group of 10 Red Hat mentors met with students weekly to talk about technical subjects like operating systems and systems engineering, software vulnerabilities and hacking, and AI/ML and neural networks. We also talked about what's exciting and what's challenging about being an engineer, from coding to managing teamwork.

Red Hat mentors (from left) Michael Santana, Salvatore Daniele (mostly obscured), Rashid Khan, and Aaron Conole talk with Lowell High School student Jackie Tran and other RAMP participants.

At the end of the session, students presented their projects to peers and mentors, then revised and finalized them according to their feedback.

This year's cohort of students split into groups to complete three projects:

- **Modulation exploration:** analyzing the transmission of audio signals using higher, typically inaudible frequencies, understanding the mathematics of amplitude modulation, and using analog and digital systems to make measurements and validate the theory
- **Operation MANGO (Magnetic Alarm Network to Gate Outsiders):** developing defensive solutions after assembling an alarm system and exploiting a variety of attack vectors students developed to bypass it
- **Rise of the machines: the hidden cost of automation:** researching and testing the advantages and disadvantages of AI automation, especially its impact on the job market and the environment

Professor Chandra brought UMass-Lowell student Anna Schmidt, one of many undergraduates who have found a launching pad at RAMP, into our conversation. Anna started working with a Red Hat mentor during the pandemic lockdown, building widgets using a TI microcontroller system—a good fit for Anna's love of mechanical work. She also worked on learning Python but wasn't sold on coding as a full-time career. "What I want her to understand is that the way you code and the way open source development works is changing

Lawrence High School students Gwen Marcotte and Naira Sanchez practice the venerable art of whiteboarding for problem solving.

significantly," Chandra said. "You don't need to know the nitty-gritty of every language to be able to contribute."

In 2023, Anna interned at the healthcare tech company Abiomed, where her mentor noticed that when she first arrived, she was quite shy and had difficulty looking people at work in the eye. After six months, however, she was out front confidently presenting her work to Abiomed engineers. "They were so impressed with her," Chandra said. "Not just because of the work she did, but because of the confidence she gained. That's just one example of how we can make that sort of difference when people stay involved and connected with the program."

Chandra said that one of the best things students get from the program is the community they build in those six weeks of RAMP, which then sticks with them for the next four years or even longer. That's a critical way of providing continuity when resources are scarce. Chandra acknowledged that finding enough mentors can be tricky, especially among younger academics focused on building their careers. "If they came out of an environment where they needed and benefited from early, close mentoring, they see the value of it," Chandra said. Otherwise, dismissing the importance of mentorship programs like RAMP is too easy. "People will ask, 'Why do we need to do this?' and then ask, 'Why haven't things changed in 35 years?'" Chandra said.

One of Chandra's goals is to keep building a network of middle school teachers, high school teachers, graduate students, university faculty, and industry professionals who share a vision for mentoring students through the challenges faced by students underrepresented in science and engineering. In 2022 a team of researchers led by Chandra received two grants from the National Science Foundation to foster this work. "Connecting people with similar experiences is the key. Anyone interested in looking at these issues in a disruptive way and understanding the vision, come connect with us," Chandra said.

ⓘ

You can reach out to Professor Chandra (kavitha_chandra@uml.edu) or to me (hdempsey@redhat.com) to learn more about getting involved.

ACKNOWLEDGEMENTS

In addition to Kavitha Chandra and Rashid Khan, RAMP participants Michael Santana, Salvatore Daniele, Anna Schmidt, and Rahul Bajaj contributed their thoughts to this article. Other Red Hat associates who shared insights on everything from machine learning algorithms to Red Hat tattoos with students in the 2023 RAMP summer sessions included Aaron Conole, John Westcott, and Wen Liang. To the 2023 RAMP students in the Red Hat subgroup—Aryaman, Luca, Nischay, Jacqueline, Nell, Neel, Katrina, Andrew, Antonio, and Janie—thanks for a great summer!

UMass Lowell is proud to collaborate with Red Hat, a Select Preferred Partner, and celebrate more than a decade of working together on research, philanthropy and building the next generation of Red Hat’s workforce.

UMASS
LOWELL

Research project updates

Each quarter, RHRQ highlights new and ongoing research collaborations from around the world in one or more of our key areas of interest: AI and machine learning, hybrid cloud/research infrastructure, edge computing, and trust. This quarter we highlight collaborative projects with university partners at Boston University and the University of Massachusetts-Lowell. Contact academic@redhat.com for more information on any project described here, or explore research projects at research.redhat.com.

In the Red Hat Research group, we don't just write papers and technology roadmaps—we actually get to build systems and software to try out crazy and not-so-crazy new ideas. We work directly with students, professors, and practicing engineers from inside and outside Red Hat on proof-of-concept demos and prototypes that let us evolve ideas, try them out in real-world research environments, and get the honest feedback needed to decide if a project can successfully transition to the open source project ecosystem. Our worldwide graduate and undergraduate student interns make essential contributions to open source research projects throughout the year, but the summer is an especially busy time for US projects, when both students and researchers can roll up their sleeves and dive into one project for a few months. Here is a sampling of work in progress in a few highlighted research areas. We share many of the final summer project presentations in live and recorded Research Interest Group sessions that can be found on the events page of our website (research.redhat.com) and on the project and people pages. Come on in—the water's fine!

—Heidi Picher Dempsey, US Research Director for Red Hat

AI AND MACHINE LEARNING

Open source education (OPE) tools

Ke Li and Griffin Heyrich of Boston University are working with the Open Education (OPE) team at BU and Red Hat to improve tools for building and using dynamic textbooks based on Jupyter Notebooks and the Red Hat OpenShift Data Science Machine Learning platform. They are improving the way the team builds testing workflows and textbook images and working to make it easier to start and run environments for a live class of about 300 students. Danni Shi detailed the Open Education project in "Open source education: from philosophy to reality" (RHRQ May 2023).

ML pipelines and file-system-based vulnerability detection
Zhongshun Zhang of Boston University expanded work done as part of the AI for Cloud Ops research project by using the Praxi pipeline in the Mass Open Cloud to detect filesystem changes in an OpenShift cluster, tokenize package installation logs, and train an ML model to predict what packages are installed on a given system. He used RHODS and Kubeflow pipelines to evaluate multiple inference models. This tool will eventually have applications in vulnerability detection and validated software supply chains.

Fine-tuning LLMs for documentation retrieval and question answering
Christina Xu's work bridges AI, cloud, and edge research areas to determine whether ML models trained on large amounts of detailed technical networking documentation can provide useful answers to questions from engineers attempting to create and optimize edge networks. Christina, a BU graduate, is working to understand the capabilities and limitations of LLMs in this context, investigating techniques to reduce model size without decreasing accuracy and testing the results of her work with knowledgeable engineers. She hopes this work will eventually produce an open source tool to share with network designers and maintainers.

CLOUD/INFRASTRUCTURE

Podman machine improvements
Jake Correnti from UMass-Lowell works with the RHEL Platforms group to improve the Podman machine subcommands that manage Podman's virtual machine, so that users can run Linux containers on Windows and MacOS as well as on Linux. (Podman is a daemonless container engine for developing, managing, and running OCI containers and container images.) Jake, who is completing his second internship, also refactored the Podman machine for improved usability and supportability, resulting in a 40% decrease in function length.

A student enjoys study space with a view at the Boston University Center for Computing and Data Sciences, which houses the Red Hat Collaboratory.

EDGE COMPUTING

Linux kernel
Han Dong and Eric Munson from Boston University both explore different means of tuning the kernel for challenging edge environments. Han experiments with kernel tuning to maximize energy efficiency without compromising performance SLAs. He uses Bayesian optimization as an ML technique to dynamically adjust SLAs and energy goals for processing while supporting a real-world in-memory key-value store workload. Eric extends previous work on Unikernel Linux by investigating alternatives for kernel-side event loop handling that minimizes latency.

Another BU PhD student, **Arlo Albelli**, also works on kernel improvements, starting with work to port the kElevate system call to the ARM architecture, which is especially relevant for energy-efficient services. This work aims to allow a user process to dynamically request and relinquish hardware privileges over time, with the expectation that this mechanism will be valuable for energy and performance monitoring.