



Bringing great research ideas  
into open source communities

## Walter Scheirer

*Future vision: on the internet,  
technopanic, and the limits of AI*



Passive network monitoring  
with eBPF

QUBIP and the transition to  
post-quantum cryptography

Scaling data with  
anchored keys



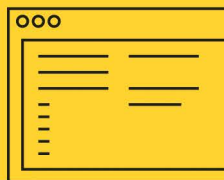
Red Hat  
Research Quarterly

Volume 5:4 | February 2024 | ISSN 2691-5278

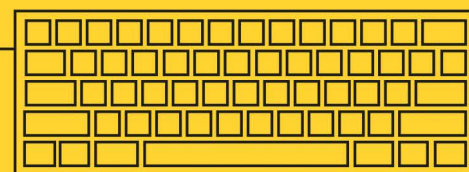
Your research,



projects

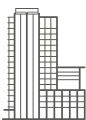
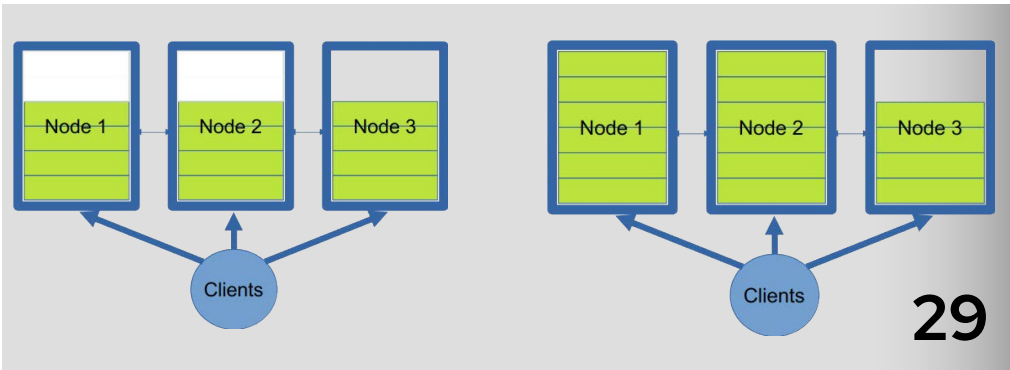
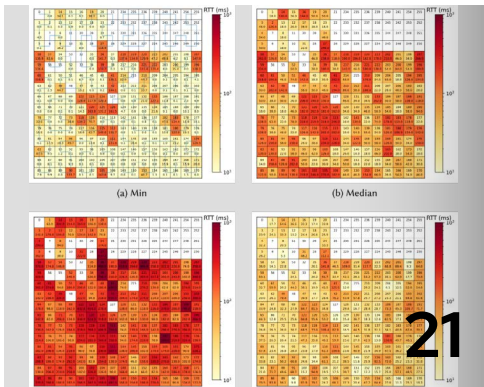


and education partner.





# Table of Contents



**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

## Departments

- 04 From the director
- 05 News: 2024 Collaboratory awards promote innovation in the cloud
- 34 Focus on AI and machine learning

## Features

- 10 Future vision: on the internet, technopanic, and the limits of AI— an interview with Walter Scheirer
- 21 Passive network monitoring with eBPF
- 26 QUBIP and the transition to post-quantum cryptography
- 29 Anchored keys: scaling of in-memory storage for serverless data analytics

## 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.


## Investing in open source research

by *Hugh Brock*

**T**his issue of RHRQ begins with a list that is the fruit of the labor of many people. It is a list of the winners of the 2024 Research Incubation Awards given by the Red Hat Collaboratory at Boston University, and it represents a major step forward in Red Hat funding. We've been through this award process three times, and although I've been happy with the outcome each year, I believe this one is the best by far. Some projects continue promising work we began in prior years, like testing for hypervisor vulnerabilities using fuzzing or the DISL project to streamline deploying working code to FPGAs, microcontrollers, and other devices. Some are brand new, like privacy-preserving ML training with differential privacy expert Adam Smith or Manuel Egele's new work on detecting deadlocks in binary-only kernel modules. What all this year's projects have in common is strong interest from Red Hat engineers and a path toward useful results. Whether those results are real code landed upstream or a well-grounded answer to a difficult technology question, they all support the mission of the Collaboratory: connecting great research ideas with upstream communities.

It is always surprising to me that I, an English and music major whose formal training in computer science was a junior high class in FORTRAN, find myself leading research efforts for a software company. So I'm pleased to offer you this issue's interview with Prof. Walter Scheirer of Notre Dame, a Red Hat Research collaborator who brings a strong background in the humanities to his work. A computer vision expert, Scheirer has made a name for himself as a technology ethicist with his book *A History of Fake Things on the Internet*, an

exploration of the history of "fake news" and how expectations around honesty and reliability online may be misplaced. "Don't believe everything you read" seems like a good maxim here, but maybe more important is the always-relevant principle, "There is no free lunch." If the "product" you're using is free, chances are that you—and the private data you're giving away—are the product. Scheirer talks about how we often ignore this principle, along with the difficult questions about what this data processing costs us, while we worry about a "fake news" problem that has always been part of life in a free-speech society.

It seems we have a lot to say about data, privacy, and security in every issue of RHRQ. This issue features two technical reviews of security and data management projects funded by EU Horizon grants. If you're not familiar with the European Union's Horizon program, [it is worth learning](#) about. The EU funds many interesting projects this way, and we've been fortunate to engage in a number of them. First up is QUBIP, an open source-focused project to develop encryption techniques that will resist quantum computing and make them ubiquitous in cryptography. Cloud Button, a second, now-completed, project, is about democratizing access to and use of large datasets by making it possible to work on them using a serverless in-memory architecture. This is very useful for both research computing needs like geospatial analysis and business requirements like diagnostics and predictive analysis. Projects like Cloud Button help make access to large datasets available to anyone interested, not just those lucky enough to have a "free service" to offer in exchange. 





## 2024 Collaboratory awards promote innovation in the cloud

Machine learning, energy efficiency, and secure data sharing are major research themes in Collaboratory-supported projects.

With the announcement of the 2024 Research Incubation Awards for the Red Hat Collaboratory at Boston University, the Red Hat-BU partnership has deepened its long-running commitment to advancing the state of the art in cloud computing, AI and machine learning, and operating systems. The Collaboratory seeks projects that will benefit from the unique combination of industry experience, academic rigor, and open source principles it was created to foster.

The 12 new and renewed research projects were chosen for their high potential to address real-world challenges in innovative ways. Security- and privacy-focused projects target risks associated with edge and cloud computing environments and seek to improve the performance of security operations centers. Energy efficiency continues to be a critical issue for systems research, as is making AI and machine learning (ML) sufficiently scalable to solve previously intractable problems.

Check out the list of this year's winners and watch RHRQ and the Red Hat Research website for ongoing updates.

**PROJECT: CoFHE: compiler for fully homomorphic encryption**

**PRINCIPAL INVESTIGATOR:** Ajay Joshi

**BU CO-INVESTIGATOR:** Rashmi Agrawal

**RED HAT INVESTIGATORS:** Aakanksha Duggal, Lily Sturmann

In the past decade, homomorphic encryption (HE) has emerged as a viable cryptographic solution that allows a cloud service provider to keep data in encrypted form while being processed. The project proposes CoFHE, a comprehensive FHE compiler framework to automate the process of generating the FHE implementations of the overlying applications using the CKKS FHE scheme. Investigators are targeting ML-based applications because ML



*Vasia Kalavri presents her work on high performance and energy efficiency in open source stream processing at the 2023 MOC Alliance Workshop.*

is pervasive in today's applications, and these applications are commonly run in the cloud, which is susceptible to data breaches.

**PROJECT: Co-Ops: collaborative open source and privacy-preserving training for edge and automotive AI**

**PRINCIPAL INVESTIGATOR:**  
Eshed Ohn-Bar

**BU CO-INVESTIGATORS:** Adam Smith, Venkatesh Saligrama

**RED HAT INVESTIGATORS:**  
Sanjay Arora, Michael Clifford, Erik Erlandson, Lance Galletti, Ilya Kolchinsky

While collaborative development and training of large AI models can drastically accelerate scalable edge applications, from open source mapping to safe autonomous driving, they are currently hindered by limitations in performance, efficiency, and privacy. This project develops generalized open source tools to address these limitations through novel mechanisms for enabling distributed data collection, protection, aggregation, and processing across potentially millions of heterogeneous platforms continuously streaming diverse data. One longer-term goal is addressing the lack of scalability in the development of edge applications and autonomous vehicles, with data

collection and model training in areas often neglected by mapping and transportation companies (e.g., rural areas, bad roads, and lower socioeconomic settings).

**PROJECT: Discovering opportunities for optimizing OpenShift energy consumption**

**PRINCIPAL INVESTIGATOR:**  
Jonathan Appavoo

**BU CO-INVESTIGATOR:** Han Dong

**RED HAT INVESTIGATORS:**  
Sanjay Arora, Huamin Chen, Heidi Dempsey, Parul Singh

This project aims to systematically discover opportunities for optimizing energy efficiency within the OpenShift orchestration platform. The investigators plan to harness the collective experience and expertise of distinct teams, uniting the capabilities of the Red Hat Kepler and PEAKS projects with ongoing endeavors in energy efficiency systems research.

**PROJECT: DISL: a dynamic infrastructure services layer for reconfigurable hardware**

**PRINCIPAL INVESTIGATOR:**  
Martin Herbordt

**BU CO-INVESTIGATOR:** Mayank Varia

**RED HAT INVESTIGATORS:** Ulrich Drepper, Ahmed Sanaullah

DISL is an abstraction layer for FPGA hardware operating system generation that enables software

developers to build custom and portable system stacks without having hardware development expertise. A critical enablement aspect for DISL is its component library of DISL-compatible hardware IP blocks, which are designed to support the required flexibility and portability. This project targets two fundamental subsystems of the stack: the interfaces to host and network. The goals are to implement and demonstrate that flexibility can be achieved in a portable fashion for these components and show the benefits of emergent customizability.

## **PROJECT: HySe: hypervisor security through component-wise fuzzing**

**PRINCIPAL INVESTIGATOR:**  
Manuel Egele

**RED HAT INVESTIGATOR:**  
Bandan Das

The security of the entire cloud ecosystem crucially depends on the isolation guarantees that hypervisors provide between guest VMs and the host system. The fact that the interfaces between the hypervisor and the host are manifold complicates these isolation guarantees. While there are well-known interfaces, such as those that virtual devices expose to the kernels running inside a guest VM, these interfaces also comprise functionality not necessarily triggered during “normal” operation of a VM. Investigators propose HySe, an approach to tackling the challenges imposed by a broader set of interfaces that

hypervisors expose to guest VMs and users of cloud deployments.

## **PROJECT: Improving cybersecurity operations using knowledge graphs**

**PRINCIPAL INVESTIGATOR:**  
David Starobinski

**RED HAT INVESTIGATOR:**  
David Sastre Medina

This project aims to improve cybersecurity operations, including automating several tasks, by synthesizing the vast amount of structured and unstructured real-world data available on threats, attacks, and mitigations. Investigators will apply Knowledge Graphs for cybersecurity purposes such as uncovering hidden relationships, identifying patterns and trends, and querying the data. A specific goal of the project is to assist software developers in identifying and patching vulnerabilities along various stages of the software development life cycle. Investigators also plan to evaluate and demonstrate the capabilities of the NERC and Red Hat OpenShift AI to support building LLM-augmented Threat Knowledge Graphs.

## **PROJECT: Lock ‘n Load: deadlock detection in binary-only kernel modules**

**PRINCIPAL INVESTIGATOR:**  
Manuel Egele

Two shortcomings in the Linux kernel/module security analysis landscape motivate this research. First, existing security analyses focus mainly on detecting memory corruption bugs and largely eschew availability bugs,

such as those induced by deadlocks. Second, the single most popular detection approach for deadlocks, Lockdep, requires the source code for the kernel and any kernel objects (KOs) and hence cannot be applied to binary-only KOs. This research project aims to develop novel capabilities that bring Lockdep’s detection mechanisms to bear on closed-source binary-only Linux kernel modules. Lock ‘n Load will provide tools that allow detection of deadlock-related bugs in binary-only KOs and enable device vendors to address threats that compromise the availability of their products.

## **PROJECT: Minimal mobile systems via cloud-based adaptive task processing**

**PRINCIPAL INVESTIGATOR:**  
Eshed Ohn-Bar

**BU CO-INVESTIGATOR:**  
Renato Mancuso

**RED HAT INVESTIGATORS:** Sanjay Arora, Jason Schlessman

Efficient cloud compute can facilitate AI model training at scale. However, due to suboptimal performance and ad-hoc integration by current cloud-edge frameworks, an edge device requiring real-time performance does not generally employ the cloud during inference. Investigators seek to develop a new paradigm and a generalized OpenShift-based tool for enabling optimal and highly dynamic integration between cloud and edge inference across diverse tasks and settings, for example, in complex environments where even a slight delay can be problematic,



such as navigation in crowded regions with multiple optimization and scheduling tasks. By enabling seamless offloading and routing between edge and cloud decisions, this adaptive framework can efficiently enable diverse real-time applications.

**PROJECT: Optimizing kernel paths for performance and energy**

**PRINCIPAL INVESTIGATOR:**  
Jonathan Appavoo

**BU CO-INVESTIGATOR:** Han Dong

**RED HAT INVESTIGATOR:**  
Larry Woodman

The growing size of modern OSES such as Linux is well documented and likely exacerbated as more features are packed into hardware. To address these challenges, a large body of work in application-specific OSES and optimizations has been developed for accelerating network applications. This project aims to conduct data-driven optimizations of the Linux kernel to advance the community's understanding of the plausibility of these techniques and their impact on performance and energy efficiency.

**PROJECT: Practical programming of FPGAs with open source tools**

**PRINCIPAL INVESTIGATOR:**  
Martin Herbordt

**RED HAT INVESTIGATORS:** Sanjay Arora, Ulrich Drepper, Ahmed Sanaullah

The problem of creating computer programs that are portable, performant, and require minimal

effort to program, port, optimize, and so forth is one of the most long-standing in computer engineering. But the logjam has broken as several trends have combined to enable rapid progress: new ML algorithms, increasing compute capability, and availability of training data. A primary focus is addressing high-level synthesis (HLS), especially for FPGAs. These devices are being widely deployed in datacenters, the edge, and IoT, but remain nearly impossible for ordinary coders to program, leaving the potential benefit of these deployments untapped. This project focuses on ML methods of optimizing compilers to perform high-quality HLS.

**PROJECT: Symbiotes: a new step in Linux's evolution**

**PRINCIPAL INVESTIGATOR:**  
Jonathan Appavoo

Linux's ability to evolve has proved invaluable in enabling everything from datacenter-scale cloud computing to wearable smart devices. However, UNIX enforces a strict boundary between what constitutes the core, or kernel, of the running operating systems and the applications programs. While this boundary ensures that programs cannot corrupt other programs, it also makes it very difficult to write applications that can directly use any part of the hardware or integrate OS kernel functionality. This work explores how a new kind of software entity, a symbiote, might bridge this gap. With the ability to shed this boundary, application software is free to integrate, modify, and evolve into a hybrid that is both application and OS.


**PROJECT: Towards high performance and energy efficiency in open source stream processing**

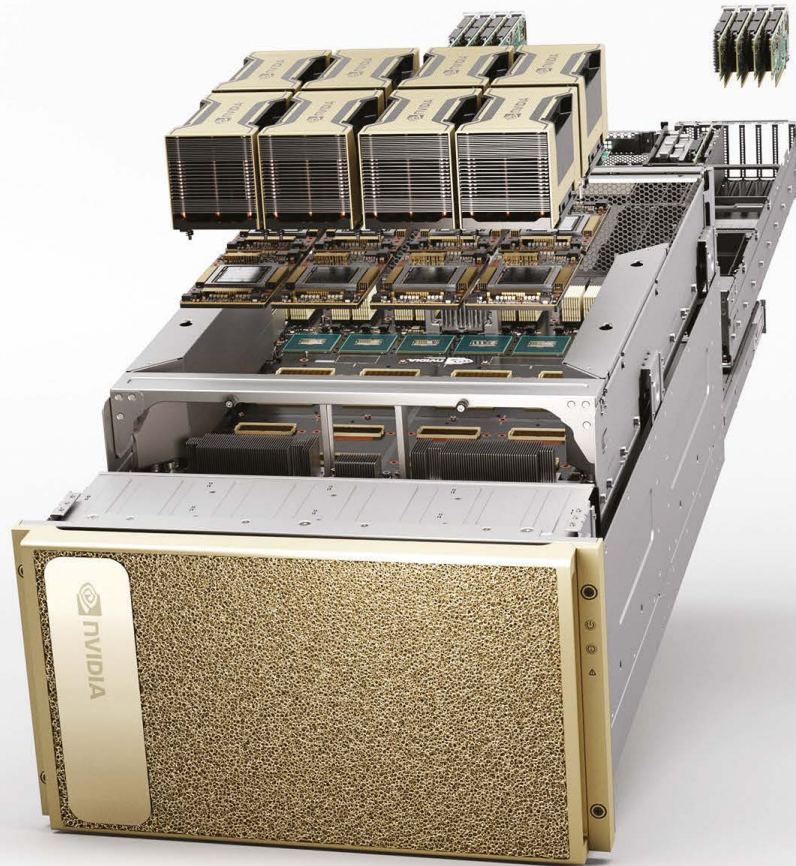
**PRINCIPAL INVESTIGATOR:**  
Vasia Kalavri

**BU CO-INVESTIGATORS:**  
Jonathan Appavoo, Han Dong

**RED HAT INVESTIGATOR:** Sanjay Arora

Continuous data streams generated by phones, cars, houses, smart cities, and electronic purchases feed information into cloud-hosted business analytics and prediction models. Cloud computing research has focused on optimizing the performance of such applications, but little effort has been devoted to understanding and improving their energy efficiency. This project aims to demonstrate that energy efficiency and the myriad layers of software that go into an open source streaming platform need not be incompatible. Researchers will leverage the open nature of the Apache Flink software to build a platform that optimizes trade-offs between energy efficiency and performance while maintaining transparency and the easy sharing of knowledge.

All software developed by these projects will be available under an open source license, and all results will be publicly available. Red Hat Research regularly provides on Collaboratory progress and announces engagement opportunities through [Red Hat Research Quarterly](#), its searchable [project database](#), [blog](#), [newsfeed](#), and live in-person and virtual [events](#). Contact [Jen Stacy](#), Senior Project Manager with Red Hat Research, for more information. 



# THE UNIVERSAL AI SYSTEM FOR HIGHER EDUCATION AND RESEARCH

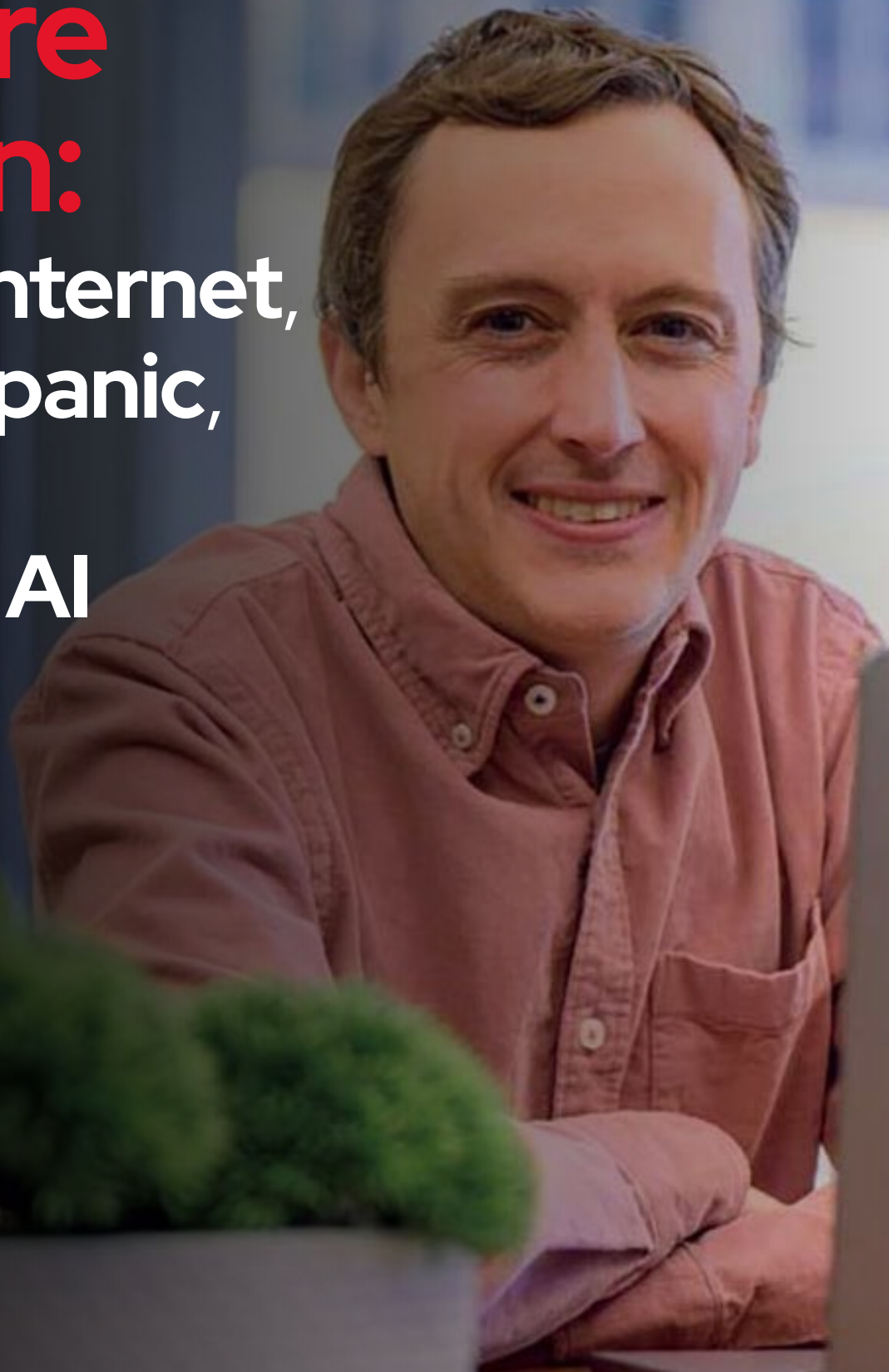
## NVIDIA DGX A100

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** @ [nvidia.com/dgx-pod](https://nvidia.com/dgx-pod)

Learn More About **DGX on OpenShift** @ [nvidia.com/dgx-openshift](https://nvidia.com/dgx-openshift)

# Future vision: on the internet, technopanic, and the limits of AI



---

*An interview with **Walter Scheirer**  
conducted by **Jason Schlessman***



## Interview

**E**veryone has an opinion on misinformation and AI these days, but few are as qualified to share it as computer vision expert and technology ethicist Walter Scheirer. Scheirer is the Dennis O. Doughty Collegiate Associate Professor of Computer Science and Engineering at the University of Notre Dame and a faculty affiliate of Notre Dame's Technology Ethics Center. In December 2023, he published [A History of Fake Things on the Internet](#) (Stanford UP), an exploration of the history of "fake news" and the technical advances that make new forms of deception possible. Professor Scheirer is also the longtime friend and research partner of Red Hat engineer Jason Schlessman. Together, they worked on the Red Hat Research-supported project "[Disinformation detection at scale](#)," which used image forensics and machine learning tools to develop a toolkit for identifying image manipulation in a scalable way. Schlessman interviewed Professor Scheirer for RHRQ about his work in computing, where ethics come into play, and why AI and the internet aren't the bad guys.

**Jason Schlessman:** I want to start with technology ethics, which is highly relevant in our industry. Could you talk about your work in that space?

**Walter Scheirer:** I think we can agree that the use of technology has generated all sorts of dilemmas. For example, the internet has become pervasive and opened up a lot of interesting avenues, but it's opened up some dangerous ones as well. That drives a lot of the discussion around fake stuff on the internet, which is the major topic of my book. Some of that fake stuff is problematic. But the trouble I see in technology ethics is everybody just wants to talk about it through a partisan political lens, and you have to think more broadly. It's a much broader topic than just politics.

**Jason Schlessman:** Historically, it seems like the focus on ethical usage has had to do with data. Would you agree?

**Walter Scheirer:** That's a big driving issue. There are lots of legitimate things to be concerned

with: Who's collecting this data? What is the harvesting process? How is that data being used? Is it being sold without the user knowing?

**Jason Schlessman:** As you point out in your book, the ethical use of data was not as heavily focused on until around 2016, but people in our industry knew data was being mined and used well before that. In the past year or two, it seems there's been a push to focus on AI as the ethical problem.

**Walter Scheirer:** These things are not mutually exclusive, especially when we think about contemporary AI. Machine learning requires data. We're moving to this paradigm of programming with data, solving problems we can't solve with procedural programming. But where did the data come from? Do the users know their stuff is ending up in chat GPT or whatever big model is out there?

**Jason Schlessman:** That's a good point, but I'm wondering whether people are just focused on AI as a concept, so these important issues are getting overshadowed by a general panic.



### About the Author **Jason Schlessman**

is a Principal Software Engineer at Red Hat Research, focused on novel AI and machine learning innovations that lead to pragmatic and feasible solutions. He especially targets projects that serve the well-being of humanity, fostering ethical uses of technology.

**Walter Scheirer:** Sure, one way to read this is that the panic is a coordinated strategy by a handful of powerful tech companies to distract people from real problems. That's a narrative out there in the news, whether or not it's true. There's a huge degree of information asymmetry: it requires a lot of technical knowledge to understand how AI works these days, and an average user is not going to have that level of understanding. That's a problem.

A lot of companies rely on data harvesting to make money, and if that's your business, of course you want to protect it. But is that in the public's best interest? These are big questions serious people in technology ethics are asking.

**Jason Schlessman:** You mentioned the problem of political partisanship in technology, but does legality get into this, with regulations and legislation?

**Walter Scheirer:** I get this every time I do an interview for my book: what do we do about regulations? But when we're talking about fake things on the internet, regulations are not feasible. That may seem surprising, but think about what this really is asking for. It's asking for control of speech on the internet. In the United States, specifically, there are strong guarantees for freedom of speech. I don't see a feasible path, going through the government. There's talk of antitrust maneuvers against big platforms, and that's probably more feasible because those regulations already exist.

**Jason Schlessman:** What about openness? Is there talk about

the openness of generative AI and LLM models, for example, with respect to provenance and stewardship? I would think ethically, it just makes sense to have things as transparent as possible.

**Walter Scheirer:** There is a vigorous debate right now about the open source aspect of this. Some companies are very much in favor of openness. For instance, Meta has been vocal about the need for open foundation models. The [AI Alliance](#), which Notre Dame and Red Hat are part of, along with Meta and several other businesses and universities, is also trying to move towards a more open source environment. Others don't want to release as much, and they argue that these things are their intellectual property. There will be some interesting court cases in the near future to try to sort this out.

**Jason Schlessman:** What made things click for me with open source was understanding how important observability and traceability can be. If you can rebuild an entire application, you can know at any given point what's happening in that application. It's the same with open source models: I could make this model if I had money to train it.

**Walter Scheirer:** That's a good point too—the scale of hardware required. Even if you had all the pieces, you can't, in many cases, actually replicate the setup. Some organizations are trying to address this, like [EleutherAI](#). On the other hand, for example in the computer vision community, there's an emphasis on small models. How far

can you get using something you can train on a single GPU? There's a lot of interest in that for low-power, smaller-compute applications. There's this myth that you need enormous cloud infrastructure to train one foundation model to solve these problems. I'm seeing a growing number of papers showing that's not the case. In some cases, you only need a fraction of those parameters to get the job done, and it's gratuitous to go beyond that.

**Jason Schlessman:** Does your work in ethics look into this idea of carbon footprint? Is it ethical to impact the environment even further by having my enormous datacenter churning for months to make this model that might be outdated six weeks after it's released?

**Walter Scheirer:** I don't think anybody's got a good handle on what that footprint really is. When you look at the carbon footprint of different industries, the datacenter is tiny. The big problems are cars and aircraft and power plants, which just dwarf datacenters. I think the good outweighs the bad in this case. That said, I could be completely wrong because no one knows how big these datacenters are. No one knows how big Amazon's or Google's cloud infrastructure is. These things are trade secrets, so even trying to guess is difficult.

## THE INTERDISCIPLINARY ADVANTAGE

**Jason Schlessman:** We'll get back to your book, but first let's talk about how you got into this field. What got you interested in working in technology and computation?

**Walter Scheirer:** It's always been a hobby, since middle school.

**Jason Schlessman:** It's worth pointing out that when you say "middle school hobby," you're talking about trying to teach yourself networking. When I was in middle school, I was just playing video games.

**Walter Scheirer:** The web was very new and websites about technology were pretty thin in terms of their technical concepts. My mom would take me to this technical bookstore, and the store people would eye me suspiciously— why does this kid want these books on TCP IP networking?

You could also go online to try to ask people, but they were horrible. Back then in tech, there was kind of an aggressive culture where you had to prove yourself before somebody would help you. All of this has become more accessible with Stack Overflow and now even large language models.

**Jason Schlessman:** Is it safe to say this exploration led you to working with Linux?

**Walter Scheirer:** Linux was on my radar right away. I knew from reading all of this material that Unix was the workhorse of the internet—that was the serious operating system. Windows was popular on the desktop, but if you really wanted to understand the internet, you needed Unix. And Linux was free. You didn't need a fancy Unix workstation to run it; you could install it on your PC.

**Jason Schlessman:** Since then, what areas have you worked in, and what are you focusing on now?



*Computer vision tools can help make old texts searchable for scholars.*

**Walter Scheirer:** When I started, I was a systems person interested in networking security. Then I was thinking about how to combine that with machine learning and computer vision, so I drifted into human biometrics. That led to more questions about core computer vision research areas like object recognition and scene analysis segmentation. Then I was interested in understanding biological vision, but for AI. Basically, could you build biologically inspired algorithms that correspond more closely with what happens in the brain? My lab is still doing a bit of that work, especially models that incorporate elements of visual psychophysics.

I was also interested in media forensics as an application— combining security and stuff from systems and visual problems. When I came to Notre Dame, I was doing all of this plus getting into the history of technology

and technology ethics. It's a lot of stuff, but it all fits together.

**Jason Schlessman:** You left out one thing: your work in digital humanities.

**Walter Scheirer:** How could I forget? My work also has applications in classics, medieval studies, and text analysis for Latin poetry. How do you find allusions across different texts? If you have high-resolution digital images of old documents, how do you transcribe them into Unicode? That would be really useful for scholars so they can do searches, and it's a challenging computer vision problem. It allows us to bring to bear a lot of state-of-the-art tools. You're not going to see that kind of work in many places—there's no money in it—so it's an interesting niche.

**Jason Schlessman:** And your undergraduate background is as much in the humanities as in computer science.



A lot of people in computing think they have a solution to all the world's problems, but they don't have a good understanding of even how to ask questions in another discipline.

**Walter Scheirer:** I was interested in international relations, things like war and politics.

**Jason Schlessman:** Did that humanities background shape your research? Your book has a lot of references to classics and poetry.

**Walter Scheirer:** Absolutely. As an undergrad, I did a liberal arts degree, and computer science was a secondary major in a liberal arts course of study. I took several philosophy courses, which shaped my thinking. In digital humanities, a lot of times somebody wants to do computing, but they just don't have the background. But the opposite tends to be more problematic. A lot of people in computing think they have a solution to all the world's problems, but they don't have a good understanding of even how to ask questions in another discipline. Because of my background, I've been able to work across these boundaries and do it credibly. I can speak the language of the various disciplines I'm working with because I have a bit more training than the average computer scientist.

I'm seeing this becoming a bit more common: at Notre Dame, we have a new BA degree in computer science and engineering that requires you to pick up another discipline as a focus area. I've advised students who have been doing computer science and English, for example. To solve some of the big problems out there these days, you need training in two areas—not superficial training, but very deep training.

#### **DON'T FEAR THE FAKE**

**Jason Schlessman:** Your most recent book is called *A History of*

*Fake Things on the Internet*. How does that come out of the work we've talked about so far today?

**Walter Scheirer:** The book talks about specific technologies implicated in fakery in some way. AI is a key theme, and it talks a bit about the dilemmas we're facing on social media and elsewhere related to misinformation. I try to treat this in the most realistic way possible. Coming back to my interdisciplinary background: I can bring to bear the tools of a number of different fields in terms of analysis. Some of the chapters are ethnographies: I interviewed some of the most interesting and strange people out there, especially from the early internet period, people who were faking the news, computer hackers, people involved in digital art, people involved in the early days of media forensics, to understand their thinking at that time and how that corresponds to our thinking today.

The major idea in the book is that the internet is filled with fake things, but that's mostly good. The internet is a creative space and was created to be that. That's why people like it. But there's this mainstream media narrative from the 90s that the internet is this "information superhighway," or that it's a database of facts that got polluted with all this fake stuff.

**Jason Schlessman:** We both know that, even going back to the BBS (Bulletin Board System) days, text files of fabricated information were being traded. This has been going on for a while. You say it took a flood of real-life content on social media



Popular and creative internet memes from A History of Fake Things on the Internet

to put the field of media forensics into the spotlight. Could that be generalized to say it took a flood of people being inconvenienced for media forensics to come into the spotlight?

**Walter Scheirer:** Definitely. The history of the internet is fascinating. Those of us in tech have been told a story that in the late 60s the US government, through the Defense Advanced Research Projects Agency (DARPA), created this distributed network to withstand a nuclear assault from the Soviet Union. That was the purpose of the internet. Then large tech companies realized it was useful for commerce. It would be useful for education. That's where you get this idea of the information superhighway in the 90s, during the handoff from the defense world to the corporate world.

But that is not really where the internet came from. If you go back a bit further, you have the ideas of the media theorist Marshall McLuhan, a famous professor at the University of Toronto associated with the 60s counterculture. He's

talking about information networks in a radically different way, saying this is going to create a global village and let the users of this infrastructure project their imaginations to other users.

He anticipates all these creative software tools we have now, everything from facial filters and tools like Photoshop that rework visual information to generative AI and tools like [Midjourney](#) that allow us to hallucinate interesting scenes by combining visual information in novel and surrealistic ways. That's the internet McLuhan is telling engineers to build. And if you look at who built the internet, you have these hippie figures at Bell Labs working on Unix interested in multi-user operating systems because they bring people together. There's this communal aspect to the internet completely missed in the conventional narrative.

By the time you get to the so-called information superhighway era of the internet, engineers are still thinking about McLuhan. In the first issue

of *Wired* magazine, the editors dub McLuhan the patron saint of the internet. As you pointed out, as soon as computer networks became popular, there were fake text files and computer hackers telling stories through these technical manuals. But along the way, it wasn't all fun storytelling and culture building. You see malicious actors moving in, and when something goes wrong, that's when media forensics feels relevant.

**Jason Schlessman:** You say in the book that we continue to blame technology for longstanding social problems instead of confronting the unethical behavior that nourishes them. For example, Chat GPT made generative AI more accessible to a large number of users. You wouldn't want to starve that creativity. You want to starve the bad intent that's under the surface of that particular field and has been for a while.

**Walter Scheirer:** This is a complicated issue. A lot of social problems in the physical world have simply moved over to the internet. I mentioned



*A simulation of a possible climate change outcome*

political polarization: that's not small, and technology is being used to perpetuate it. A lot of people don't want to confront the root issue because they can't seem to resist arguing about politics on the internet.

I have been writing about dialogue as a response to this. This is an idea going back to Plato: instead of having a heated debate about a controversial topic, we talk through it in a reasonable way, bringing diverse ideas that are perhaps in conflict with one another but not arguing about them—instead, trying to understand them. What we've done is develop a framework for this type of dialogue on the internet specifically. I have another book coming out later this year called *Virtue and Virtual Spaces* that makes this case.

**Jason Schlessman:** My favorite part of the book was where you talked about

a study on predicting the future of climate change. Could you discuss that?

**Walter Scheirer:** A chapter of the book talks about a generative AI model that will give you visual depictions of the future. It's designed to predict what a certain geographical location will look like 50 years in the future after climate change has taken effect. This seems kind of scientific and useful on its surface. We know climate change is a problem. We know there are lots of sophisticated models to predict what the climate is going to look like. But then the reasonable person steps back and says wait a minute, this oracle is showing me the far away future. Is that really possible? I interrogate this idea and look at this long-standing human belief in predicting the future and why in nearly all cases that is impossible. We'd all be rich and successful if we could do that.

**Jason Schlessman:** That was also the section with my favorite quote from the book: "Ascientific work is being cast as science." How can we address this?

**Walter Scheirer:** What's interesting about this specific AI model is it came out of Yoshua Bengio's lab. Bengio is a Turing Award winner, one of the three fathers of deep learning, and one of the most-cited researchers in computer science. Yet a bit of work coming out of his lab is not scientific in any way. We put a lot of faith in experts, but experts get things wrong from time to time. It's a classic problem in thinking: when you've achieved a great success rate, it's not hard to fool yourself into thinking you can solve an impossible problem. This is not the only case of that kind of error, but it was worth writing about because it is well situated in a long-standing misunderstanding about what is possible and impossible.



## BUILDING ON OPEN SOURCE

**Jason Schlessman:** That gets back to the open source movement. We can all benefit by being open to criticism, making mistakes, and taking other people's thoughts and input. At least in theory, none of us wants to be the be-all, end-all; we want to be part of a community.

You said Linux and Unix were initially important to you, but here we are decades later and you're still a strong proponent of open source. What value do you see?

**Walter Scheirer:** Open source has had a tremendous positive impact on the industry. It's improved the quality of software drastically. In my own career, if you look at the impact the field of computer vision's been having, it's huge. A big piece of its success has been the move to Open Access about a decade ago. Every time we do a paper, there's a repo associated with it. We want people to be able to use this stuff. It's not just about replication, it's about other people who want to use this and build on it. It should be a stepping stone for bigger work.

**Jason Schlessman:** What impact have you seen in making things more open in academia? Are you seeing more best practices being incorporated in your lab as a result of interacting with folks like us at Red Hat?

**Walter Scheirer:** Absolutely. The repos are important, and good documentation as well. Academics still have this problem of thinking, "We open sourced it, we're done." What am I supposed to do with it? Is there a useful Read Me for this? Can I get

it up and running? Is there a process to reach out and get some help? All of that stuff is important. We want people to understand how to run this stuff, and also, just for general awareness, give people a digestible explanation. Technical papers can be challenging to read. We did a Medium post with our code and tried to explain it in a clear way. You have to combine all these elements.

---

It's a huge advantage  
to have computing  
professionals working with  
academics.

---

**Jason Schlessman:** For the sake of readers: you're referring to a successful upstream contribution in the form of an [image forgery detection toolkit Python package](#). Even with that, I found that maintenance is a big thing: you put the repo out there, but what if people need help or want to contribute? I've seen, on the academic side, increasing openness to that mindset. If somebody has a pull request, instead of seeing it as "We already published the paper, what do we care?" it's more, "Wow, we have people getting involved, and we can revisit this." You once said that revisiting these problems also opens the doorway to thinking up new research problems and new innovations.


**Walter Scheirer:** Exactly. It's not just about the paper. It's also about improving that code base over

time. Some of the most successful research contributions in AI-related fields are those projects where there was a useful software package that became indispensable over time, and new features were being added, bugs were being fixed. Maintenance behind the scenes really made it better.

**Jason Schlessman:** Where do you see your collaborations with industry heading in the future?

**Walter Scheirer:** So many projects we have in the lab right now would benefit from exactly what we're talking about. It's a huge advantage to have computing professionals working with academics. A graduate student's job is basically to come up with cool ideas and implement those ideas in a proof-of-concept way. But if you want to make a project successful, you have to go beyond that. You have to remember there are users on the other side, even if it's still in an academic context.

Our work started in media forensics, and we will continue doing that, but there are a number of other really interesting AI areas, perhaps more fundamental stuff. My lab works a lot on open world recognition problems, which cross a lot of different application domains. There's a big need for open source software for a fundamental operation like that, and other interesting application areas to explore, so there's a big space for us to continue.

**Jason Schlessman:** Very good. Thank you for your time, Walter. I'm looking forward to future collaborations. 

Clouds that  
compete can't  
connect.

Says who?



/Keep your options open  
[redhat.com/options](https://redhat.com/options)



Copyright © 2023 Red Hat, Inc. Red Hat and the Red Hat logo are trademarks or registered trademarks of Red Hat, Inc., in the U.S. and other countries.

- ☐ A AWS
- ☐ B Azure
- ☐ C Google Cloud
- ☒ D All of the above



*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](http://www.massopen.cloud)



[contact@massopen.cloud](mailto:contact@massopen.cloud)

*Housed at:*

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

**BOSTON**  
UNIVERSITY



# Passive network monitoring with eBPF

Passive network latency monitoring offers a more holistic view of network performance without creating additional traffic. Researchers are developing a new tool to enable it efficiently.

*by Simon Sundberg, Anna Brunstrom, Simone Ferlin-Reiter,  
and Toke Høiland-Jørgensen*

**N**etwork latency is a determining factor in users' Quality of Experience (QoE) for applications including web searches, live video, and video games. That's why network latency monitoring is critical. Monitoring latency makes it possible to find problems and optimize the network proactively, and it has a wide range of other use cases, such as verifying Service Level Agreements (SLAs), finding and troubleshooting network issues such as bufferbloat, making routing decisions, IP geolocation, and detecting IP spoofing and BGP routing attacks.

Since 2020, our collaborative programmable networking research team at Karlstad University in Sweden has explored the potential of eBPF in the Linux Kernel to alleviate several problems with latency monitoring. This led to the development of [epping](#), a tool that leverages eBPF to passively monitor the latency of existing network traffic. In this article, we present some of the insights

gained from this project so far and the future directions we plan to take this research.

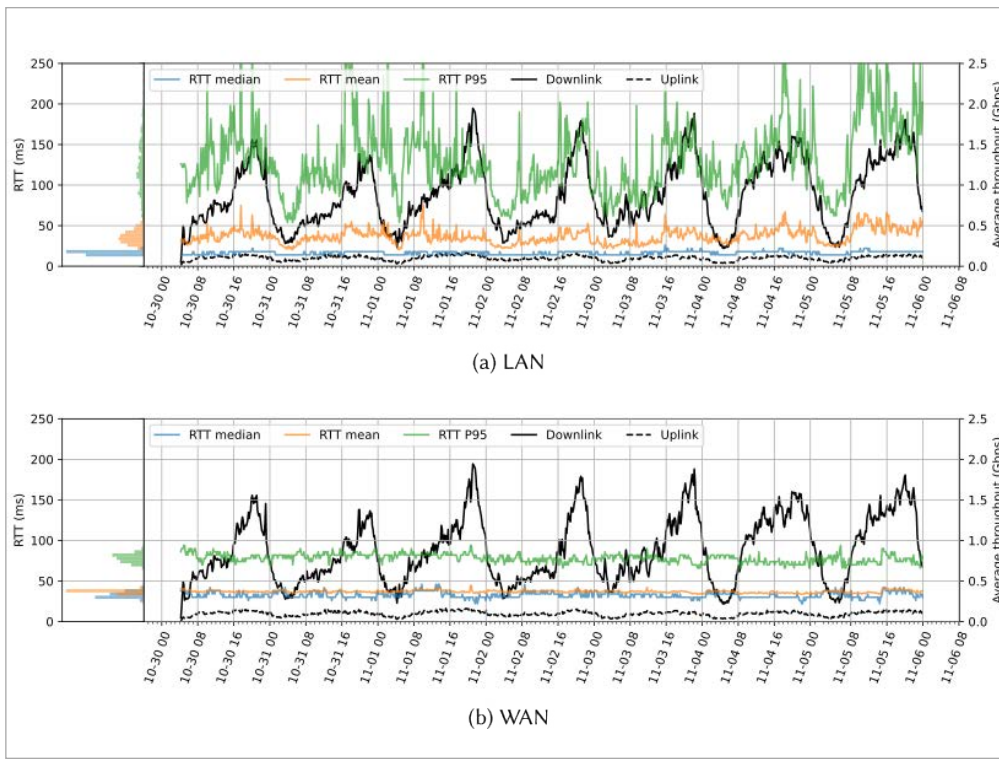
## WHAT IS PASSIVE LATENCY MONITORING?

While active network monitoring, like ping, is useful for measuring connectivity and idle network latency in a controlled manner, it cannot directly infer the latency application traffic experience. Network probes may be treated differently from application traffic by the network, for example, due to active queue management and load balancing, and therefore their latency may also differ. Furthermore, many active monitoring tools require agents to be deployed directly on the monitored target, which is not feasible for an ISP wishing to monitor the latency of its customers.

By contrast, passive monitoring techniques observe existing application traffic instead of probing the network. This means passive monitoring can run on any device on the path that sees the traffic, not only at the endpoints.



See "[Programmable networking project reports on its first year of progress](#)" in *RHRQ 3:3* (Nov. 2021)



**Figure 1.** RTT and average throughput at 10-minute intervals for one week out of the total one-month measurement period

In this project, we have implemented a passive monitoring solution that extends the functionality of the original Passive Ping ([pping](#)) utility and adapts it to use the eBPF technology that is part of the Linux kernel. eBPF adds the ability to attach small programs to various hooks that run in the kernel, making it possible to inspect and modify kernel behavior in a safe and performant manner. eBPF is generally well suited for monitoring events in the kernel, including processing network traffic, and is thus a nice fit for passive latency monitoring.

The utility we have developed, evolved Passive Ping (epping), is

an open source utility that runs on any Linux machine and can monitor the latency of TCP and ICMP traffic visible to the machine. That means traffic originating from the local host as well as traffic passing through the host when it is deployed as a middlebox, either standalone or as a container or virtual machine host.

## RESULTS FROM AN EPPING MEASUREMENT STUDY

To evaluate the performance of epping in a real-world environment, we set up a measurement study to collect measurements from within the core network of JackRabbit Wireless, a wireless ISP operating in El Paso, Texas, USA. JackRabbit

serves approximately 400 subscribers, of which around 95 % are households. JackRabbit relies on fixed wireless connections for both access and backhaul networks. Once traffic reaches the core router at JackRabbit's central site, it is routed through a middlebox/bridge running [LibreQoS](#), an open source QoS platform for ISPs. The bridge applies fair queuing, active queue management, and shaping to provide a good QoE for each customer, and also serves as our measurement point where we deployed epping. Measurements were gathered without impacting the forwarding capabilities of the machine: the overhead of running epping was low enough that the machine in question could still serve the offered load.

## Latency variation

The selected measurement point enables us to capture and analyze all traffic going in and out of the ISP network and to look at the traffic in both the internal (customer-facing) and external parts of the network. This allows us to examine how the latency varies over time in both directions. **Figure 1** shows one week of data from the total one-month measurement period.

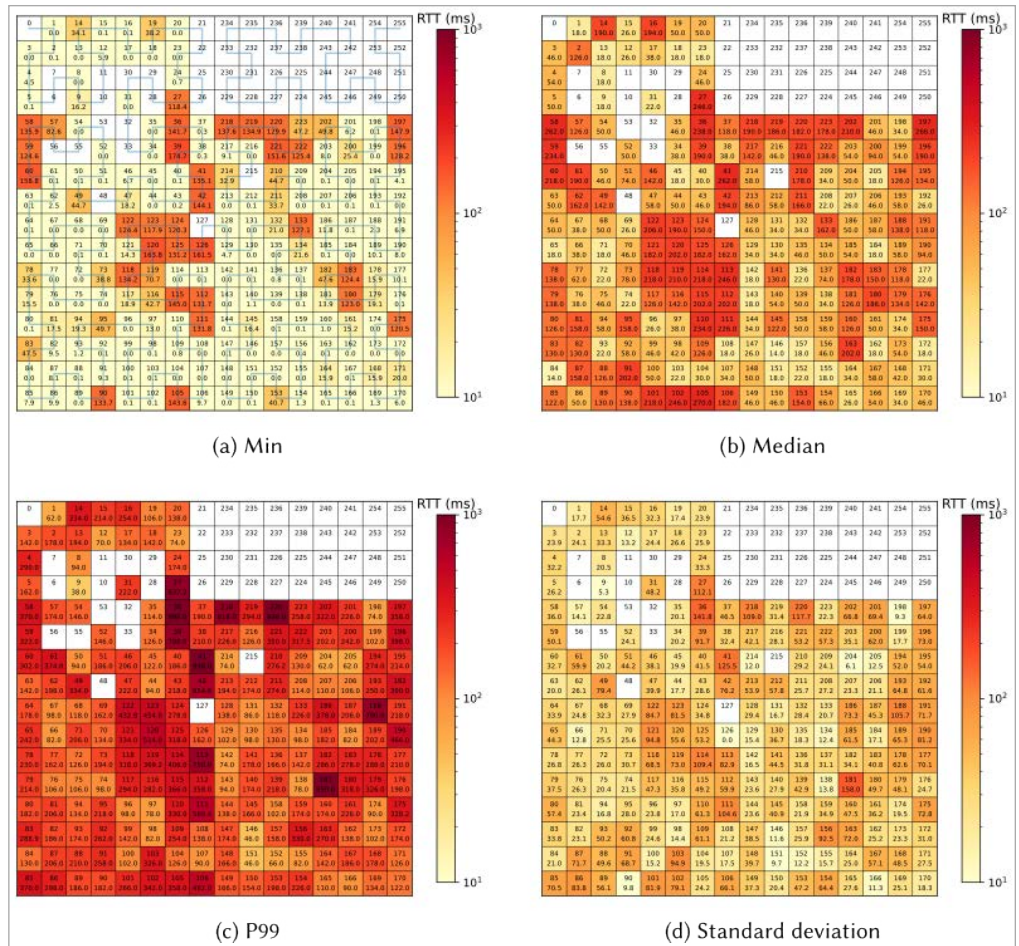
The figure includes information on the traffic load, making it possible to examine its potential correlation with latency. The uplink and downlink traffic loads (right y-axis) show a very clear diurnal behavior, with downlink traffic peaks of up to around 2 Gbps around 20:00-22:00 in the evening, then drops down to a few hundred Mbps between 02:00-05:00 in the morning. The

mean and 95th percentile RTTs on the Local Area Network (LAN) side vary over time, while the median appears more stable. Furthermore, the mean and 95th percentile RTTs appear to be partly correlated to the traffic load, where they also appear to be higher during the afternoon and evening and lower during the night and early morning. In contrast, the RTTs on the Wide Area Network (WAN) side do not show any significant variations over time.

The most likely culprit for the latency increase on the LAN side is the WiFi link at the subscriber's premises. The RTTs in the measurement study are likely mainly collected for traffic from portable devices connected via WiFi. It is well known from previous work that the WiFi hop often makes up a substantial part of the end-to-end latency and that its latency can increase drastically when it becomes the bottleneck link or when many devices are competing for access. During busy hours, the radio spectrum is also more congested, with the wireless links facing more interference and concurrent access. This, in turn, leads to a higher probability of packet errors and collisions, which are translated into more retransmissions and backoff periods on the link layer and, thus, additional latency. A congested radio spectrum could also impact latency in the ISP wireless access network, not just inside customer homes.

### Subnet traffic analysis

Another interesting property of the traffic we can analyze using epping is the variation in latency between different destinations on the internet. This is possible because epping captures traffic



**Figure 2.** RTT per /8 address block in the IPv4 space

metrics on a per-subnet basis (in /24 address blocks in this measurement regiment). In **Figure 2**, we further aggregate this into /8 blocks, allowing us to visualize the entire public (IPv4) address space at once.

The figure layout is inspired by the xkcd “[Map of the Internet](#),” and the map layout follows a Hilbert curve to situate numerically consecutive address blocks close to each other. The upper number in each block

shows the number of the block (e.g., 42 is 42.0.0.0/8), and the lower value shows the RTT in ms.

While most /8 blocks may contain traffic to a multitude of services and organizations, there are still some clear differences in the RTT statistics between the different /8 blocks. Furthermore, several of the blocks showing higher RTTs are clustered together. Looking at the minimum latency, in particular, it is interesting

that it identifies blocks that do not appear to have any nearby Points of Presence (PoPs). We can see that, among others, the blocks 27, 36, 39, 41-42, 58-60, many in 111-126, 196-197, and 218-222 show minimum RTTs well above 100ms. This largely corresponds to blocks managed by the Asia Pacific Network Information Centre (APNIC) and African Network Information Centre (AFRINIC) and, thus, likely primarily hosted in the Asian and African regions. Tail latency in the 99th percentile follows similar patterns as the minimum and median RTT, with mostly the same APNIC and AFRINIC blocks having the highest values. However, some blocks with low median RTT, such as 13, 31, 65, and 103, show a large relative increase in 99th percentile RTT, roughly an order of magnitude. This indicates that the RTTs in these blocks have


a long tail, even if the absolute RTT values are not that high.

While this analysis does not directly allow us to infer the sources of higher latencies, an overview of the entire networking space like the one above can be useful to understand the traffic originating from a particular network and can be used to initiate further analysis to determine the root cause and optimize network behavior.

#### **PLANS FOR FUTURE WORK**

As can be seen in the examples from the measurement study outlined above, with epping it is feasible to run a longitudinal analysis of the latency behavior of an entire network and extract detailed data that can be used to optimize the network. This serves as a proof-of-concept of the tool itself, and the measurements themselves also

provide an interesting insight into real-world network behavior at an ISP.

We extended our collaboration in 2023, and over the next two years, we plan to build on this work to expand the capabilities of the epping utility and incorporate ideas from it into Red Hat products. In particular, we plan to incorporate the passive latency monitoring capabilities of epping to enable the same kind of analysis of egress cluster traffic showcased above. In addition, we will work on enhancing the capabilities of epping itself so that it is more readily deployable as a standalone monitoring and measurement tool. This includes working on enhanced reporting capabilities, extending its coverage to additional protocols such as QUIC, and new features based on the feedback from the study excerpted above. 



#### **About the Author Simon Sundberg**

is a PhD student at Karlstad University, Sweden.



#### **About the Author Anna Brunstrom**

is a professor and research manager for the Distributed Systems and Communications Research Group at Karlstad University.



#### **About the Author Simone Ferlin-Reiter**

is a senior software engineer at Red Hat.



#### **About the Author Toke Høiland-Jørgensen**

is a senior principal software engineer at Red Hat.





# AI ON INTEL®



NOW BUILD THE AI YOU WANT  
ON THE CPU YOU KNOW.

Learn more at [ai.intel.com](https://ai.intel.com)



## Feature

**About the Author****Gordon Haff**

is a Technology Advocate at Red Hat, where he works on emerging technology product strategy, writes about tech trends and their business impact, and is a frequent speaker at customer and industry events.

His books include *How Open Source Ate Software*, and his podcast, in which he interviews industry experts, is *Innovate @ Open*.

## QUBIP and the transition to post-quantum cryptography

Quantum computing could put secure communication at risk sooner than you think. Current research aims to solve the problem before it starts.

*by Gordon Haff*

**P**ost-quantum cryptography (alternatively, quantum-resistant cryptography) probably consumes more bandwidth than it should in quantum computing discussions. That's because the potential to incrementally improve the efficiency of important but mundane tasks like optimizing logistics is a yawn for many people. Breaking today's public key cryptography, on the other hand, is both a concrete objective and something that could be a unique capability of quantum computing.

Given the singular importance of security protocols in so many modern uses of the internet, this potential for future quantum computers to crack current encryption protocols is a matter of legitimate concern—and therefore worthy of our attention.

### WHAT'S THE PROBLEM?

In their "Report on Post-Quantum Cryptography" [NISTIR 8105](#), published in 2016, the US National Institute of

Standards and Technology (NIST) offered the following background:

"In the last three decades, public key cryptography has become an indispensable component of our global communication digital infrastructure. These networks support a plethora of applications that are important to our economy, our security, and our way of life, such as mobile phones, internet commerce, social networks, and cloud computing. In such a connected world, the ability of individuals, businesses, and governments to communicate securely is of the utmost importance.

"Many of our most crucial communication protocols rely principally on three core cryptographic functionalities: public key encryption, digital signatures, and key exchange. Currently, these functionalities are primarily implemented using Diffie-Hellman key exchange, the RSA (Rivest-Shamir-Adleman) cryptosystem, and elliptic curve cryptosystems.

The security of these depends on the difficulty of certain number theoretic problems such as Integer Factorization or the Discrete Log Problem over various groups."

The problem is that a set of algorithms for quantum computers, developed by mathematician Peter Shor in 1994, could—on a sufficiently large and fast quantum computer—break current encryption schemes. This includes public key schemes such as RSA, ECDSA/ECDH (Elliptic Curve Cryptography), and DSA (an example of Finite Field Cryptography) in particular. However, quantum computing could also affect symmetric algorithms, including the security of the shared secret key exchange, and could force the use of larger key sizes.

Fundamentally, Shor's algorithm—which usually refers to his algorithm for finding the prime factors of an integer—provides the ability to find the prime factors of any integer number in polynomial time on a quantum computer, rather than the exponential time that a classical computer algorithm takes. This could reduce the time to solve the problem from wildly unrealistic years to potentially hours. That's a problem because public key algorithms rely on the fact that it's quick to determine if a number is a valid prime factor, but this key cannot be found by brute force because it's necessary to find the prime factors of integers that range from 1,024 to 4,096 bits.

## WHY DOES IT MATTER TODAY?

However, none of today's quantum computers are remotely capable of using

Shor's algorithm to factor the size of integers used by today's cryptography standards. So, no problem, right?

Not exactly. For three reasons.

The first is that changing cryptographic infrastructure takes a long time. NIST notes, "It has taken almost 20 years to deploy our modern public key cryptography infrastructure." In the Fortune 500, transitioning from RSA to ECC has taken five to seven years.

Second, many initial post-quantum cryptography proposals submitted to NIST were quickly broken; it's reasonable to expect that other vulnerabilities may be found in the future. Although the draft algorithms have been extensively vetted, it will likely still take time to prove them out prior to deployment at scale.

The third is that data created today could be vulnerable to decrypting once sufficiently fast quantum computers become available. Even if the data is re-encrypted once post-quantum cryptography becomes available, it will still be potentially vulnerable if someone made copies prior to the cryptography migration. And data can be sensitive for decades. There is a wide range of expert opinions on when a quantum computer will be able to break RSA-2048 in 24 hours, but the general consensus is in the range of 15 to 20 years.

## WHAT ARE INDUSTRY AND GOVERNMENTS DOING?

A number of different standards and other organizations are working on post-quantum cryptography standards. For example, in the US, NIST has taken the lead. In 2016, they called for proposals, of which they received 82

from industry and academia. By 2020, there were 15 remaining candidates split between public-key encryption and key-establishment algorithms/digital-signature algorithms. Many used lattice-based encryption technology. [Three draft standards came out in mid-2023, and the public comment period closed last November.](#)

In Europe, the [QUBIP project](#) is one effort to address the transition to post-quantum cryptography of protocols, networks, and systems. The project, a global collaboration among businesses, universities, and NGOs, kicked off in September of 2023. QUBIP's main objective is to define a standard and replicable transition process involving the adoption of post-quantum cryptography in hardware, including constrained IoT devices, cryptographic libraries (such as OpenSSL, NSS, Mbed TLS), operating systems such as the Linux Fedora distribution initially, communication protocols (TLS and IPSec), and applications (Firefox browser and digital identity). Red Hat is a contributor at the library and OS level and is helping with standardization.

Starting from the transitions of these five building blocks, QUBIP addresses their integration into three real-world systems (IoT-based digital manufacturing, Internet browsing, and Telco operator software network environments) at the system level, considering all possible cascading dependencies.

Challenges faced during QUBIP's first months have been:

- Transitioning the IoT component through the implementation of a



# NEVER MISS AN ISSUE!

Available  
in PDF and  
printed  
version

**SUBSCRIBE NOW**

Scan QR code to subscribe  
to the Red Hat Research  
Quarterly for free and keep  
up to date with the latest  
research in open source

**red.ht/rhrq**

secure element that provides a set  
of post-quantum cryptography  
implementations in hardware

- Transitioning cryptographic libraries through loadable modules to make post-quantum cryptography implementations available as part of the libraries' capabilities and to enable post-quantum/traditional (PQ/T) hybrid schemes for TLS 1.3
- Dealing with several cascades of dependencies to make PQ/T hybrid TLS available at higher levels (operating systems and applications such as browsers and digital identity frameworks)
- Hybridizing post-quantum cryptography and Quantum Key Distribution (QKD) for key exchange in IPSec, often used to secure communications in telco operators' software environments

To keep up with the QUBIP project, follow [their blog](#) for updates. With the public comment period complete, [NIST expects to announce](#) that the three new algorithms are ready for use in 2024. **RH**

**Funded by  
the European Union**

*The **QUBIP** project is funded by the European Union under Grant Agreement No. 101119746. 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.*

# Anchored keys: scaling of in-memory storage for serverless data analytics

The strategy for scaling data capacity varies according to volume, access patterns, and cost-effectiveness.

We look at an approach that achieves optimal results in the context of serverless data analytics.

by *Tristan Tarrant*

**B**ig data holds great promise for solving complex problems, but data-intensive applications are necessarily limited by the difficulty of supporting and maintaining them. The [CloudButton project](#) has created a serverless data analytics platform to democratize big data by simplifying the overall lifecycle and programming model using serverless technologies. Working together, the CloudButton project partners have created a FaaS (Function-as-a-Service) compute runtime for analytics that overcomes the current limitations of existing serverless platforms.

A key component of the CloudButton architecture is Infinispan, an open source in-memory data grid providing a key-value data store that distributes data across elastically scalable clusters to guarantee high performance, high availability, and fault tolerance. Infinispan can serve as both a volatile cache and a persistent data store.

The CloudButton runtime leverages Infinispan as a mutable shared-memory store for input and output data and intermediate results used and produced by serverless functions. While Infinispan addressed CloudButton's functional and performance requirements, we determined that its scalability model was not a good fit for the lifecycle of serverless workloads in a cloud environment. This article illustrates how we enhanced Infinispan to suit our needs.

## INFINISPAN'S DISTRIBUTED STORAGE

Infinispan stores data in memory using the following scheme. Key-value entries are stored in data structures called caches, which can be replicated or distributed. In replicated caches, all nodes have a copy of all the entries. In distributed caches, a fixed number of copies of any entry is stored across nodes. This allows distributed caches to scale linearly, storing more data as nodes are added to the cluster. Keys are mapped to nodes using a [consistent-hashing](#)<sup>1</sup> algorithm

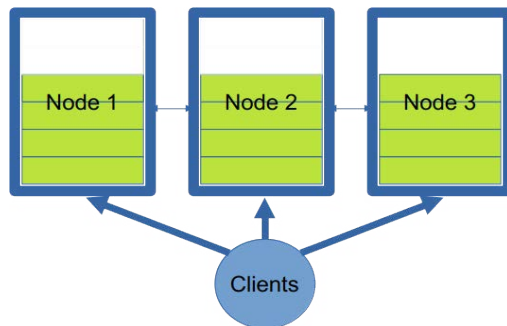


### About the Author

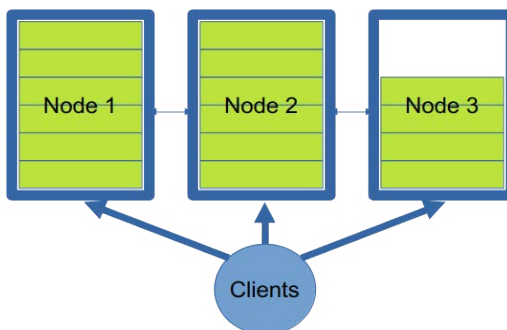
#### Tristan Tarrant

Tristan has been leading the Infinispan Engineering Team at Red Hat for the past eight years as well as being Principal Architect for Red Hat Data Grid. He's been a passionate open source advocate and contributor for nearly three decades.

<sup>1</sup> See also D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing (STOC)*, pp. 654–663, 1997.



**Figure 1.** *Balanced data*



**Figure 2.** *Scale-to-fit*

(currently [MurmurHash3](#)). These nodes are known as the primary owners.

Copies are replicated to other nodes, known as backup owners. These are chosen using anti-affinity rules to avoid colocation on the same server, rack, or site. For convenience, the consistent-hash space is divided into equal-sized segments. Remote clients are aware of the consistent-hash and segment allocation, allowing low read/write latency by performing operations directly with the primary owners in a single network hop. When nodes are added or removed, a rebalancing algorithm redistributes segments to nodes. The strategy attempts to minimize the number of ownership changes and, therefore, the amount of data to transfer. The state transfer

algorithm is non-blocking: clients continue to read/write entries, which are transparently redirected to the appropriate owner (see **Figure 1**).

This approach works well when the operational capacity is known upfront or changes infrequently. In these scenarios, priority is given to data integrity during node failure and the performance of a stable cluster. The rebalancing that occurs after adding or removing nodes has a noticeable impact on the network and the CPU, potentially introducing unwanted latency. In a cloud environment, however, consuming resources for capacity that may not always be needed is not economical. Instead, the cluster should be just big enough for the data it contains. This also

means that, if possible, it should scale to zero nodes if there is no data.

### ANCHORED KEYS

To address the above scale-to-fit scenario, we implemented a scaling strategy called anchored keys. Anchored keys relax some of Infinispan's distribution rules to remove the impact of state transfer by not rebalancing data when nodes are added.

In this implementation, the primary owner of new keys is the last node in the cluster. Writes can be issued against any node, but they will be applied only by the primary owner. The key-to-node mapping is stored in a secondary, replicated cache, which is accessible by all nodes in the cluster. A node is filled with data until it reaches its capacity. When a new node is added, only the key-to-node mapping cache is replicated to the new node. This is much cheaper than rebalancing the full entries with the values.

Without a predictable hashing strategy, clients can no longer determine the owner of a particular entry and must operate in a round-robin balancing mode. As the size of the cluster increases, the likelihood of a request hitting the primary owner of an entry decreases, significantly impacting latency. This only affects reads, since writes will always go to the node added last to the cluster (see **Figure 2**).

### ANCHORED KEYS AND FAULT TOLERANCE

To address the need for fault tolerance when using anchored keys, we need to bring back the concept of primary and backup nodes. Each node will, therefore, be associated with one or more backup



nodes: writes destined for a node will be replicated to all of its backups. The primary and its backups form a replica group. If one of the nodes fails, it is replaced by one of its backups, a new backup node is started, and data is replicated to it from the survivors in its group. **Figure 3** illustrates the architecture of this approach.

A drawback of introducing fault-tolerance to anchored keys is that nodes are assigned a role, either as primary or backup. This asymmetric nature makes managing their lifecycle significantly more complex: in the rebalancing architecture, all nodes are symmetrical and interchangeable.

## PERFORMANCE COMPARISON

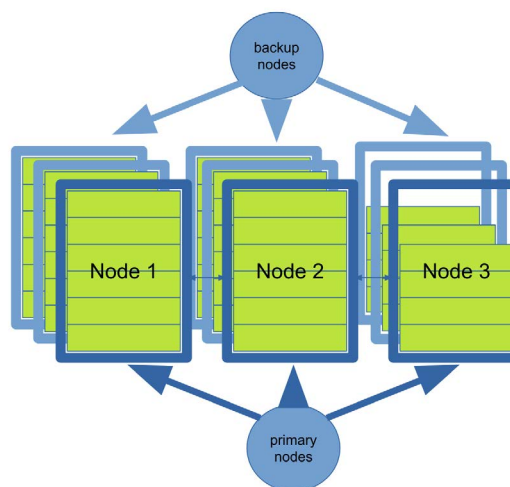
### State transfer

The advantage of using anchored keys over rebalancing becomes significant in the context of rapid scaling as desired in a serverless environment.

**Figure 4** shows that state transfer in the anchored keys mode is nearly 2.5 times faster than in rebalancing mode when dealing with 1 million entries, each storing a 2KB value. As the size of the values increases, the savings become even more significant since anchored keys do not need to transfer them.

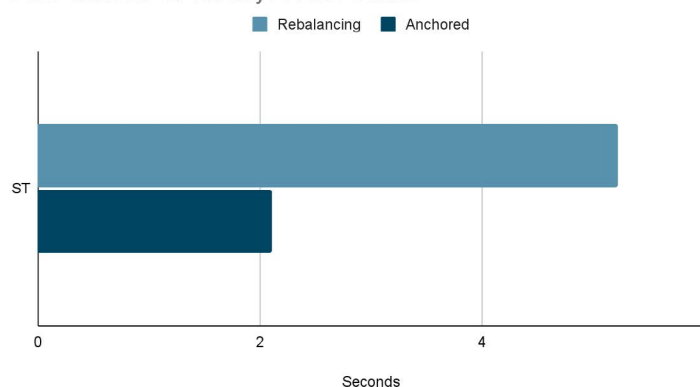
### Reads

Because anchored keys trade scalability performance for data-access performance, it's important to show the impact of its storage strategy on regular reads and writes. Infinispan's default consistent-hashing client intelligence ensures that reads and writes maintain near-constant performance, no matter how large the cluster.



**Figure 3.** Anchored keys with backup nodes

State transfer of 1M keys / 2KB values



**Figure 4.** A comparison of state transfer times for anchored keys vs. rebalancing

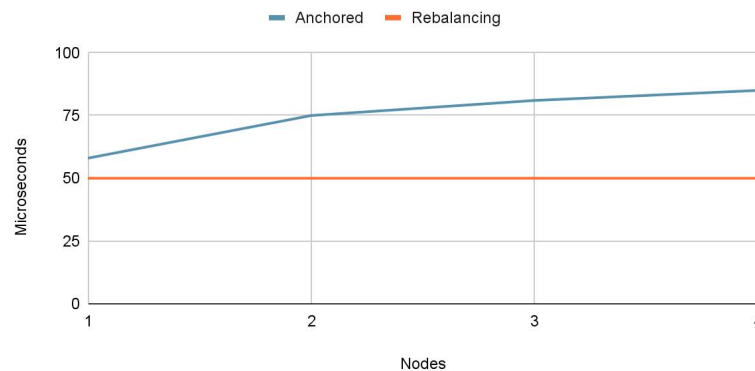
Because anchored keys forego the benefits of consistent hashing, we have to resort to Infinispan's basic intelligence, which is applying a round-robin algorithm on the nodes in the cluster. This means that, as the cluster size increases, the probability of a client interacting directly with the primary node decreases, thus causing an additional network hop between the server node handling the request and the primary owner

of the requested entry. While this degradation could be significant for traditional caching workloads, it may be considered negligible for computation-heavy serverless functions. **Figure 5** (next page) shows the impact of anchored keys compared to rebalancing based on the same scenario as above.

With a single node, the impact of looking up the ownership of an anchored

## Read performance

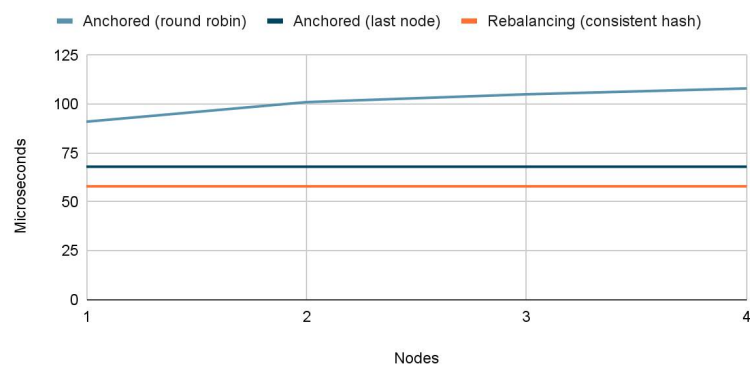
(10 threads, 2KB values, 1m warmup, 5m load)



**Figure 5.** A comparison of read performance times for anchored keys vs. rebalancing

## Write performance

(10 threads, 2KB values, 1m warmup, 5m load)



**Figure 6.** A comparison of write performance times for anchored keys vs. rebalancing

key incurs a 16% penalty, which is negligible. However, as the size of the cluster increases, the requirement to forward the operation to a remote owner makes the penalty much larger.

### Writes

The performance of writes shows the same progression as that of reads: as the size of the cluster increases, the round-robin algorithm has a decreasing chance of hitting the primary owner. The approach used by anchored


keys, however, allows us to introduce a write-specific optimization to the load-balancing algorithm. Since entries are only written to the last node in the cluster, and clients receive an ordered list of server nodes, clients can send write requests to the owner directly. **Figure 6** compares the performance of all three approaches.

As expected, the round-robin algorithm suffers the same penalty as seen for the read scenario. The last-node

approach, however, ensures linear performance comparable to the optimal consistent-hash algorithm. The impact of maintaining the additional key-to-node mapping makes writes 17% slower on average, which we believe is acceptable.

## ANCHORED KEYS AND SERVERLESS DATA ANALYTICS

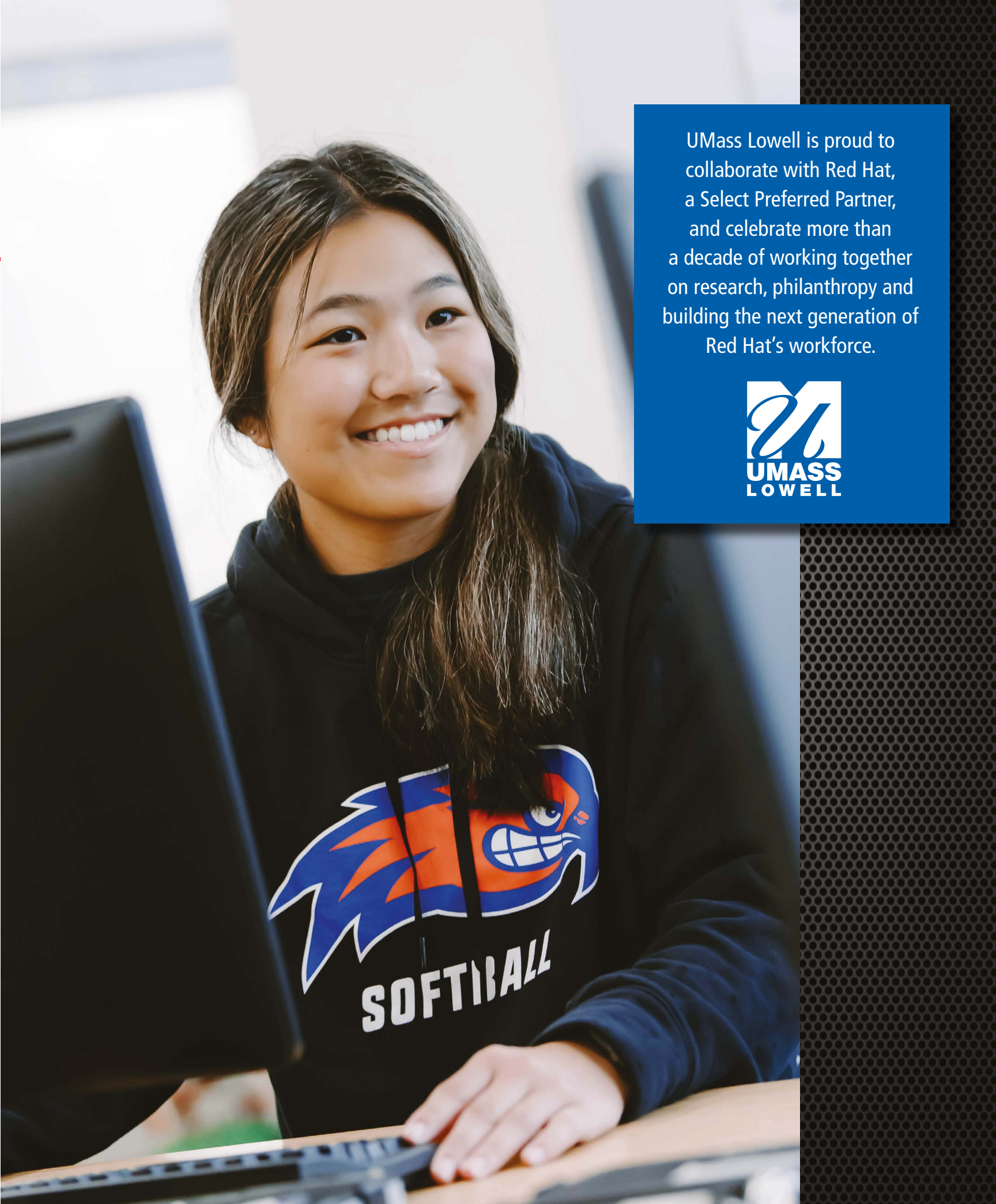
The introduction of anchored keys to Infinispan has made it an ideal choice as a high-performance shared data store for serverless analytics, where scale-to-fit capacity is required to balance infrastructure costs with throughput and availability. Combined with the other components developed in the context of the CloudButton project, such as the [Lithops](#) multi-cloud framework and the [Crucial Distributed Shared Objects](#) library, it provides a compelling platform for a variety of use cases, including scientific scenarios such as metabolomics and geospatial analysis, as well as broader business scopes, such as diagnostic and predictive analytics.

To learn more about the [CloudButton project](#), please visit the project homepage. 



**Funded by  
the European Union**

*Funded by the European Union under Grant Agreement No. 825184. Views and opinions expressed are 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.*



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.





## Column

**About the Author****Sanjay Arora**

leads the AI agenda for Red Hat Research and is mainly interested in the application of machine learning to low-level systems.

## Focus on artificial intelligence and machine learning

AI/ML research is driving performance and efficiency gains and building better developer tools.

by Sanjay Arora

---

*Red Hat Research and its university partners focus strategically on projects with the most promise to shape the future of how we use technology. Each quarter, RHRQ will publish an overview of our research in a specific area, such as edge computing, hybrid cloud, and security. In this issue, we focus on projects related to artificial intelligence and machine learning.*

---

A key area of focus at Red Hat Research is artificial intelligence (AI) and machine learning (ML). This includes building and optimizing systems that can run AI workloads, using AI/ML techniques to optimize systems software, and collaborating with the edge and hybrid cloud teams at Red Hat Research to use AI for their use cases. We collaborate closely with our university partners as well as Red Hat engineering.

A broad theme we focus on is using AI/ML to learn heuristics and policies that lead to more optimal systems. Optimality here refers to a performance metric, such as tail latency, throughput, or resource consumption, with energy consumption being especially important.

One example of this type of project is [NIC Tuning](#), based at the Red Hat Collaboratory at Boston University. Quantifying the impact

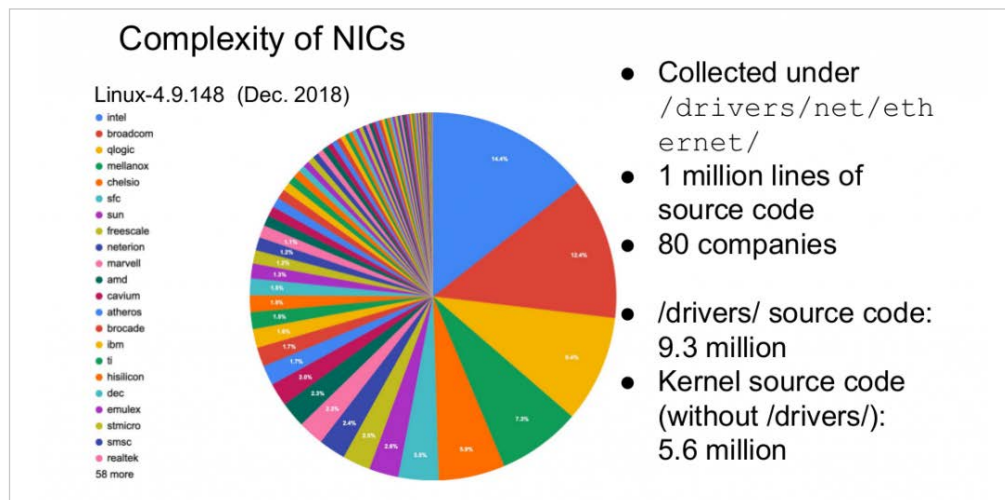
of interrupt delays (ITR) and dynamic voltage and frequency scaling (DVFS) on tail latencies and energy consumption for network-heavy workloads. On bare-metal Linux, we have seen significant savings in energy without notably compromising performance by carefully tuning ITR and DVFS. This tuning is generally done using black-box gradient-free techniques like Bayesian Optimization. We are now extending this work to (a) learning a dynamic ITR and DVFS policy (using reinforcement learning techniques) and (b) workloads running on OpenShift, which is a far noisier environment than a carefully controlled bare-metal Linux setup.

Another is [Compiler Optimization](#), also at the Collaboratory. The goal is to exploit an optimizing compiler's capabilities by injecting relevant information, through pragmas as an example, or by learning better heuristics for, say, selecting

optimization pass sequences to generate more performant or more efficient code. The compiler target in this case is an FPGA, but it could just as easily be an x86 processor. We have observed significant performance gains by learning policies to insert pragmas or to choose the next compiler pass. While our past work required a new training run for each new application, we are now focused on learning policies that can generalize across applications by reserving part of the state vector to be application code embeddings. A major project is using graph neural networks (GNNs) and control flow, data flow, and call flow graphs to learn code embeddings in a self-supervised way.

With the Red Hat Kernel Performance Engineering Team, we are working on [Linux Kernel Tuning](#). Tuning Linux kernel subsystems is also becoming a significant area of interest. The kernel has thousands of parameters that can affect a running workload in significant ways. Jointly optimizing these parameters in a sample-efficient way is a substantial challenge that requires careful performance measurements, tracing, and meaningful code embeddings, as well as cutting-edge developments in Bayesian optimization and reinforcement learning.

Given the effectiveness of large language models (LLMs), we also have projects that explore their use for developer tools. In the [Rust project](#) with Columbia University, we developed a static analysis tool called [Yuga](#) to detect lifetime annotation bugs in the Rust language. We are now exploring the use of LLMs to augment Yuga for both better bug detection and generation of corrected code. Unit



*Distribution of code contributions to Linux NIC device drivers by vendor. There are ~9 million lines of NIC device driver code as opposed to ~5.5 million lines of core (non-driver) kernel code.*

test generation is an essential part of robust software development. A recent project with Emerging Technologies aims to generate useful unit tests with LLMs. Based on our preliminary results, it seems likely that combining LLMs with classical static analysis will lead to a significant reduction in developer time spent on writing unit tests.


### NEAR-TERM OUTLOOK

In 2024, we hope to progress significantly on all the projects above. Some targets include:

- Evaluating various tracing tools (e.g., KUTrace, eBPF, etc.) to capture a running program's behavior with low overhead for downstream kernel tuning tasks. The hypothesis being tested is that traces contain vital information that can guide a search/optimization process
- Training graph neural networks to learn code embeddings from graphs derived from source code. These code embeddings are state

inputs to RL policies (in addition to other state variables) that learn various compiler-level heuristics. In general, for any tuning problem, we need our models to condition on a representation of the running application. These can be traces of the running application (the first target above) or static embeddings generated from underlying source code (this target).

- Evaluating the effectiveness of LLMs for generating unit tests and, more generally, exploring the combination of traditional static analysis with LLM code generation.

All these targets are fundamental building blocks for Linux kernel tuning, compiler tuning, and unit test generation respectively. In addition, we'll continue collaborating closely with our academic partners on projects applying AI/ML to systems engineering and our goal of building more performant and more efficient computing systems. 

# GET A LAPTOP THAT IS AI READY

AMD RYZEN™ AI TECHNOLOGY  
IS NOW BUILT IN

**AMD**  
RYZEN AI



\*Available on selected systems

**AMD**  
RYZEN

7000 SERIES

**AMD**  
RADEON

GRAPHICS