

# RH RQ

Bringing great research ideas  
into open source communities

Cloud AI unlocks performance  
in real-time edge devices

Can LLMs facilitate  
network configuration?

Meet Perun:  
a performance  
analysis tool suite

## Tomáš Vojnar

*A marriage of true minds:  
making university-industry  
collaborations succeed*



Red Hat  
Research Quarterly

Volume 6:3/4 | Spring 2025 | ISSN 2691-5278

### + BONUS INTERVIEW



## Akash Srivastava

InstructLab's chief  
architect talks custom  
language models and  
democratizing AI





# AI ON INTEL®



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

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



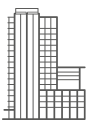
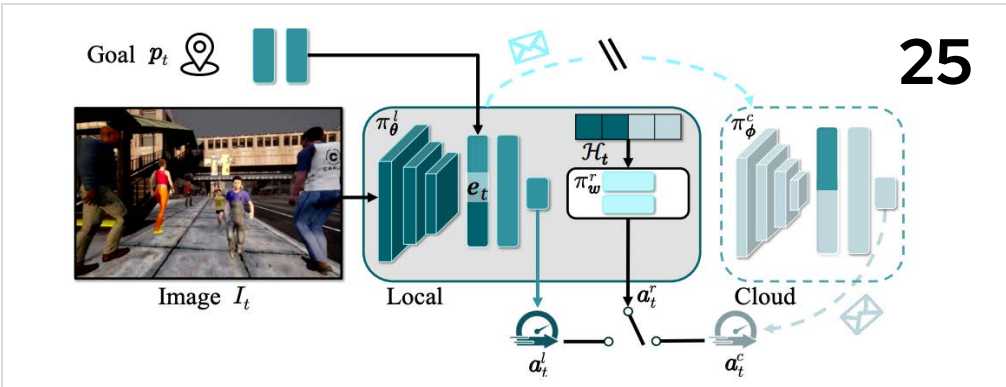
# Table of Contents



8



16



**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 editor
- 45 Making a research will: the human side of project migration

## Features

- 08 Making university-industry collaborations succeed – an interview with Tomáš Vojnar
- 16 AI DIY: how research is making custom language models work with more of us – an interview with Akash Srivastava
- 25 Bringing cloud AI into real-time edge devices to unlock performance
- 30 Can LLMs facilitate network configuration?
- 37 Meet Perun: a performance analysis tool suite

## From the editor

**About the Author**  
**Shaun Strohmer**

is the editor of Red Hat Research Quarterly. She has worked as a writer and editor in academic publishing for over twenty years, and since 2014 she has focused on software development, cybersecurity, and computer science.

## AI is changing research collaborations. How will open source research impact AI?

*by Shaun Strohmer*

**R**esearch isn't necessarily about getting the right answers, but asking the right questions. After five years as the editor of RHRQ, I'd like to think I've gained some perspective on the questions we want this publication to address. To me, the big-picture questions we're trying answer four times a year, across multiple disciplines and continents, are:

- What—and who—is new or newsworthy in the world of industry-academia research based on open source development?
- What unique advantages distinguish this kind of collaboration from other collaborative partnerships?
- How can we optimize these collaborations to have the biggest impact not only for businesses or universities, but for society at large?

In the past year or more, we've had to add another question to these three: How will AI change all of the above?

I'm excited to say that this issue of RHRQ tackles these questions head on, with some fresh perspectives. For instance, in each issue of RHRQ we interview someone in research, usually university based, about their work. The interview is the centerpiece of each magazine because it allows us to make connections between technological innovations and the contexts

that made them possible: the backgrounds of the people involved, the interdisciplinary collaborations that become more than the sum of their parts, the past trends and future vision their work is situated in. Put differently, it's not just about the results of open source research, it's about open sourcing a research approach that's proven very effective. And in this issue, you're getting a double helping.

You'll meet both Tomáš Vojnar, a long-time Red Hat Research collaborator who is now the head of the computer science department of Masaryk University (Czechia), and Akash Srivastava, the founding manager of the Red Hat AI Innovation Team who came to Red Hat by way of the MIT-IBM Watson AI Lab. Two PhDs, two deep theoretical thinkers, two different paths: Vojnar collaborates with industry engineers regularly but has chosen to stay in academia, while Srivastava found the opportunity to do research in an industry job. In each conversation, they address the impact AI has on research, in terms of opportunities, resource constraints, and partnership dynamics. If you read their stories side by side, you'll see that although they each function in different spheres, both are finding ways to balance the freedom and creativity of the academic side of research with the industry push for real-world impact, on deadline, with profitable results—and both cite working with



an open source company as key to their successes. As a professor, Vojnar has supported several open source projects in automated analysis and verification, including testing and dynamic analysis, that have been shared widely in the open source community and deployed for enterprise use. As an industry researcher at IBM and Red Hat, Srivastava developed the novel solution for synthetic data generation that became InstructLab, an open source project designed to put customizing LLMs within reach for users not trained in machine learning.

That said, as Vojnar points out, not all solutions developed in research are destined for life outside the lab. Vojnar is one of the developers of Perun, a performance analysis toolkit that began life with a small team of researchers at Brno University of Technology. The BUT team began working with Red Hat Research to enhance Perun with kernel-space analysis capabilities, then worked with the Red Hat Kernel Performance Engineering Team, responsible for kernel performance for Red Hat Enterprise Linux (RHEL). Their article in this issue, "Meet Perun: a performance analysis tool suite," describes the development of the tool and its application in the RHEL use case, and it also very helpfully outlines the challenges and requirements for making a research tool usable for industry users. As the authors observe, addressing those challenges often drives further research and leads to new solutions—solutions that might not exist without the push and pull of industry-academia relationships.


Our other two technical features this issue focus on asking the right questions about how AI can bring

value to computing systems. Simone Ferlin-Reiter, a Red Hatter who works with researchers at the Swedish KTH Royal Institute of Technology, asks the question "Can LLMs facilitate network configurations?" The short answer is a qualified yes—hopeful news for making network configuration less prone to human error and limiting the outages caused by network misconfiguration. But the questions raised by the team's research are the most valuable part of the story: How much impact does the batch size have on accuracy and cost? What is a tolerable balance between accuracy and cost? What opportunities are in reach, and what has to happen before we reach them?

In the article "Smarter AI, fewer resources: bringing cloud AI into real-time edge devices to unlock performance," Boston University professor Eshed Ohn-Bar asks whether we can design edge systems that use machine learning to seamlessly balance cloud and local resources to optimize for real-time accuracy, efficiency, and safety across different situations. Short answer: again, yes. In fact, UniLCD, the framework described in the article, is currently being integrated into Red Hat OpenShift, providing a flexible solution for large-scale, real-world deployments across various communication and modeling configurations. The article asks one of the most exciting questions research can raise: what's next? If reducing the energy consumption and cost of using powerful AI models at the edge is possible through solutions like UniLCD, could we extend it to domains like transportation, healthcare, or disaster response?

Red Hat Research and our collaborators get to engage in these questions because we provide the technology platforms and problem-solving that make it possible. US Research Director Heidi Dempsey has often been in the trenches with research projects reluctantly transitioning to new technology—say, migrating from VMs on OpenStack to Red Hat OpenShift and OpenShift Virtualization. Despite the benefits to be gained, the struggle, as they say, is real. Her column in this issue, "Making a research will: the human side of project migration," provides a very clear, readable guide to the process.

I said research isn't always about right answers, but if I could give a one-word response to my opening questions, I'd say "inclusion." An open source development model makes room for ideas from multiple sources so the best ones find each other and get even better. Collaborating across disciplines in an open source way drives better solutions because everyone has the opportunity to win. We can optimize these collaborations by finding ways to get more people involved—bringing a diverse set of skills and bases of knowledge to bear, but also helping people access the resources needed to test, implement, and improve technologies in multiple ways and settings.

How does AI change all that? Maybe a better question is how all that will impact AI. Ethical, open, transformational AI will happen in part because we're asking good questions and engaging lots of stakeholders to ask even more. So join us! 

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

# A marriage of **true** **minds**

A portrait of a man with short, graying hair and a light beard, wearing a red and white checkered shirt under a dark jacket. He is standing in front of a dense background of green ivy leaves.

**Making university–industry  
collaborations succeed**

---

*An interview with **Tomáš Vojnar**  
conducted by **Martin Ukrop***



## Interview

**T**omáš Vojnar has been a researcher, professor, vice-dean, and department chair—what about a marriage counselor? In conversation with Red Hat Research engineer Martin Ukrop, Tomáš—now head of the Department of Computer Systems and Communications in the Faculty of Information Technology at Masaryk University—joked that a good relationship between academic and industrial partners can be like a marriage: to be successful, you need to focus on the long-term, communicate well, and accept each other's quirks. (Then again, marriage counselors don't have to deal with the rise of AI.) Red Hat's eight-year partnership with Masaryk University has definitely stood the test of time—in December 2024, [Red Hat committed to another five years](#) of supporting research collaborations in cybersecurity, AI, and other strategic research areas with Masaryk. In this interview, Martin and Tomáš discuss the benefits and challenges of university-industry collaborations, how AI could influence the dynamics, and why open source is a key ingredient for success. —Shaun Strohmer, Ed.

**Martin Ukrop:** Let's begin with a big question. How does cooperation between industry and academia work? While industry does its own research for its own applied use, and some people in academia are interested in applied research, research generally has different incentives and works at a different speed. Where do you see the benefits of the two sides cooperating?

**Tomáš Vojnar:** It is different. In academia, you have more freedom. You can spend years trying to solve some hard problem, so long as you can publish. In industry, it's much more goal oriented. There are exceptions in some research centers of the biggest companies, but they still communicate with the production groups. But I think that these two approaches can naturally benefit from each other.

**Martin Ukrop:** Wouldn't they clash into each other? From my experience on the industry side, although academia is where new ideas arise, it seems to move more slowly compared to industry.

And from the academic view, industry seems to be interested in just the first step and not necessarily in the depth underneath it and the principles. So how do they complement each other?

**Tomáš Vojnar:** Because of the freedom in academia, there is a challenge to come up with something new. And then to make it applicable, that's the task of industry. I'm not saying anything new here, but industry can give academia interesting problems to solve.

If the situation goes in the best possible direction, there may be some researchers that say, "Yes, this is something where my methods apply," or if it's not directly applicable, it's close enough. Part of the problem in collaboration between industry and academia is that industry sometimes identifies a problem and they expect people from academia to immediately start working on it. But that's seldom the case. And sometimes new ideas in academia arise independent of what industry asks for, but if



### About the Interviewer **Martin Ukrop**

is a Principal Research Software Engineer 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, Czechia, focusing on human aspects of computer security. He remains an active teacher as well as a life-long learner.

Academia doesn't  
have the resources  
that some of the  
richest companies  
have.

the connection is there, companies can realize those ideas are good and can be exploited in some way.

### THE IMPACT OF AI

**Martin Ukrop:** With the focus on AI, it seems to me that industry-academia cooperation is a bit different. While academia is often at the forefront of new ideas, in AI, industry—at least publicly—seems to be ahead. Do you see it that way?

**Tomáš Vojnar:** I'm not an expert in AI, but of course I see changes. I see that in many areas, including mine, people are trying to come up with combinations of what they were doing plus AI. This makes perfect sense, because machine learning and AI alone will not solve everything. I recently attended a premier conference on logic programming, and many of the talks were about combinations with machine learning and machine reasoning—for example using ChatGPT or similar systems to help translate natural language to more formalized, structured notation that is amenable to precise, reliable machine reasoning. But a human expert evaluates that the translation is correct—that it was not shifted completely.

As for the speed and depth of AI research in industry and academia, academia mostly doesn't have the resources that some of the richest companies have. It's almost impossible for people from academia to do training of large neural nets. Here, naturally, academia is behind industry. That said, I think it pays to be skeptical of some industry AI claims. Some of them are more marketing oriented,

because companies need to say their products are AI-powered or AI-enabled, even if it is of no real benefit to customers. So while it's true that there are strong companies leading the crowd in AI, some of what we hear may be driven by a marketing bubble.

At the same time, there may be researchers at universities working on the deep theoretical background of AI, which will be needed. We need to have some understanding of what's happening in the neural networks in order to have some guarantees that the reasoning is meaningful. Or we should have combinations of machine reasoning and neural networks, so we can have more trust in the results.

**Martin Ukrop:** You have to admit that the AI boom shows the immense capability of industry to pivot on what we might call "The next big thing." A lot of companies started implementing true AI and AI enhancements into their production research pretty quickly, while implementing AI in university research is slower. I haven't decided whether that's an advantage or a disadvantage, however.

Are you seeing any direct impact or improvements from AI in your own research?

**Tomáš Vojnar:** In one branch, yes, directly. With one student, we are combining static program analysis with machine learning. The aim is to use graph neural networks to prioritize warnings produced by static analysis for the developers so they can concentrate on those errors that are more likely real errors. Such approaches do already exist; they are,



however, often closed source. We are aiming at an open source solution.

In other areas, we have plans to apply AI in the combination of machine learning and machine reasoning, translating from natural language to more structured language, as I mentioned earlier. We have a collaboration with Honeywell and some other partners working on this subject in the area of critical systems. Using an approach of the kind I mentioned, we could, for example, translate aircraft systems specifications in a natural language to some more formalized notation on which some formal reasoning can be done, then translate the results back. So these are two concrete examples.

## COOPERATION BENEFITS

**Martin Ukrop:** Speaking of your own research, what benefits do you get from collaborations with industry?

**Tomáš Vojnar:** For me, inspiration is probably the most important factor. I want to work on problems that are real—real in the sense that somebody in industry is interested in solving them and trying to apply the tools or theories I’m working on. If it gets applied, that’s extra rewarding, though of course you don’t always get those results. But when you have a result that is nice theoretically and it’s applied, that’s an excellent feeling.

**Martin Ukrop:** So it’s not only about the inspiration at the beginning but also the applicability at the end?

**Tomáš Vojnar:** Yes, though I’m still primarily a researcher. I do not insist on having everything applied. It’s



*Members of the Red Hat Research team in the recently created Red Hat Chill-Out Zone for students in Masaryk University’s Faculty of Informatics.*

enough for me to see that it has real potential for being applicable, because getting it to real applications is a long, long journey. We see it now, with [Perun](#) and other things we’re working on. (Perun is a system for software performance analysis and testing support plus storage of performance data across multiple software versions for comparisons and visualizations. See the article “Meet Perun: a performance analysis tool suite” in this issue of RHRQ.)

**Martin Ukrop:** If you get a research result applied upstream and used by literally millions of users, is that valued in academia?

**Tomáš Vojnar:** Scientific publications are still the most highly valued, which is natural. I can’t imagine a researcher

without a publication, but I can imagine research without applied results. However, I would say that, at sensible universities, applied results are taken into account seriously. One problem is that they are even more difficult to evaluate than papers.

**Martin Ukrop:** With papers, there are metrics and quartiles of publishers and conferences, but with applied results, how do you determine what is important, interesting, or impactful?

**Tomáš Vojnar:** In my view, there’s no other option than peer review. One has to describe the result, and several evaluators have to go through it and evaluate it. And, according to me, one should base the result on research. If there are publications related to the work, you can say, “I started with this

But if the two  
sides are trying to  
communicate, there  
will be successes. It's  
like a marriage.

idea, and it was published at a great conference or in a reputable journal, and it attracted some citations. Then we turned it into a tool, and the tool is used in a company or public body.”

You should try to estimate how many users one has; however, if a tool is specialized for a few people, say for police, you cannot expect the same number of users as something like ChatGPT. So this must be evaluated by people. You can also consider whether it's used in one country or globally, whether it generated some economic impact, if it is expected to be monetized, and so on.

**Martin Ukrop:** During your research you have cooperated with multiple companies and seen the industrial influence on academic research. Where did that come into your career path?

**Tomáš Vojnar:** I started studying at Brno University of Technology (BUT), Faculty of Electrical Engineering and Computer Science. I graduated from Master's studies in 1996, then continued at the same faculty as a PhD student. I finished my PhD studies in 2001, then went to Paris as a postdoc. I spent two years in a laboratory called LIAFA (Laboratoire d'Informatique Algorithmique: Fondements et Applications), then returned to Brno and joined the new Faculty of Information Technology (FIT) as an assistant professor, gradually proceeded to full professor, and for some time I was the FIT Vice Dean for science and research. Most recently, I moved to the Faculty of Informatics of Masaryk University (FI MUNI), though I still have an appointment with BUT to finish working with my PhD students there.

My experience with cooperative research started when I was employed in a European project and we had cooperation with Ericsson. But I have never been a pure theoretician, working with pen and paper and being happy just proving something. I've always wanted to have prototypes and play with them in some real-as-possible case studies, and I began to recognize that it's not so easy to get those opportunities without cooperation with industry.

That's when I started cooperating with Red Hat, with the motivation that I needed some links to industry to have some real stuff to work on. At the same time, there was interest from FIT to have somebody working with Red Hat. So with a push from both sides, I went for it.

**Martin Ukrop:** Which fields of research did you start with?

**Tomáš Vojnar:** At the very beginning, I worked on Petri nets (a modeling language for the description of concurrent systems) and their use for analysis and verification. I was really focused on this one modeling language, but I started to realize that it's not the language but the utility of the language, and that there are other languages and other means that seemed better. So I started work on analysis and verification methods in general, trying to select those that seemed suitable for those analysis and verification tasks that I was interested in.

Since then, I have always worked on formal methods as well as testing approaches applied mainly for analysis and verification but also in other areas such as optimizing compilers, regex pattern matching, or network



traffic analysis. In addition, I have also been interested in the background theories of automata and logics.

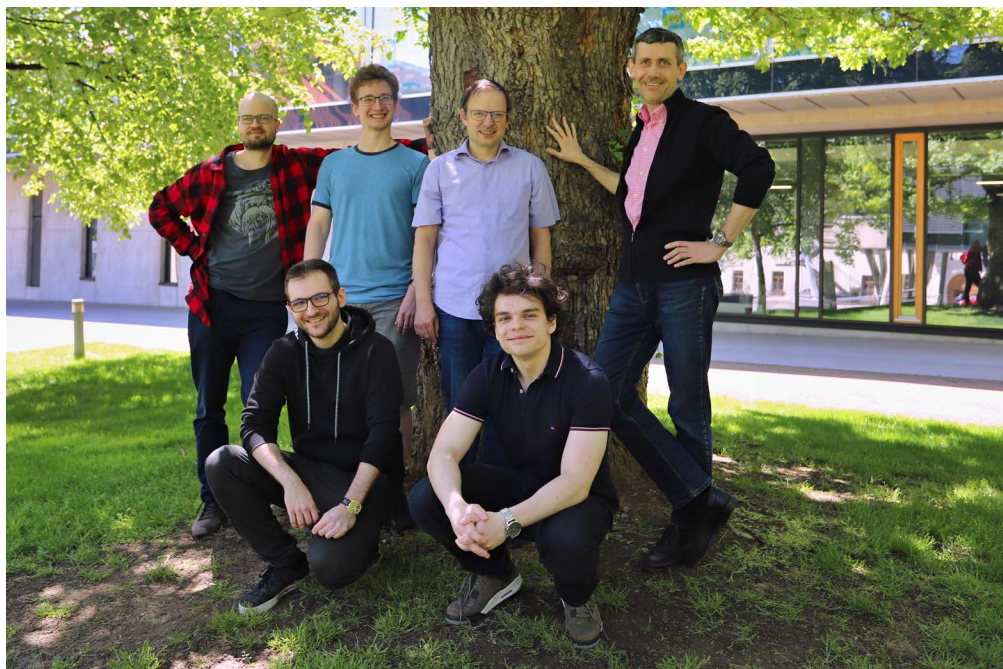
## OPTIMIZING COLLABORATIONS

**Martin Ukrop:** So far you've described two patterns of cooperation: when industry comes to academia with a commission, a problem they want solved, and when an academic researcher has an idea and industry notices it. And academics can reach out to industry. Do you see that some of these patterns function better than the others?

**Tomáš Vojnar:** I think the best situation is when you have long-term collaborations between industry and academia in which both sides slowly get acquainted with each other and they start understanding how the other side works. You are right that, in academia, the process is slower. People study a long time to get deeper results, and they are not willing to change their direction so quickly and so easily. In industry, you need results faster, concerning concrete problems.

Each side needs to understand these two things and tolerate the fact that some effort will go in vain. There will be suggestions coming from industry that nobody in academia will take. There will be ideas coming from academia that nobody in industry will be interested in. But if the two sides are trying to communicate, there will be successes. It's like a marriage.

**Martin Ukrop:** Would you say that research cooperation is different when you have an enterprise Red Hat's size compared to a middle-size company



*Vojnar with collaborators from Brno University of Technology, including Perun leads Tomáš Fiedor (far left) and PhD candidate Jiří Pavla (standing to the right of Fiedor).*

or a small regional startup—or even a university spinoff doing research?

**Tomáš Vojnar:** If it's a huge company, you usually already have some employees who are more research oriented, so it's easier for them to find some common subjects with academia. If it's a spinoff, which really must concentrate on one concrete product, it's difficult. Either there is a perfect match from the very beginning, or there will be no cooperation.

**Martin Ukrop:** So it seems that the smaller the industrial partner is, the more fragile the cooperation would be. Larger companies have the flexibility of going into research that may not be useful and directly applicable to them today, but

maybe tomorrow or the day after tomorrow—figuratively speaking.

You mentioned earlier that industry-academia cooperation works best when there is a long and slow period of getting to know each other. How does that apply to the story of growing the cooperation between your research group and Red Hat?

**Tomáš Vojnar:** It's been a long time. At the beginning, I was supervising some students working in Red Hat and overseeing administrative relations. But then I started really trying to find applications for what I was working on—program analysis—in Red Hat.

**Martin Ukrop:** This was more in the direction of you seeking applications

in industry rather than industry coming with a commission?

**Tomáš Vojnar:** Yes, and some of the directions were not that successful—rather, they were successful in academia but they didn't find real applications in Red Hat. Still, one of those, the collaboration with Kamil Dudka about verification of memory safety (AUFOVER), resulted in a tool called Predator, which is still winning in some international competitions. And some of the ideas were implemented in other tools, especially [CPAchecker](#) for Linux drivers.

Perun also traveled academia to industry. The Red Hat Kernel Performance team works with it on a regular basis, so it's been successfully applied. As of now, Perun is deployed in the day-to-day CI pipeline of kernel performance testing, and the detailed analytical results are then used for deeper investigation of individual performance degradation cases.

There's also hope for further collaboration, which I'm really excited about. I was showing some visualizations of performance data from Perun at the Lab Day here at Masaryk University. My colleague Barbora Kozlíková, who works on visualization, saw it, and now there is a chance we will start collaboration and spark something new. Such possible collaborations do not appear all the time, however. One needs to work a long time in the area to encounter them.

**Martin Ukrop:** As a person responsible for overseeing research cooperation with universities, I can confirm the truth of the graduality of it all. You

described an advanced point of implementing deep research within the pipeline, but there's been a years-long working cooperation to get to that moment. And even before your research cooperation, there were smaller things, like supporting smaller research engagements with bachelor's and master's students and presentations on university events, that were nudging the relationship from both sides until we arrived at the point of supporting PhD research that will get applied. So this is years in the making, and hopefully we'll continue for many more years.

**Tomáš Vojnar:** I very much hope so.

**Martin Ukrop:** Given your experience cooperating with industry, are there any dos and don'ts that you would suggest if a company wants to cooperate with academia, or vice versa?

**Tomáš Vojnar:** As I said earlier, you have to accept that it will take a long time, that the cooperation must develop gradually. If someone says, "Here are the subjects, take it or leave it," things will usually stay on the "leave it" side.

Another thing that's really crucial is open source. If the research is about technology that academia cannot publish and cannot speak about, this is very sad. Fortunately, Red Hat has always been a leader in that.

**Martin Ukrop:** Some companies are doing interesting research but keeping it proprietary and unpublishable.

**Tomáš Vojnar:** Yes, which is really bad. Even for students, at least in Czechia, the results must be made public. There may be some delay, but not too long.


You can hardly have students working on something that must stay secret.

**Martin Ukrop:** Another learning that I would emphasize is having the right scope. In almost all the projects you mentioned in cooperation with Red Hat, first successes came via reducing the scope—by doing the verification just on some of the drivers and not the whole Linux kernel, or doing the performance analysis of just a piece, rather than the whole.

**Tomáš Vojnar:** Or if there is already some technology, then really try to find where it could be applied. And eventually, with luck, there will be an application for it.

**Martin Ukrop:** A final question: you mentioned that there is interesting research outside academia, be it large corporate research centers or dedicated research teams in small companies. Have you ever been tempted to transition to this type of research, potentially cooperating with academia still, but from the other side?

**Tomáš Vojnar:** I value the freedom I have at the university. If I decided to move to industry, I would go to a large, strong company to be close to how universities work and have more freedom. But that's a question of personality. I would never go to a spinoff or startup either—that's too fragile for me. That said, I would love to support somebody starting a spinoff or startup in an area close to me.

**Martin Ukrop:** OK, so if there's a startup reading this article, they can approach you! Thank you for the interview. 

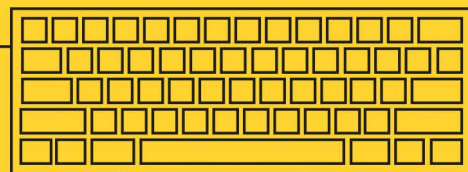
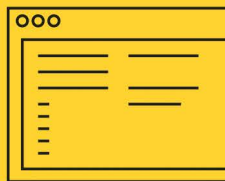
Your research,



projects



and education partner.





# AI DIY



How research is making custom  
language models work with more of us

---

An interview with **Akash Srivastava**  
conducted by **Heidi Dempsey**

## Interview

**H**ow many lives am I impacting?" That's the question that set Akash Srivastava, Founding Manager of the Red Hat AI Innovation Team, on a path to developing the end-to-end open source LLM customization project known as InstructLab. A principal investigator (PI) at the MIT-IBM Watson AI Lab since 2019, Akash has a long professional history as a researcher, which makes him a great person to shed some light on the often obscure pathways from research to product to product adoption. Red Hat US Research Director Heidi Dempsey interviewed Akash about his own path from curiosity to research to real-world impact, how he sees the democratization of AI evolving, and what cool things he and his team are working on next (hint: it's bleeding-edge research in Generative AI, and it involves Red Hat's recent acquisition of Neural Magic). Akash and Heidi also discuss balancing creativity with business demands and keeping the life-of-the-mind excitement and risk-taking spirit of research alive in industry settings. —Shaun Strohmmer, Ed.

**Heidi Dempsey:** You and I have a common interest: the importance of research as something that feeds development and eventually product. Let's start by talking about that in relation to InstructLab and democratizing AI development. Do you see that happening in multiple steps along the way, or by creating something that goes out to millions of people and then it's democratized?

**Akash Srivastava:** I like to think about democratization happening in different tiers. We started with the ChatGPTs of the world: proprietary models where the science itself was not democratized. Then some labs and companies broke through that tier by creating models and writing about them in detail, sharing the science behind them. To some, that was democratization. "To some" in this instance means "people who could afford the hardware." But it was a step forward.

Then someone said, it seems like language models only work when you have a lot of GPUs. Can we make smaller models? That was the next tier. In both software and hardware used for

training models, we made tremendous progress that allowed people to use commodity GPUs to use for training. A lot of open source effort focused on how to squeeze out every ounce of compute these hardwares offer. Now you can actually use your gaming PC. You don't have to get a \$50,000 card; you can get a \$4,000 card and start playing out this technology from your desk. To me, that is democratization.

Now we've reached a point where AI researchers and engineers can basically do everything they want to on these models, but what about people not trained in machine learning? That's where we're trying to make an impact now. Say I have domain expertise, I don't have any knowledge about how Generative AI works, but I know it's very useful for the things I do. My productivity goes up if I can use it, but I'm stuck relying on other companies to create something for me. With something like [InstructLab](#), I can use one of these models, and the tools to use these models, in my task. I think that's the ultimate level of democratization, when LLMs become useful for everybody.



About the  
Interviewer  
**Heidi Picher  
Dempsey**

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





# InstructLab



*Akash loves the Labrador breed of dogs, so much so that he initially named the InstructLab project simply "Labrador." To celebrate the successful launch of InstructLab, Akash got a Labrador puppy and named it Ladoo—the same name, by popular vote, given to the InstructLab cartoon mascot.*

**Heidi Dempsey:** So the first stage was democratization of the math and the programming, and the second stage is making it easier to use. Is there more to it?

**Akash Srivastava:** First the science got democratized, then the engineering got democratized, and now the application layer is coming. And when I say engineering, I mean low-level engineering, hardware- and kernel-level engineering. Now that we have the science and engineering democratized, application developers and domain experts can come in and start building these beautiful apps for a much wider audience.

**Heidi Dempsey:** This reminds me of the invention of HTTP: first there were distributed systems, but they were expensive, and the hardware and

software to run them was all research lab stuff. Then eventually because of the protocol, we increased its availability. Now it's on your phone, and all that stuff from the early days doesn't matter to users anymore—it's all behind the scenes. When do you think we'll see the same thing with AI, and people only using AI on their phones?

**Akash Srivastava:** I think a lot of people are already using it on their phones!

**Heidi Dempsey:** They definitely are. People want AI tools like Perplexity or ChatGPT or Apple Intelligence without having to understand what's behind them.

**Akash Srivastava:** I use AI tools very often. The way I program has changed. [GitHub Copilot](#) was the first

tool that actually impacted my daily life, and now [Cursor](#) with its fancy composer makes it seem almost like I'm programming in English.

**Heidi Dempsey:** You're almost becoming a scientist instead of a programmer, right?

**Akash Srivastava:** I will admit, I was never much of a programmer [laughs]. For me, programming was a tool that helped me do my research. I was never incredibly good at it, but with these tools I'm empowered to take my ideas to prototyping remarkably fast. The current generation of language models is democratizing natural-language-related work because a lot of our business processes are basically a transaction in language, whether it's financial forms or legal processes or business processes. Kids these days can create



apps using these things. They don't need to go to a university or wait to get advanced knowledge. With these tools, software engineering democratization is happening right now. My gut feeling, at least on the research side, is we're going to see a lot more democratization of engineering in general.

Here's an example: we have some active projects at MIT with the mechanical engineering department, where we're looking at the equivalent of IDEs for them, like CAD tools and other engineering design simulators. It takes ages to develop expertise there, but what if there were tools like Copilot for engineering design, chip design, or circuit design? We're going to see a flurry of startups and research labs producing Copilot- or Agent-like products for different branches of engineering, which is true democratization.

Any place where you would naturally use computer programs, it's just a matter of time until a piece of generative or other form of AI will come and help you as an assistant. One issue right now is how critical the domain is, and what the safety standards are. As you move towards domains like mechanical engineering, civil engineering, or electrical engineering, you don't have much slack. A tiny error in a silicon chip design is going to cost you millions if not billions.

**Heidi Dempsey:** And speaking as a former civil engineer, if your bridge doesn't line up, that's even worse.

**Akash Srivastava:** Absolutely. That's why the human expert in the loop is going to remain a dominant

paradigm in those fields in the near future. The requirement for precision in these mission-critical domains is not something current Generative AI can match. One of our grants for the MIT-IBM lab is for precision generative modeling, which is aimed at pushing the boundary of precision in generating modeling to create engineering designs that are so precise, they can be sent directly to the machine shop or a 3D printer.

## EXCITEMENT MAKES AN IMPACT

**Heidi Dempsey:** Going backwards a little bit, how did you get into AI and computer engineering? Were you a math nerd, or did you take your toys apart?

**Akash Srivastava:** In elementary school I became fascinated with the idea of connecting human brains. What would happen? My dad got me this book by Ray Kurzweil, *The Singularity is Near*. I didn't understand 90% of it because I was very young, but my dad and my sisters helped me make sense of it. After reading the book, I knew I needed to study this.

At the University of Sheffield, where I ended up, this particular degree (BSc in AI and Como Sci) was new, and people were still figuring out what it should include, so I did computer science, math courses, psychology courses, a bit of probability theory, chaos theory. For my Master's and PhD, I was at the University of Edinburgh, which is amazingly good for machine learning. Thomas Bayes, who came up with Bayesian theory, and Geoffrey Hinton, who pioneered deep learning, both went there. By far the best time of my life was doing my PhD.

That's the  
ultimate level of  
democratization,  
when LLMs become  
useful for everybody.

Being at Red Hat, we can not only do this kind of work and publish papers about it but also put our code and model out there and allow the entire community of makers, coders, and students to iterate upon it and make it better.

**Heidi Dempsey:** It's the life of the mind, right? When you're walking around thinking about your problem five or six levels deep while you're just eating a sandwich. So you used up all the knowledge in the UK, then you came to the United States and MIT. What was the difference?

**Akash Srivastava:** When I came over to the US and the MIT-IBM lab, they kept that university environment very intact. The best part was working with people—especially students—who are better than you. You question yourself: "Why are you working with me again?" Now, at Red Hat, my team actively engages with PhD students who are excited to work with us on the cutting edge to help the mission of democratizing AI.

**Heidi Dempsey:** Eventually, you have to move into industry because you decide not to be a professor. And you want to preserve some of that excitement and curiosity, but industry is trying to make money. So how do you walk that line of maintaining the adventuring spirit of research while delivering something for the bottom line?

**Akash Srivastava:** I was lucky because my PhD was in generative modeling, which at the time meant you could throw a stone and hit a job offer, and that job often was pure research. When I joined the MIT-IBM lab, I don't think I ever felt like I had a real job because there was no pressure other than the normal conference and paper deadlines.

But at some point I started questioning my impact. Okay, I'm producing papers, which makes an

incremental difference, but how many lives am I actually impacting? The question for me and a lot of people on my team was, how do we get to a point where what we do helps more people than just experts in our domain? The solution was very natural. This technology we just happened to have studied is transforming people's lives. That's how we pivoted into language modeling and figuring out how to make a good language model. Everybody was struggling: typically research is all out there but in the field of LLMs people would not publish details, especially right after ChatGPT came out. So that became our mission.

My team was at NeurIPS in New Orleans, and I was at a workshop on multilingual models. It was a completely unrelated topic, but there was a picture of a taxonomy, and it just clicked. I ran and found the guys on my team and we made a bet that this was how you could synthetically generate the data for the alignment problem in language models. You generate data using a taxonomy, and you can define what goes in your model. These guys, I am not kidding, in three days they were able to prototype this thing—during a conference where half the time you're looking at posters and half the time you're tipsy from all the parties they have.

**Heidi Dempsey:** So that turned into the taxonomy for [Granite](#)?

**Akash Srivastava:** Yes, that's the basis for [LAB](#) (Large-scale Alignment for chatBots), which is the basis for InstructLab and how Granite models

are aligned. After an intense frenzy of work, we showed it to my then-boss and we were blown away. We were beating Meta's [Llama](#), and we went from being super underdogs to beating the best models at that time.

**Heidi Dempsey:** That's really cool. But still: when you get to industry and you have a roadmap and things you have to deliver on a certain schedule, how do you retain and encourage creativity and the willingness to take risks? It sounds like at the MIT-IBM lab you could do that because the emphasis is on the research part, but when you came to Red Hat the emphasis was more on the product.

**Akash Srivastava:** I think it's a two-part thing. First, the team really matters. I always say whenever I'm hiring, I can compromise on your degree or your expertise, but I will not compromise on your excitement. If discussing ideas, implementing them, and doing research doesn't give you the joy it gives the rest of the team, you're just going to feel left out.

Part two is articulating very clearly how everybody is making an impact. The simplest way to understand my team is that when they wake up, they need to beat something to feel like they won the day. And they need a clear line of sight as to how the company will benefit from it. If they feel like a little cog in some big machinery, they get bored. In fact, they come to me and tell me off—there's no filter. "You explain to me right now, how is this thing helping the business? We joined this thing because we wanted to make an impact, and I don't know

how I'm making an impact." I think for researchers in most places, the line of sight as to why you're doing something is never made clear. Our team doesn't have this problem, and it's such a refreshing change.

---

By the time you're reading  
this, we should have put out  
our work detailing the new  
state-of-the-art inference  
scaling technique.

---

**Heidi Dempsey:** We see the same thing in Red Hat Research, with the excitement about measuring and analyzing stuff. I had a team that changed from using small memory maps to big memory maps and they were really excited to see the flame graph of the performance of the CPU and GPU when they're running certain programs for memory access. It wasn't a significant impact on the product at that point; it was just, "Wow, look how much of a measurable change we made with this one thing."

So what are you and your team fired up about now?

**Akash Srivastava:** Right now InstructLab is the only example in the market of an end-to-end LLM customization solution, so we want to continue our efforts on the research and development side to keep it in the number one spot this year too.

Everyday with our collaborators at MIT, other academic partners, and IBM, we're working on the third generation of synthetic data generation and model alignment techniques. Being at Red Hat, we can not only do this kind of work and publish papers about it but also put our code and model out there and allow the entire community of makers, coders, and students to iterate upon it and make it better.

Every year in my team we set a grand challenge. Last year it was scaling small models via data, and we invented InstructLab as a result. This year we've taken up the challenge of adding inference/test time scaling tools to our offering. By the time you're reading this, we should have put out our [work detailing the new state-of-the-art inference scaling technique](#). This is bleeding-edge research in the Generative AI domain, and it's very strategic to our business. With Neural Magic joining us this quarter, we have an opportunity to establish ourselves as the leader in inference scaling techniques for small language models.

## BREAKING INTO AI

**Heidi Dempsey:** That's very cool. So let's talk about the flipside of that dynamic. There's somebody sitting in a group somewhere who wants to do something not LLM-related. They can't get any funding and their creativity is being suppressed because industry is so focused on solving everything with LLMs right now.

**Akash Srivastava:** This is a problem everywhere, not just in industry but in academia or in getting funding for



a startup. I always have to look for resources, and it's so much easier if there is an LLM or Generative AI use case attached. This is a game that in academia we play very well. Understanding the broader impact of a particular technology is always very helpful when it comes to writing grants or pitching ideas.

It's also that 80/20 thing, right? I'm happy to spend 20 percent of my own time to prove value for that other thing I'm convinced will have a benefit. We're researchers, and we should be looking two, three, or more years down the road and preparing for that. In the meantime, make yourself productive and learn the business, learn the tools. Pivoting into Generative AI is not hard. Imagine this is the first year of your PhD. How do you learn? Work with senior people as the fourth or fifth person on their paper and learn the techniques.

**Heidi Dempsey:** So you have to do the same thing with your new idea: find your group of collaborators and do your proof of concept. We used to do that upstream—Upstream First, right? But Upstream First with AI models is just not feasible.

**Akash Srivastava:** Upstream First requires fairly rigorous software engineering practices, but chances are, an average researcher, like some of my PhD students and some of the researchers on the team, might not have done a single pull request in their lives. It's like asking a software engineer who's trying to get into AI research to start by composing a well-written research paper. These

are different workflows and skill sets from different domains.

Neither should change their workflow; it's what makes them productive at their respective jobs. Research code will be dirty code, or at least not at production level. But—this is your way in. If you have a cool idea, go offer help. Working together requires some adjustments on both sides, and Red Hat is a great place for that.

---

Working together requires  
some adjustments on both  
sides, and Red Hat is a great  
place for that.

---

**Heidi Dempsey:** Let's take that issue of engineers working with non-specialists back to InstructLab. At first, there was a lot of talk about synthetic data and having a teacher model and a critic model. It doesn't seem like that's caught on as much as the rest of InstructLab. Do you have theories about why?

**Akash Srivastava:** This is a very interesting question. InstructLab, at a high level, is a tool for customization of language models. It's a prescriptive method, where instead of Red Hat deciding what goes in your model, you make a list of things you want your model to know (knowledge) and a list of things you want your model to be able to do (skills). The machinery takes your prescription and converts

it into some data, then we take that data and train the model. In many ways, the secret sauce is synthetic data generation. If you look across the industry, everybody trying to offer a customization toolkit is not giving you any way of generating the data. And buying it is super expensive. So one of the biggest problems InstructLab is solving for many users is generating data.

**Heidi Dempsey:** They also have the domain knowledge—that's why they're coming to you.

**Akash Srivastava:** And this is why I liked your previous question, because it gets to what this tool is. It's a customization toolkit. That's important because there's nothing else right now available from any other company. That doesn't exist except for InstructLab or RHEL AI or OpenShift AI.

**Heidi Dempsey:** Very cool. That's a lot like us in research. We're working with pathologists and biologists and other researchers in the same way. And I love that outlook because it's concentrating on the things your users do know and what they can contribute to the eventual model that's going to solve the problem.

**Akash Srivastava:** I like that, because to me that's the Red Hat way. And tell them the technical things so they will come back and contribute.

**Heidi Dempsey:** Thank you very much for such a lively conversation.

**Akash Srivastava:** Thank you—that was really fun! 



*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.**



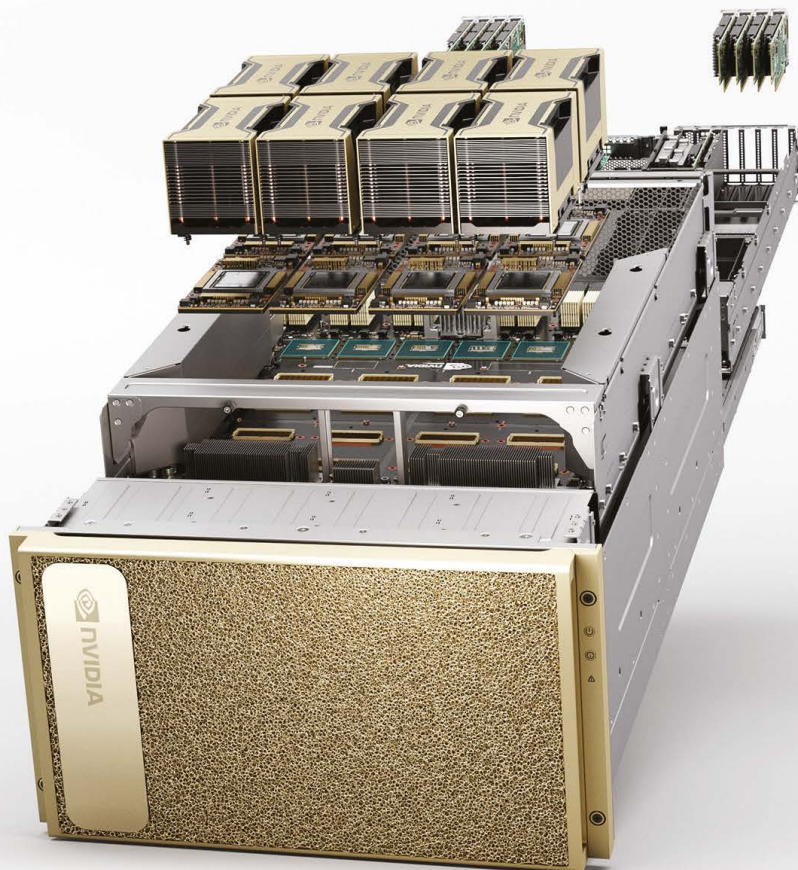
@mass-open-cloud



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



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



# 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)



# Smarter AI, fewer resources: bringing cloud AI into real-time edge devices to unlock performance

A new AI framework for edge systems overcomes the communication and energy obstacles that limit their use in real-time applications by integrating local and cloud decision-making while maintaining strong performance.

by Eshed Ohn-Bar

**A**rtificial intelligence (AI) models with vast and generalized knowledge are increasingly being integrated into everyday devices, from smartphones that provide personalized assistance to mobile robots and vehicles that continuously monitor and interact with their surroundings. Yet these powerful AI models are currently constrained by the limited resources of these edge devices.

Running a large and accurate AI model on a smartphone or a mobile robot can quickly drain its battery within minutes and require significant energy and hardware resources. As these models continue to grow in size and computational demands (e.g., requiring expensive GPUs), deploying them across millions of everyday devices becomes increasingly difficult, expensive, and environmentally unsustainable. As part of the collaborative project [Minimal](#)

[Mobile Systems via Cloud-based Adaptive Task Processing](#), researchers at Red Hat and Boston University developed a new framework that optimizes computation to enable more efficient real-time AI applications without sacrificing model accuracy.

## MOTIVATION

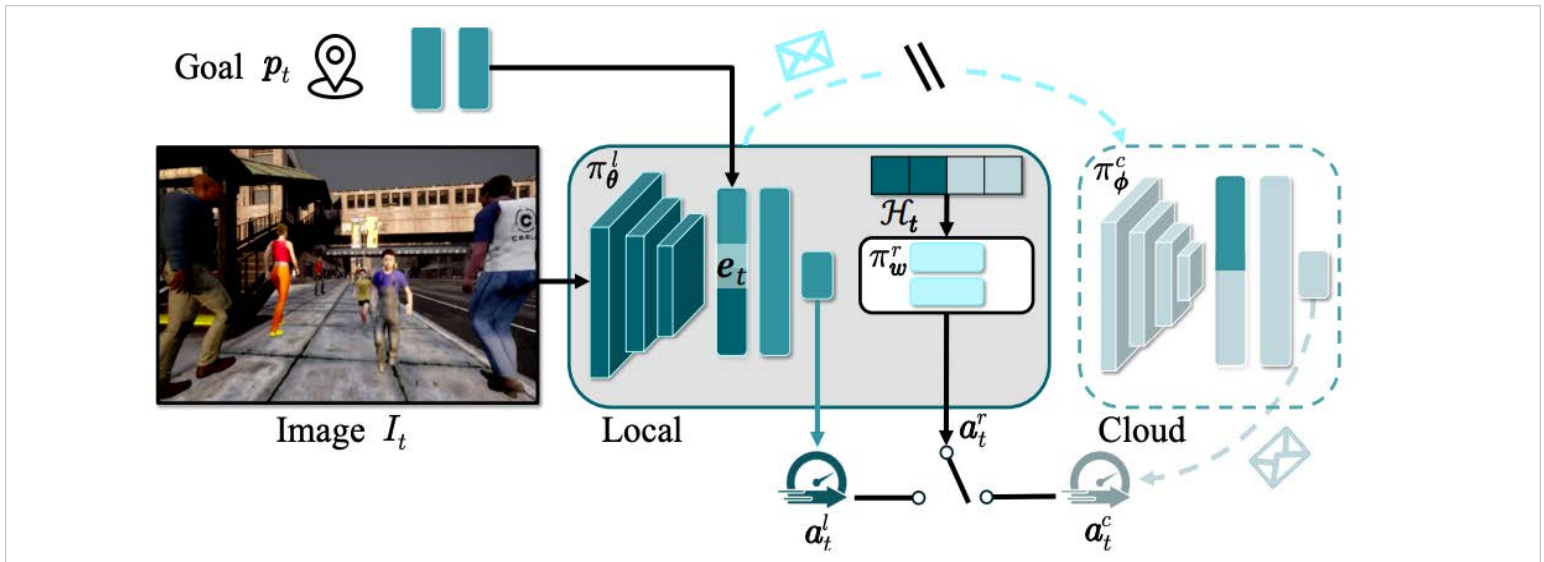
Traditionally, AI computations are offloaded to remote servers. This can save on-device resources, as local image and text data are sent to models in the cloud. Smart assistants often use this approach to offload as much computation as possible to the cloud, helping to preserve energy and local device resources. While this method is widely used today in systems like ChatGPT, relying on the cloud can introduce delays, making it unsuitable for real-time or safety-critical applications. For a robot, even a brief delay can be dangerous—for



## About the Author

### Eshed Ohn-Bar

Dr. Eshed Ohn-Bar, an Assistant Professor in the Electrical and Computer Engineering Department at Boston University, is passionate about building robust, efficient, and safe AI at scale.



**Figure 1.** Overview of UniLCD for a robot navigation task. The framework learns to offload tasks to the cloud while maintaining real-time performance.

example, causing a mobile system to collide with a nearby pedestrian. As a result, latency-constrained edge systems often depend on expensive local hardware and resources to ensure quick responses. Can we design edge systems that seamlessly balance cloud and local resources to optimize for real-time accuracy, efficiency, and safety across different situations?

To address urgent societal and sustainability needs with existing systems and models, engineers today may leverage various ad hoc strategies. Developers may try to use lightweight and compressed models, but these smaller models will suffer from degraded accuracy and result in unreliable performance, such as, again, failing to detect that nearby pedestrian. Models can also be carefully tuned for specific devices and scenarios but struggle when faced with diverse operational tasks

that may need more computational power. One promising alternative is systems that automatically adapt on the fly, adjusting when, where, and how computations are performed as needed.

In work presented at the European Conference on Computer Vision 2024, researchers from Red Hat and Boston University collaborated to develop a novel framework that dynamically learns to balance shared computation across various devices and operational settings. The proposed system, **UniLCD (Unified Local-Cloud Decision-Making)**, introduces a new approach based on a field in machine learning called reinforcement learning (RL), where the system learns by trial and error, receiving rewards or penalties based on its actions. This method trains a flexible model to decide, based on the current scenario and task, whether to offload computation to the cloud or process it locally.

### OUR METHOD—UNILCD

UniLCD is a dynamic approach that empowers resource-constrained devices—such as smartphones, autonomous vehicles, and mobile robots—with the ability to leverage both local processing power and cloud resources.

At its core, UniLCD comprises a context-dependent routing module, which takes as input an embedding, that is, a compressed representation of the current state and the history of past system decisions. This routing module is trained using RL to determine a decision policy, such as whether to implement a local action based on a lightweight but less accurate model or choose to transmit local information to the cloud server model, which is larger and more accurate but also induces latency. While this approach can be applied to any real-time AI application and edge device, **Figure 1** illustrates

an example system for a camera-based mobile robot navigation task.

The primary goal of our system is to learn when to offload computations to the cloud while meeting safety and real-time requirements. As shown in Figure 1, the local decision-making model (also referred to as the local policy) consists of a truncated neural network designed to rapidly process image and goal observations. The extracted features, or embedding, are then combined with a memory buffer that stores a history of past observations, providing additional context for the system. This historical data enables the system to observe latency dynamics and adapt to various constraints, such as limited communication settings. The memory is passed to a multi-layer perceptron (MLP) routing module, which determines whether to offload the current embedding to the cloud for further processing with a subsequent neural network or to classify a navigation action—such as steering, braking, or accelerating—locally. The complete algorithm for training the routing policy is shown in **Figure 2**.

As shown in the algorithm, UniLCD learns by receiving a reinforcement signal, or reward, based on the outcomes of its decisions. For example, a mobile system should learn to strategically interleave cloud computation, particularly when encountering challenging scenarios, to improve the accuracy of the lightweight, lower-accuracy local model. In the case of our navigation task, if the mobile robot successfully moves closer to the goal, reduces energy consumption, or selects effective action ranges and speeds,

#### Algorithm 1 UniLCD's Routing Policy Training with Reinforcement Learning

```

1: Input: Image  $\mathbf{I}$ , next waypoint  $\mathbf{p}$ , local policy  $\pi_{\theta}^l$ , cloud policy  $\pi_{\phi}^c$ 
2: Initialize: Number of iterations  $T$ , history  $\mathcal{H}$ , routing policy  $\pi_{\omega}^r$ , reply buffer  $\mathcal{S}$ 
3: Collect on policy samples:
4: for  $t = 1$  to  $T$  do
5:   Obtain local action  $\mathbf{a}_t^l$  and embeddings  $\mathbf{e}_t$  using local policy  $\pi_{\theta}^l(\mathbf{I}_t, \mathbf{p}_t)$ 
6:   Append  $(\mathbf{a}_t^l, 0)$  to history  $\mathcal{H}_t$ 
7:   if  $\pi_{\omega_t}^r(\mathcal{H}_t, \mathbf{e}_t) = 0$  then  $\mathbf{a}_t = \mathbf{a}_t^l$ 
8:   else
9:     Send  $\mathbf{e}_t$  to cloud,  $\mathbf{a}_t = \pi_{\phi}^c(\mathbf{I}_t, \mathbf{p}_t)$ 
10:    Update last value of  $\mathcal{H}_t$  to  $(\mathbf{a}_t, 1)$ 
11:   end if
12:   Compute instant reward using Eq. (2)
13:   if Arrived destination then break
14:   end if
15:   Update replay buffer  $\mathcal{S} = \mathcal{S} \cup \{\mathbf{I}_t, \mathbf{p}_t, \mathcal{H}_t, r_t\}$ 
16:   Update routing policy parameters with PPO
17: end for

```

**Figure 2.** Training a generalized routing policy with reinforcement learning. The algorithm continuously updates a minimal local neural network that classifies between local and cloud operations.

it gets a positive reward. If it is close to collision with an object, which is undesirable, it receives a negative reward, where the complete reward in each time step is computed as:

$$r = (r_{goal} \cdot r_{speed} \cdot r_{energy} \cdot r_{action})^{\alpha} - r_{collision}$$

Here, alpha is a scaling factor that adjusts the overall reward to fall within the range  $[0, 1]$ . This reward ensures that the resulting policy optimizes both task performance as well as energy and communication constraints. In general, designing a multi-objective reward function can be complex, even for relatively simple tasks (e.g., robot navigation without dynamic objects, as often explored in prior work). RL typically requires extensive iteration in training. One key finding is in how the reward function impacts training efficiency

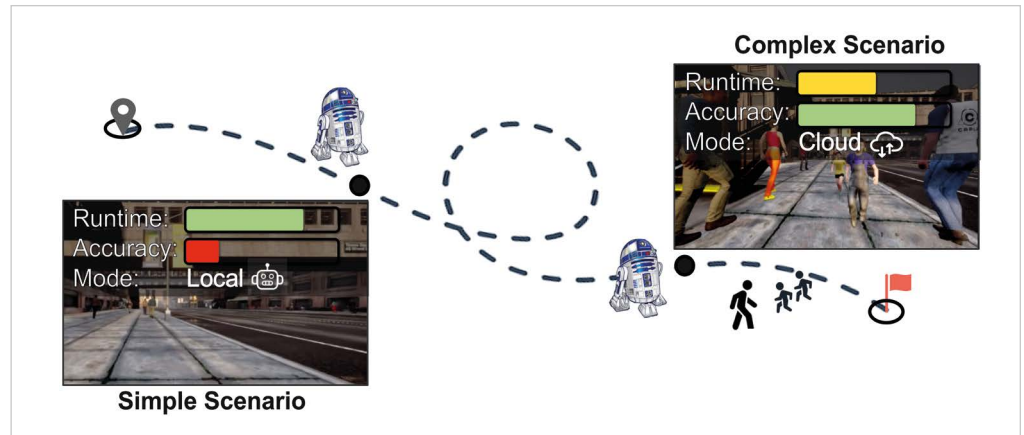
and convergence significantly. By multiplying the different reward terms, the need for extensive tuning of individual components is reduced—if one term is low, it diminishes the overall reward, encouraging an effective policy to emerge within just a few minutes of operation. Once this initial training is complete, the policy can be deployed without additional training, though the model can be updated over incoming observations continually (e.g., for further efficiency gains) or automatically adapt to novel scenarios, platforms, and communication modes.

## RESULTS

To rigorously validate the system, a simulation environment was developed for sidewalk robot navigation in crowded outdoor settings. This environment captures



UniLCD has the potential to reshape the future of edge computing by seamlessly integrating local and cloud-based decision-making.



*A hypothetical robot moves between local and cloud-based decision-making depending on the complexity of the situation and the level of accuracy demanded. In the complex scenario, higher accuracy is required to avoid collision with pedestrians.*

complex scenarios that require frequent switching and high responsiveness, thus showcasing UniLCD's robust capabilities in handling challenging, dynamic tasks that demand seamless cloud-edge integration. To realistically model real-world constraints, the simulation also introduces stochastic delays in data transmission between the local device and the cloud server, effectively capturing the impact of latency.


In the most difficult and dense settings, UniLCD showed an improvement of over 35% compared to all prior baselines in an introduced Ecological Navigation Score, a metric that combines task performance (e.g., collisions, route completion, overall task time) with overall energy costs. In these intricate settings, baselines relying on naive model splitting or pruning resulted in poor navigation and frequent collisions as their design does not holistically consider environmental, communication, and safety contexts.

The strong performance persisted across environmental conditions and different models, including very small local models for resource-limited use cases. This remarkable generalizability marks a significant step toward broad, ultra-low-cost deployments, which are currently being explored in follow-up research. Real-time, cloud-integrated systems with lightweight local models and minimal hardware requirements—such as smartphones—could be deployed in broader and more diverse settings, delivering high-performance operation with minimal degradation.

### **FUTURE APPLICATIONS**

UniLCD has the potential to reshape the future of edge computing by seamlessly integrating local and cloud-based decision-making. This novel framework is currently being integrated into Red Hat OpenShift, providing a flexible solution for enabling large-scale, real-world deployments across various communication and modeling configurations. While challenges

remain, including accelerating RL model training to solve for an optimal local-cloud policy within just a handful of interactions, there are several exciting future opportunities. Given the generalized nature of the routing mechanism, a potential approach to speeding up training further could be collaborative training over data from different platforms and tasks.

By significantly reducing the energy consumption and cost of powerful AI models, UniLCD could unlock transformative possibilities to address societal needs across a range of domains, including transportation, healthcare, and disaster response, where real-time and efficient processing is essential. For example, autonomous vehicles could offload tasks to cloud models to conserve energy and enhance safety. Lower-cost assistive robots could operate with precision and energy efficiency in various home environments, minimizing failures associated with low-accuracy edge models or delays from waiting for cloud-based predictions. In disaster zones, robots could manage resources efficiently, adapting to different communication infrastructures and operating for extended periods without sacrificing accuracy during the most crucial moments. Handheld smartphones could provide continual and reliable support when assisting users without rapidly depleting battery life. As researchers continue to push the boundaries of what's possible, UniLCD brings us one step closer to a future where smarter, faster, and more sustainable AI systems are seamlessly integrated into our daily lives. 

## NEVER MISS AN ISSUE!



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](https://red.ht/rhrq)

## Feature

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

is a Senior Performance Engineer at Red Hat working with networking and performance in general in telco 5G. She completed her PhD in computer science on the topic of improving multipath transport robustness with MPTCP in use cases ranging from 5G to the Internet. She also holds an adjunct senior lecturer position at Karlstad University, where she researches in the areas of network and system performance.

## Can LLMs facilitate network configuration?

The networks that connect everything from cell phones to datacenters require frequent—and error-prone—human intervention for configuration. Recent research evaluates the effectiveness of applying various machine-learning models to the task.

*by Simone Ferlin-Reiter*

Since 2023, Red Hat Research's collaborative project [Securing Enterprises via Machine-Learning-based Automation](#) (SEMLA), in partnership with the Kungliga Tekniska Högskolan (KTH Royal Institute of Technology) in Sweden, has been exploring the potential of large language models (LLMs) to address network configuration challenges—for example to make them less prone to human errors.

This work led to the development of the first model-agnostic network configuration benchmark for LLMs: NetConfEval, which examines the effectiveness of different models by translating high-level policies, requirements, and descriptions specified in natural language into low-level network configuration in Python. Having such a benchmark is crucial for tracking the fast-paced evolution of LLMs and their applicability for networking use cases, as done for other tasks. This article presents insights gained from this research so far and future directions we plan to take.

### **WHY IS NETWORK CONFIGURATION IMPORTANT?**

Networks are the backbone of today's communication infrastructure, powering everything from simple online interactions to mission-critical services. Network operators wield significant control over the flow of data in a network. These configurations—which can affect devices and services ranging from switches/routers, servers, network interfaces, network functions, and even GPU clusters—must be carefully configured to ensure the reliable transmission of information. Currently, network outages happen often, if not everyday. Network misconfiguration is among the common causes of unintentional outages, sometimes bringing down services for billions of users.

Although academia and industry widely adopted software-defined networking (SDN) to simplify network operation, network configuration still entails frequent human intervention, which is



**Network components:**

- 4 switches: s1, s2, s3, s4
- 2 end-hosts: h1, h2

**Requirement set:**

- All the switches can reach all the destination hosts.
- Traffic from s1 to h1 should travel across s2.
- The traffic from h1 to h2 is load balanced on 3 paths.

```
{
  "reachability": {
    "s1": ["h1", "h2"],
    "s2": ["h1", "h2"],
    "s3": ["h1", "h2"],
    "s4": ["h1", "h2"]
  },
  "waypoint": {
    ["s1", "h1"]: ["s2"]
  },
  "loadbalancing": {
    ["h1", "h2"]: 3
  }
}
```

**Figure 1.** High-level requirements translated into formal, structured, machine-readable specifications

costly and difficult. It requires expert developers who are familiar with large and complex software documentation and API interfaces, as well as knowledge about libraries, protocols, and their potential vulnerabilities.

There have been many efforts to simplify this process by compiling a high-level policy specified by a network operator into a set of per-device network configurations and to minimize errors by generating configurations with provable guarantees via verification. Nevertheless, network configuration remains an arduous, complex, and expensive task for network operators because they must acquire proficiency in a new domain-specific language that may not be widely used and could potentially have flaws.

**LEVERAGING LLMS FOR NETWORK CONFIGURATION**

While LLMs hold great potential for simplifying network configuration, there are a number of critical challenges that may hinder their widespread deployment. First, LLMs remain notoriously unreliable,

producing outputs that may be completely incorrect, often called hallucinations. Second, reducing inaccuracies produced by LLMs highly depends on the way the user prompts the LLM, a concept known as prompt engineering. Third, operating or using LLMs is expensive: the cost of training, like fine-tuning an LLM such as GPT-4, may quickly grow to millions of dollars.

In NetConfEval, we highlight the potential benefits of using Natural Language Processing (NLP) and LLMs to address the following networking problems:

1. Translating high-level requirements (expressed in natural language) into formal, structured, machine-readable specifications;
2. Translating high-level requirements into API/function calls, which is particularly interesting for SDN and automation protocols in modern network equipment;
3. Writing code to implement routing algorithms based on high-level descriptions;

4. Generating detailed, device-compatible configuration for various routing protocols.

In this article, I focus only on Task 1 to demonstrate NetConfEval. Use cases 2-4 can be found in the original paper, "[NetConfEval: Can LLMs facilitate network configuration?](#)" by KTH authors Changjie Wang, Mariano Scazzariello, Dejan Kostic, and Marco Chiesa, with Alireza Farshin (NVIDIA) and Simone Ferlin (Red Hat). The paper was awarded the 2025 Applied Networking Research prize at the Internet Research Task Force open meeting in Bangkok. We discuss various opportunities to simplify and potentially automate the configuration of network devices based on human language prompts/inputs.

As an example, **Figure 1** shows a sample input in high-level natural language, and its corresponding output in structured, low-level, formal language (Python).

Depending on the complexity of the network requirements and

policies, a network operator may directly add or remove new entries in the formal specification format, for example, to consider link preferences and/or resilience to more efficiently configure the network.

We devised an experiment as follows:

1. Generate 3,200 network requirements focusing on reachability, waypoint, and load balancing, using Config2Spec<sup>1</sup> on a topology composed of 33 routers;
2. Randomly pick a certain number of requirements and slice them with various batch sizes<sup>2</sup>;
3. For each batch, convert them into the expected formal specification format using a Python script;
4. Transform them to natural language based on predefined templates;
5. Ask an LLM to translate these requirements from natural language to the formal specification; and
6. Evaluate the efficiency of different LLMs by comparing the translated version of formal specification with the expected one.

We evaluated different combinations of policies (e.g., Reachability, Reachability + Waypoint, and Reachability + Waypoint + Load Balancing). The batch size definition varies with the number of policies: for example, a batch size of 2 in the Reachability +

Waypoint scenario indicates that the batch contains a Reachability and a Waypoint specification.

In our analysis, we use various OpenAI (GPT-3.5-Turbo, GPT-4, and GPT-4-Turbo) and Meta CodeLlama (7B-instruct and 13B-instruct) models, also fine-tuning GPT-3.5-Turbo5 and CodeLlama-7B-Instruct models with OpenAI's dashboard and QLoRA. To this end, we created a dataset similar to the one used for the evaluation but with slightly different templates and then fine-tuned the models for three epochs.

**Figure 2** shows the results of our analysis. GPT-4 performs similarly to GPT-4-Turbo<sup>3</sup>. It is important to find the appropriate batch size when translating high-level requirements into a formal specification format. GPT-4-Turbo achieves higher accuracy than GPT-3.5-Turbo and CodeLlama.

The results of our analysis demonstrate that:

**Selecting the appropriate batch size is key for cost-effective and accurate translations.** Since each inference request should contain preliminary instructions within the prompts, batching the translations could reduce the per-translation cost (prompts are conversation-wide instructions to the LLMs). Our results show that the accuracy of translations is worsened with larger batch sizes (especially for non-GPT-4 models).

It is therefore important to carefully select a suitable batch size for each model to ensure the right trade-off between accuracy and cost. For instance, translating 20 requirements in one batch with GPT4-Turbo is around 10 times cheaper compared to translating 20 requirements one by one, while still achieving 100% accuracy (**Figures 2a** and **2d**).

#### **Context window matters.**

Translation accuracy decreases as we increase the batch size. We speculate that this reduction in accuracy may be related to reaching the context length (e.g., 4,096 maximum input/output tokens for all models except GPT-4-Turbo, which supports 128k input tokens). In most of the experiments, we noticed that the generated LLM outputs are always truncated when the batch size gets closer to 100.

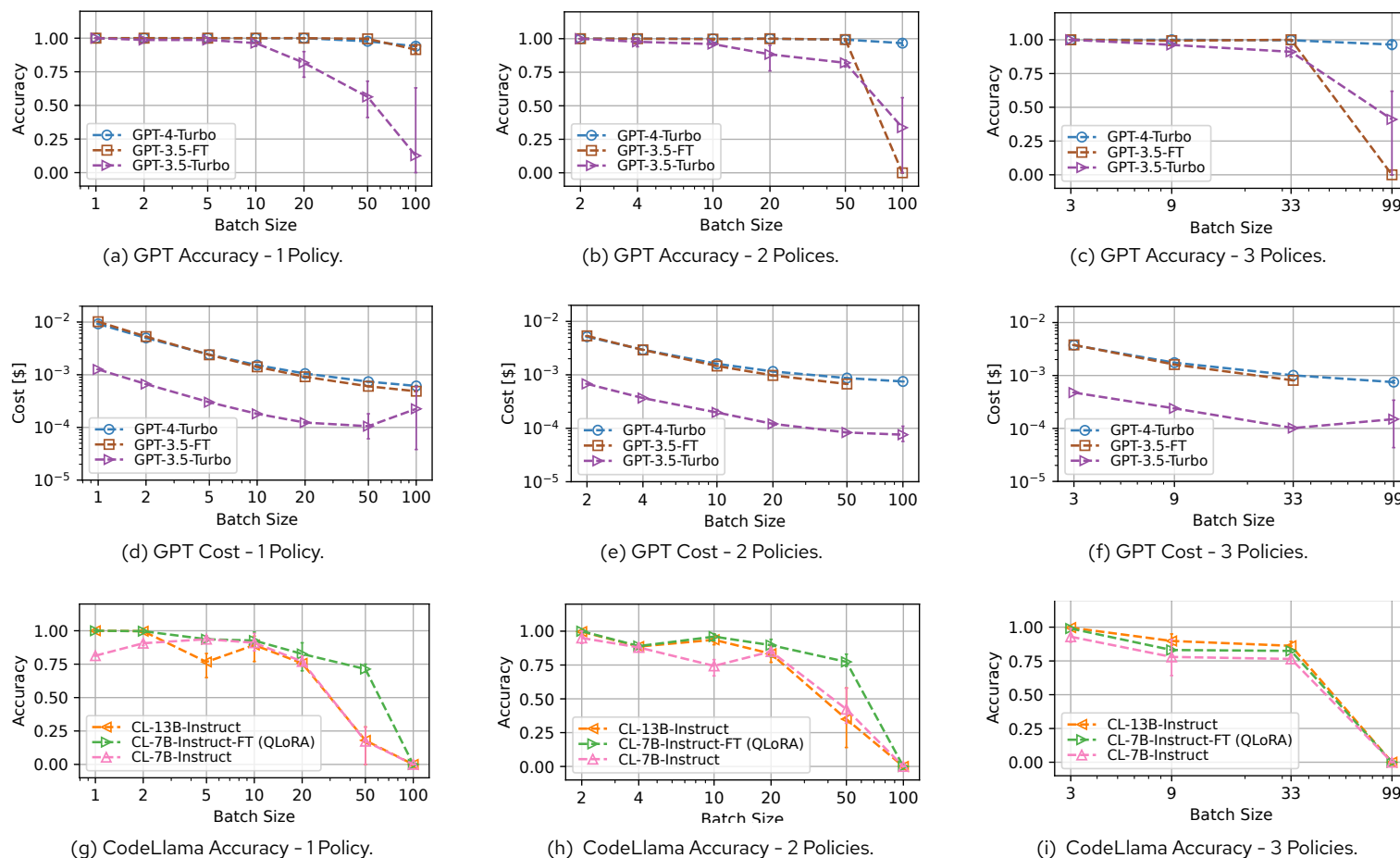
#### **Fine-tuning improves accuracy.**

Fine-tuning LLMs for a specific purpose could optimize their accuracy. While GPT-3.5-Turbo apparently performs worse than GPT-4-Turbo, **Figures 2a, 2b, and 2c** show that a fine-tuned version of GPT-3.5-Turbo achieves similar accuracy to GPT-4-Turbo, but with a higher cost, because OpenAI sets a higher per-token price for fine-tuned models. **Figures 2g, 2h, and 2i** show a similar takeaway for CodeLlama models, where fine-tuning the CodeLlama-7B-Instruct model using QLoRA can

<sup>1</sup> Rudiger Birkner, Dana Drachsler-Cohen, Laurent Vanbever, and Martin Vechev, "Config2Spec: Mining Network Specifications from Network Configurations." In (2020) 17th USENIX Symposium on Networked Systems Design and Implementation, USENIX Association, Santa Clara, CA, 969-84.

<sup>2</sup> We initialized the random function with a specific seed to ensure consistent results across various models.

<sup>3</sup> We do not show the results for better visibility in the figures.



**Figure 2.** It is important to find the appropriate batch size when translating high-level requirements into a formal specification format. GPT-4-Turbo achieved higher accuracy than GPT-3.5-Turbo and CodeLlama. We run CodeLlama on the Leonardo supercomputer equipped with NVIDIA custom Ampere GPU 64 GB.

achieve better accuracy than the original model and sometimes better than the 13B-Instruct model.

### GPT-4 beats the majority of existing models in our experiments.

GPT models generally achieve higher accuracy than their open source counterparts (e.g., CodeLlama). We also experimented with other open source models (e.g., Mistral-7B-Instruct and Llama-2-Chat) and Google Bard<sup>7</sup>, and they generated less accurate translations.

The ambiguity of human language and unfamiliarity with specific classes of problems may result in misinterpretations. Even when a single network operator is involved, contradictory network requirements can still occur, especially when the number of requirements is large.

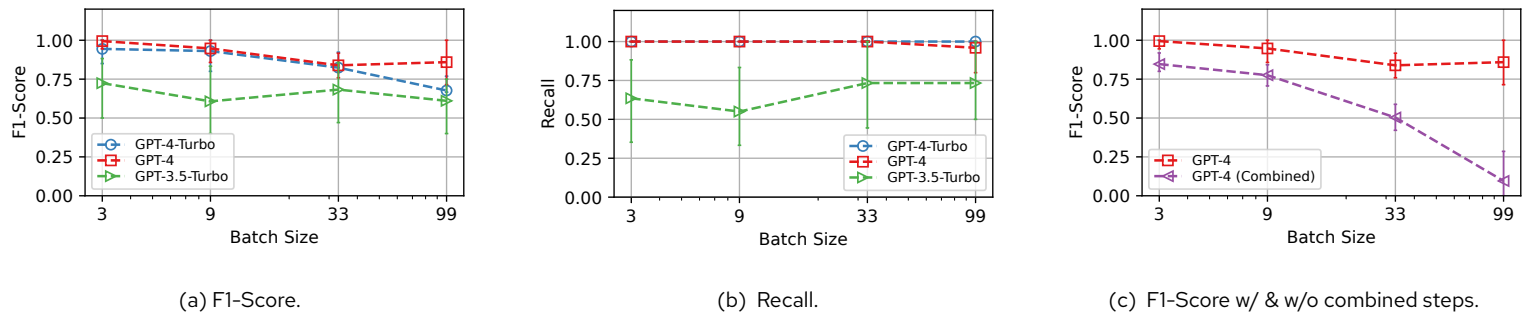
### Simple conflicts

A common case is two requirements that explicitly include contradictory information. For instance, a requirement specifies *s1* to reach *h2* while another

requirement prevents *s1* from reaching *h2*. To evaluate LLMs' performance in conflict detection, we designed a set of experiments where we randomly selected one requirement from each batch, generated a conflicting requirement (e.g., the conflicting requirement of "*h1* can reach *h2*" is "*h1* cannot reach *h2*"), and inserted them back into the batch.

We evaluate the effectiveness of LLMs in detecting simple conflicts in two scenarios:





**Figure 3.** GPT-4 can successfully detect simple conflicts in the provided high-level requirements.

- Detecting conflict as a separate step and explicitly asking an LLM to search for a conflict and report it
- Asking the LLM to perform conflict detection during the translation of requirements into a formal specification format, a scenario we refer to as Combined

**Figures 3a** and **3b** show the results of various GPT models when performing conflict detection. These results show that GPT-4 and GPT-4-Turbo reach almost 100% recall<sup>4</sup> for different numbers of input requirements. These results suggest that such models are always capable of detecting conflicts when a batch contains a conflicting requirement (i.e., they do not report a false negative). **Figure 3c** demonstrates that conflict detection is much more accurate when done in isolation. As opposed to GPT-4 models, our results demonstrate a poor recall and F1-Score for GPT-3.5-Turbo model.

In order to determine whether this performance degradation is related

to the smaller context window size of GPT-3.5-Turbo, we designed a new experiment to measure the impact of the position of a conflicting requirement in a batch: that is, to understand whether adding a conflicting requirement at the beginning, middle, or at the end could affect the accuracy of conflict detection. More specifically, we select a few batches with 33 requirements. For each requirement in the batch, we iterated through all the possible positions (indices), where we could insert a conflicting requirement.

**Figure 4** shows the number of conflicts detected out of 10 runs. One can observe that GPT-3.5-Turbo may be better at detecting conflicting requirements at the end of the batch: see the relatively darker squares at the hypotenuse of the heatmap. Finally, we compare the performance of GPT-4 when performing conflict detection separately and combined with translation (see Figure 3c).

### Complex conflicts

An example of such conflicts is when a requirement specifies s1 to reach h2 through s2, while another requirement prevents s2 from reaching h2. We observed that most of the time GPT-

4 translates these types of conflicts into Reachability and Waypoint specifications without reporting any conflicts, which is not desirable. To address this issue, we propose conducting intra-batch conflict detection before translating the requirements. If no conflict is identified within the batch, the translation results can be merged into the formal specification. Once the translation is completed, it is possible to use Satisfiability Modulo Theories (SMT) solvers to ensure there exists a solution for a given formal specification. In case of detecting any contradictions, an LLM can interpret them and provide feedback to network operators, which remains as our future work.

### TAKEAWAYS

Our micro-benchmarking can be summarized into the following principles that could help network developers design LLM-based systems for network configuration:

**Breaking tasks helps.** Comparing the accuracy of conflict detection when a) performed as a separate task and b) performed during translation, we observe that separating the conflict detection and translation

<sup>4</sup> The recall metric reports the ratio of true positives (i.e., true positives divided by the true positives and false negatives).

results in better accuracy (i.e., a higher F1-Score). This finding motivates the necessity of splitting complex tasks into multiple simpler steps and solving them separately.

#### Simple conflicts can be detected.

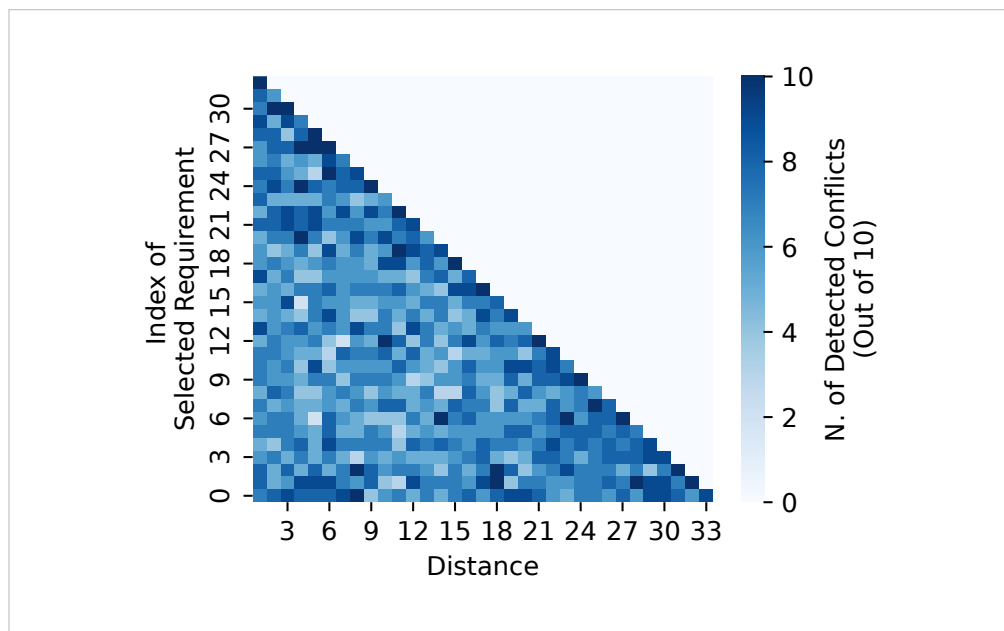
GPT-4 and GPT-4-Turbo models are capable of successfully detecting all those simple conflicting requirements we presented to them.

#### Detected conflicts could be false positives.

GPT-4 and GPT-4-Turbo sometimes report false positives (i.e., they detect a conflict when there is none). A concrete false positive example is "For traffic from Rotterdam to 100.0.4.0/24, it is required to pass through Basel, but also to be load-balanced across 3 paths which might not include Basel." LLMs tend to overinterpret the conflict by, for example, considering Load Balance conflicting with Waypoints. It is, however, possible to minimize false positives by providing examples for possible conflicts in the input prompts.

#### FUTURE WORK

Our main findings show that some LLMs are mature enough to automate simple interactions between users and network configuration systems. More specifically, GPT-4 exhibits extremely high levels of accuracy in translating human-language intents into formal specifications that can be fed into existing network configuration systems. Smaller models also exhibit good levels of accuracy, but only when these are fine-tuned on the specific tasks that need to be solved, thus requiring expertise in the specific tools and protocols that one expects to use.



**Figure 4.** The impact of distance on GPT-3.5-Turbo when detecting simple conflicts

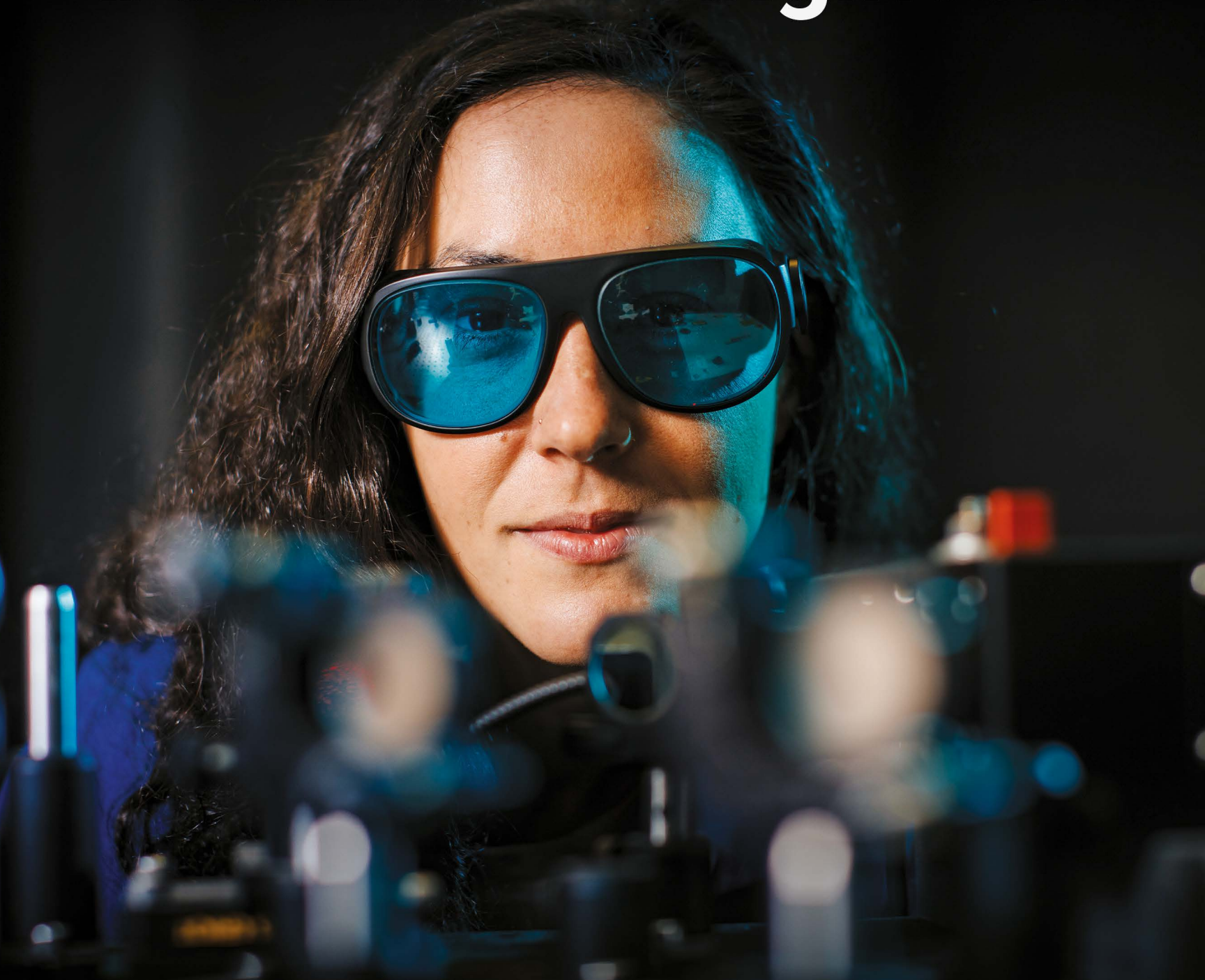
For instance, LLMs can potentially simplify the cumbersome task of managing Kubernetes-based clusters, as these get larger and more distributed, or simplify network troubleshooting tasks. We also observed that finding the correct prompts is challenging and highly affects the results. We confirm that techniques based on step refinement<sup>5</sup> are more effective also in tasks such as routing-based code generation. We observed that small models were ill-suited for code generation tasks, even those that were specifically fine-tuned on Python coding. We believe that fine-tuning models on network-related problems will not be sufficient, as network operators often need

to write new functionalities that cannot easily be envisioned when fine-tuning the model (e.g., writing code based on new ideas from scientific papers, RFCs, etc.).

We hope that our work with [NetConfEval](#) motivates more research on employing AI techniques on network management tasks. Future iterations of our benchmark could a) enhance complexity by incorporating additional policies, implementing more sophisticated and distributed routing algorithms, and creating advanced configuration generation tasks and b) explore the impact of different task decomposition strategies or applying LLMs in network policy mining. 

<sup>5</sup> Kumar Shridhar, Koustuv Sinha, Andrew Cohen, Tianlu Wang, Ping Yu, Ram Pasunuru, Mrinmaya Sachan, Jason Weston, and Asli Celikyilmaz, "The ART of LLM refinement: ask, refine, and trust." 2023, arXiv:2311.07961.

Be bold. Be boundless.  
**Be a Baskin Engineer.**



UC SANTA CRUZ |  Baskin  
Engineering

[engineering.ucsc.edu](http://engineering.ucsc.edu)



## Meet Perun: a performance analysis tool suite

How do you turn a research project into an industry tool? Learn how the creators of Perun built a better performance analysis toolkit then brought it from academia to real-world implementation.

*by Jiří Pavela, Tomáš Fiedor, Jiří Hladký, and Tomáš Vojnar*

Everyone has a horror story about poor performance in a continuously evolving product. Managing the performance of reasonably complex software is simply a difficult task. With large software systems, things get even more entangled. The Linux kernel, web browsers, operating systems, or databases—these code bases all exceed millions of lines of code. The Linux kernel, in particular, has been developed for several decades by tens of thousands of developers, and it has a constant stream of new kernel versions every day, both in the upstream and in the distribution-specific branches, such as Red Hat Enterprise Linux (RHEL).

This article introduces Perun, an open source, lightweight Performance Version System that tracks performance profiles corresponding to different versions of underlying projects. Perun began in 2016 as a project at the Brno University of Technology with a small team of researchers aimed at developing a performance analysis tool suite to help performance engineers and developers working on complex user-space software. In 2023, the BUT team joined forces

with Red Hat Research to enhance Perun with kernel-space analysis capabilities.

Perun is basically what every performance engineer needs in one place: tools for measuring metrics, storage of results, interpretations of performance data, and a link between performance results and different versions of a software project. Perun is designed to support a variety of architectures and workflows. In this article, we will describe how we worked with the Red Hat Kernel Performance Engineering Team in Brno, responsible for RHEL kernel performance, to develop Perun further. However, this is merely one use case, and we encourage others to [try Perun](#).

### WHY PERUN?

Whenever a new RHEL or ELN kernel version passes functional testing, the performance team has to evaluate its performance, and they have to do it quickly. The evaluation is based on running a set of benchmarking suites on new kernel versions and then analyzing potential performance changes compared to previous versions. The goal is to locate drops in performance and the root

Diff View Generated by Perun v0.23.7			
2025-03-05 08:59:08 UTC			
Baseline (base)		Target (tgt)	
[-]	Profile Specification	[-]	Profile Specification
boot info?	BOOT_IMAGE=(hd0,gpt2)/vmlinuz-5.14.0-427.13.1.el9_4.x86_64 root=UUID=84fc5815-cf15-427c-8b38-d5d3c795bf6f ro resume=UUID=d10b0089-fa81-4a40-9437-f450d99f7b26 console=ttyS1,115200n81 crashkernel=1G-4G:384M,4G-16G:512M,16G-64G:1G,64G-128G:2G,128G-:4G	boot info?	BOOT_IMAGE=(hd0,gpt2)/vmlinuz-5.14.0-509.el9.x86_64 root=UUID=ec9af29f-3eb3-4913-8c26-bb471faf21ef ro resume=UUID=b0925fd2-1ab2-4c80-8a66-c7bdae61b9a1 console=ttyS1,115200n81 crashkernel=1G-4G:384M,4G-16G:512M,16G-64G:1G,64G-128G:2G,128G-:4G
collector command?	kperf --repeat=1	collector command?	kperf --repeat=1
command?	stress-ng --mmapmany 1 --verbose --oomable --metrics-brief -t 23	command?	stress-ng --mmapmany 1 --verbose --oomable --metrics-brief -t 23
cpu (total)?	192	cpu (total)?	192
exitcode?	0	exitcode?	0
host?	intel-emerald-rapids-platinum-8558-2s.lab.eng.brq2.redhat.com	host?	intel-emerald-rapids-platinum-8558-2s.lab.eng.brq2.redhat.com
kernel?	5.14.0-427.13.1.el9_4.x86_64	kernel?	5.14.0-509.el9.x86_64
memory (total)?	499GiB	memory (total)?	499GiB
origin?	HEAD	origin?	HEAD
vulnerabilities?	[+]	vulnerabilities?	[+]
[-]	Profile Stats	[-]	Profile Stats
Maximum Trace Length [#] (value)?	27	Maximum Trace Length [#] (value)?	34
Overall Inclusive Samples [#] (value)?	94937660971	Overall Inclusive Samples [#] (value)?	95644110420
[-]	Profile Metadata	[-]	Profile Metadata
gcc?	v14.2.0	gcc?	v14.2.1

**Figure 1.** General context information about the baseline and target profiles being compared. This includes selected kernel and hardware properties, user-defined statistics, and/or metadata.

causes of these drops. Much of this process involves manual inspection and comparison of performance data, metrics, statistics, and other reports (e.g., the flame graphs invented and popularized by [Brendan Gregg](#)).

In many cases, performance engineers have to inspect the environment as well: the boot, system, or hardware configurations—such as the set of enabled mitigations for CPU vulnerabilities—which are often scattered in different reports and logs. This manual process is tedious and time consuming, and it cannot be easily

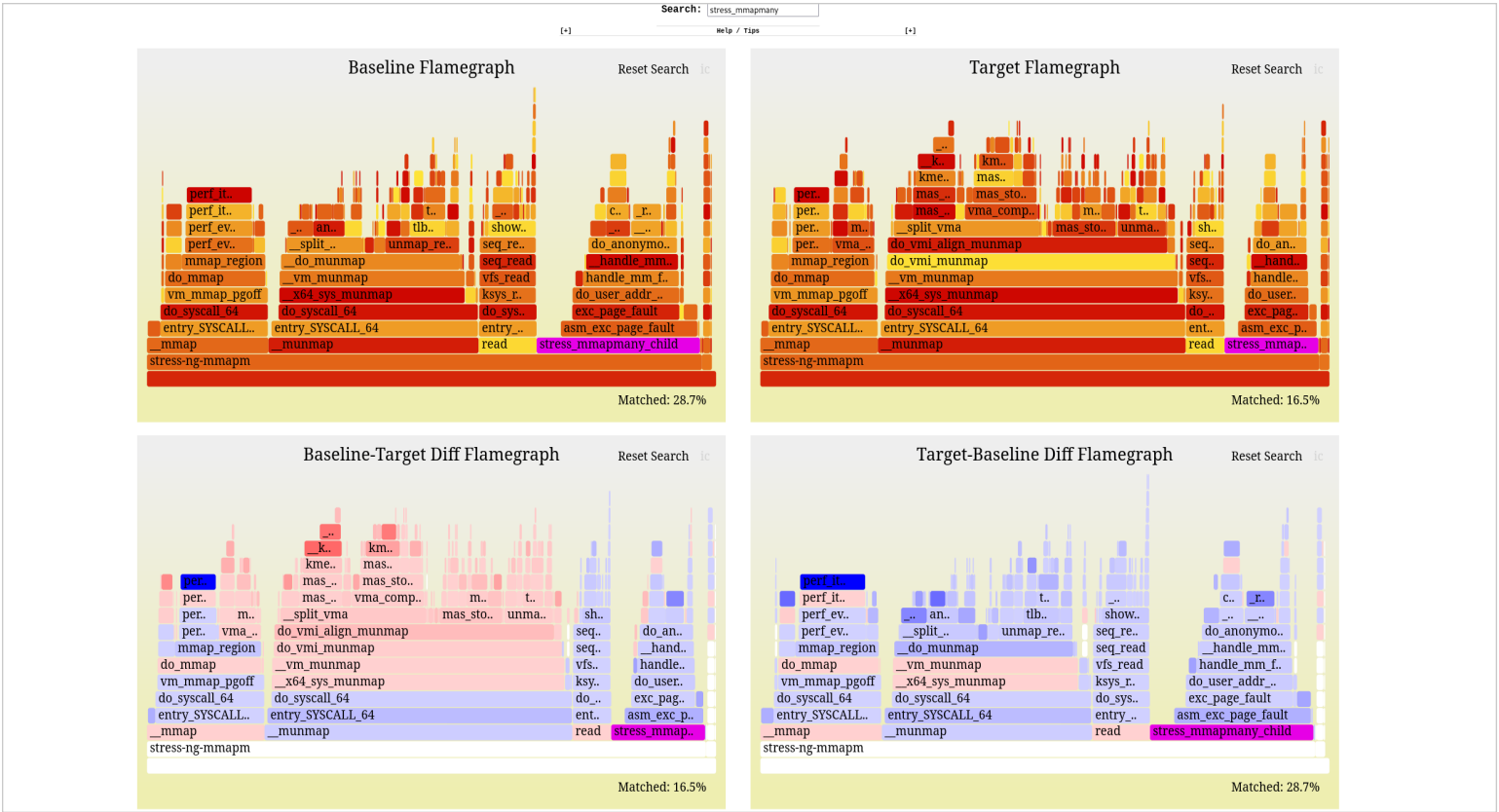
automated. It requires a performance engineer’s domain knowledge and a deep understanding of the kernel. Extending the performance engineer’s toolbelt is undoubtedly welcome.

### PERUN’S ARCHITECTURE

Perun’s core acts as a lightweight database that stores compressed performance results and maintains their link to concrete function changes (represented as software versions—for Git, this corresponds to pairs of a branch and a commit) identified by hashes. Technically, this is realized similarly to how it works

in Git: Perun resides in a parallel file system next to the version control system. Moreover, Perun provides a suite of tools (both experimental as well as wrappers over existing tools) to help with collecting, managing, and interpreting performance data.

For kernel analysis in particular, we have implemented two profilers. **kperf** is based on the well-established **perf** tool that samples the kernel stack together with traces. Ktrace uses the **libbpf** library to trace the kernel function executions precisely. (Note that this is highly experimental, as



**Figure 2.** A grid of interactive flame graphs and flame graph differences between the baseline and target profile. This allows the engineers to spot significant performance differences at a quick glance or drill down into suspicious functions.

tracing every function call in the kernel provides both considerable overhead and potential event loss due to the vast number of traced events.)

However, Perun offers much more: its architecture is modular and can be extended with new experimental tools quite easily. In past years, it has served as a platform for experimenting with other performance aspects, including performance fuzz-testing, non-parametric performance models, performance regression detection techniques, and an energy-consumption profiler, among many

others. Perun is implemented in Python; it is still in development and supports Python versions ranging from 3.9 to 3.13. You can install Perun from the PyPI repository using `pip install perun-toolsuite`.

### THE RED HAT USE CASE

For the Red Hat team, we integrated Perun as part of the benchmarking toolchain mainly to generate reports highlighting differences between different performance profiles of different kernel versions. Our goal was to create self-contained reports with intuitive, interactive, and compact

visualizations that make it easier to interpret performance results and quickly drill down into functions with suspicious performance behavior.

**Figures 1 to 4** show the four main parts of the Perun report.

Perun displays general context information about the baseline and target profiles being compared. It is also possible to provide user-defined metrics that will be compared here (**Figure 1**). Next, interactive flame graphs and flame graph differences are displayed (**Figure 2**). Additional details are available in a tabular view



[+]

Help / Tips

[+]

10 ▾

entries per page

Search:

Unit	Change	Metric	Absolute Difference	Relative Difference
[+] do_vmi_munmap	not in baseline	Inclusive Samples [#]	48,693,053,021	100.00%
[+] do_vmi_align_munmap	not in baseline	Inclusive Samples [#]	47,048,976,257	100.00%
[+] __x64_sys_munmap	in both	Inclusive Samples [#]	18,027,169,545	36.61%
[+] __vm_munmap	in both	Inclusive Samples [#]	18,011,497,880	36.63%
[+] __munmap	in both	Inclusive Samples [#]	16,382,607,421	31.52%
[+] mas_store_prealloc	not in baseline	Inclusive Samples [#]	14,321,848,146	100.00%
[+] do_syscall_64	in both	Inclusive Samples [#]	14,107,001,762	18.34%
[+] entry_SYSCALL_64	in both	Inclusive Samples [#]	13,416,292,669	17.25%
[+] vma_complete	not in baseline	Inclusive Samples [#]	12,810,324,602	100.00%
[+] __split_vma	in both	Inclusive Samples [#]	11,826,744,727	48.22%

Showing 1 to 10 of 871 entries

«

<

1

2

3

4

5

...

88

>

»

**Figure 3.** An interactive table that aggregates measured data on a per-function basis to view the total performance change across different calling contexts. The table also allows the exploration and traversal of the most expensive traces for each function.

under the Browse All tab (**Figure 3**). At the bottom, an interactive Sankey graph is rendered, which allows the user to further explore and filter the call stack and see the differences in the number of collected baseline and target samples (**Figure 4**).

The integration of Perun in the kernel performance process gave us encouraging results: Perun has already helped the Kernel Performance Engineering Team pinpoint the source of several performance results. The root causes are rich: excessive calls to XFS file system functions, needless calls to SELinux policy functions (see **Figure 5**), nonoptimally designed barriers, or inefficient mitigations of some recent kernel vulnerabilities.

Figure 5 illustrates that the function has been seen in 34% more call

stack samples in the target profile compared to the baseline. Moreover, the number of exclusive (also known as self) samples is 127% higher in the target, as indicated by the red exclamation marks (not part of the Perun visualization). This means that the `selinux_socket_sendmsg` function likely spends more time directly in its code, as opposed to other functions called from it. In this particular case, it thus becomes a candidate for further analysis by performance engineers.

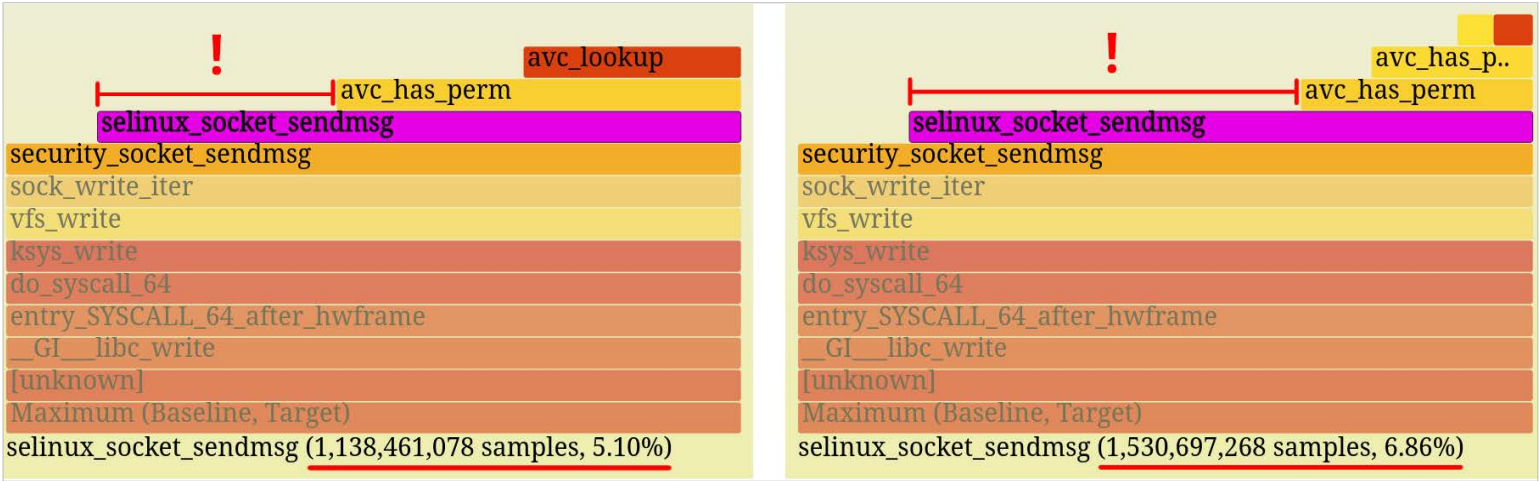
Overall, Perun makes it easier and faster for performance engineers to identify the source of performance degradations. Based on the Kernel Performance Engineering Team’s experience using Perun so far, we estimate that, on average, Perun saves approximately four hours out of a typical 8-hour process to

examine and triage new performance bugs thanks to the newly introduced automation of generating more verbose and self-contained difference reports. Moreover, engineers can more precisely locate the source of performance drops in more cases than before, giving kernel developers more context information and fixing more performance bugs.

In addition, Perun’s self-contained and highly interactive reports make exchanging results easy. This is especially important in global companies where engineers work in different time zones. Last but not least, with Perun, engineers no longer have to wait for the hardware to become available and can instead analyze performance data offline. Performance engineers have welcomed the new addition, and we are extending the set of



**Figure 4.** An interactive Sankey graph that aggregates traces into a single visualization, which allows the engineers to traverse and explore individual function traces and see how the performance changed in each function within the call chain.



**Figure 5.** An example of a performance degradation in the `selinux_socket_sendmsg` function.

Perun features based on their feedback to help them even more.

We helped uncover some performance issues and changes at Red Hat, including changes in network performance between early versions of RHEL-10 and

RHEL-9.4. Since RHEL-10 uses kernel 6.12 and RHEL-9 kernel 5.14, reviewing changes in kernel code is virtually impossible. Perun's intuitive visualization uncovered a 15% slowdown caused by Intel BHI mitigation influencing SELinux handling.

After the kernel developers resolved this problem, we started to uncover more subtle changes. The first was newly introduced synchronization barriers in the code, which had a lot of potential for performance improvements and allowed performance gains

in the 5-10% range. Finally, we found performance issues in low-level assembler functions on some platforms that copy data between buffers. Since then, we have provided Perun reports along with every kernel test result. This helped communicate results to the developers and made the reported performance bottlenecks obvious.

### INTEGRATING PERUN

To integrate Perun into Red Hat processes, we separated the data collection from post-processing. We have a dedicated server running Perun and integrating results collected from individual test machines. Systems under test run a different kernel and/or have different configurations and environments. In our initial setup, we deployed Perun to each new machine and used our dedicated kernel profilers (based on **perf** or BPF). The test environment spans different OS versions, from RHEL-8 to RHEL-10 up to CentOS Stream, and maintaining Perun instances with respect to ever-changing machine dependencies and requirements was not feasible. For this reason, we now instead rely solely on remote profiling using **perf** (which itself is widely supported by kernels) and import the raw data into Perun on our dedicated server.

The dedicated server with Perun's instance maintains a database of these results, preserving the context of profiling: the actual data, selected statistics, environment, and machine specification, list of vulnerabilities, and other things necessary to debug the root cause of performance degradation.

The server then computes a comparison of two selected kernels on demand, generating our proposed performance reports. The functionality is available from the internal dashboards. This lowers the barrier to adopting Perun by offering the following alternative lightweight workflow, where Perun acts as a front-end for performance data collected by **perf** with additional environment context:

1. As soon as the new version of the kernel passes functional tests, performance tests start automatically on a wide range of different hardware configurations. For some tests, performance data are automatically collected. For others, performance data can be collected on demand. The limiting factors are the runtime, **perf** tool overhead, and disk space required to store all **perf** profiles.
2. Quick statistical performance checks are performed, along with automatic machine-learning-enhanced reports tagging, to assess whether performance is degraded for any pair of kernels or configurations.
3. A performance engineer either reviews the generated Perun reports, if they are already available (only for selected benchmarks), or schedules specific benchmark runs with **perf** data collection enabled that will automatically generate the required Perun reports. The engineer examines the reports to identify potential sources of the reported degradation. Flame graphs are handy for gaining a quick overview of performance changes.

4. Once a potentially problematic function is identified, the tabular report and Sankey graphs can be used to further assess the severity and the extent of the change. When a concrete function's performance is deemed suspicious, we can start comparing results with changes in the kernel code or contact Kernel Developers for further consulting.

5. Once the performance problem is confirmed, an engineer can open a Jira ticket to track the issue. Here, the standalone self-contained Perun reports are convenient for effortlessly describing the problem and sharing performance testing results.

### DEPLOYING A RESEARCH TOOL IN PRACTICE

Making a research tool created in academia usable by engineering teams in industry often entails numerous challenges and obstacles.

#### Presentation of results:

Researchers developing an experimental tool typically do not care much about how their tool presents results to users. As (usually) the only users of their own tools, they are used to deciphering slightly cryptic command line output, text logs, or files scattered in different directories. Perun was, in some respects, no different from other research tools. Although Perun supports various visualizations of performance results, there was no concise and self-contained report summarizing the performance of a specific project version, or how it compares to other versions. However, as we quickly learned,



presenting results concisely is one of the most important aspects of a research tool striving to make its way into the real engineering world.

**Scalability:** Most research tools are being developed by a handful of contributors, so there usually are not enough hands to address all the potential performance or scalability issues. Immediately after Perun was integrated into the Red Hat performance analysis toolchain, the Kernel Performance Engineering Team discovered that Perun does not scale well with the amount of performance results they generate each day. Luckily, we managed to quickly pinpoint and hotfix issues stemming from costly, eager imports of third-party libraries by adopting the [SPEC 1 recommendation](#) and making Perun scalable enough to handle the workload. Similar scalability issues often hinder the adoption of many promising research tools in practice.

**Installation and distribution:** As most researchers are aware, installing an academic tool can be a feat in itself. Before being deployed in Red Hat, installation from source was the only way to install Perun. This quickly became a pain point for Red Hat engineers, and the slightly obsolete, incomplete, or, at some points, confusing installation instructions certainly did not help. Since then, Perun has been made available as a package on PyPI, the most popular platform for distributing Python packages. However, the distribution and installation of Perun is still an ongoing challenge, with many more steps to go until Perun can be made

more accessible to wider audiences through a more straightforward installation process, possibly using system packaging managers.

**Applying research in new domains:** In the past, we have introduced and later refined [a new algorithm for diff analyses of performance results](#) between two versions of the same software. Although this algorithm worked well for locating the sources of performance drops in user-space programs (such as [CPython](#)), using a tracing profiler, we found that the algorithm in its current form is not easily applicable for diff analysis of kernel-space performance profiles for multiple reasons: the Kernel Performance Engineering Team collects different performance data, the source code and call graphs are not always available to Perun, and the amount of changed code and functions between compared kernel versions is too large for the results to be useful to the person using the tool.

The general lesson is that applying existing research in new (or even just slightly different) domains is often a struggle for research tools—however, that struggle drives further research and leads to new solutions.

## **FUTURE WORK ON PERUN**

Perun is still being actively developed. We are gradually improving user experience based on feedback both from the Kernel Performance Engineering Team in Red Hat and from other kernel developers. We are also pursuing other new research opportunities focusing on industry

Applying existing  
research in new  
domains ... drives  
further research  
and leads to new  
solutions.

and academic collaboration. We are now working on the following enhancements, among others:


**More advanced and self-contained**

**reports:** We aim to improve the difference reports so that they become more informative, contain more advanced and interactive visualizations, and allow performance engineers to annotate reports with their own findings and insights. We believe that by making reports as self-contained as possible, we will be able to save even more of the time now spent on describing findings in emails or Jira tickets to colleagues.

**New research challenges:** We are also working on several new research challenges that emerged

with the adoption of Perun in Red Hat. One of the main challenges is to analyze the differences between kernel performance profiles with widely dissimilar execution traces or call stacks, yet limited context data (such as detailed control or data flow) to provide accurate hints and suggestions regarding the likely source of performance drops. Moreover, we are interested in the efficient collection of more detailed performance data and metrics in kernel-space with tracing (e.g. with eBPF), which becomes particularly difficult when inlined functions and inlined assemblies are used extensively. Finally, we are considering training and/or leveraging AI models to assist

performance engineers with root-cause analysis of performance bugs.

**User experience:** One of our goals is to make Perun more accessible to wider audiences, and improving Perun's distribution and installation process will undoubtedly help with this task. We aim to minimize the number of core dependencies and provide modular installation for systems with tight dependency constraints. We also plan to package and distribute Perun through other packaging systems such as [Fedora Copr](#), making installation easier. Lastly, we would like to improve the scalability and performance of Perun, making it more accessible for environments with limited time and memory resources. 

**About the Author****Jiří Pavela**

is a PhD student at Brno University of Technology, Faculty of Information Technology, working under the supervision of Prof. Tomáš Vojnar and Assoc. Prof. Adam Rogalewicz as part of the VeriFIT research group. His research, supported by Red Hat, focuses on efficient software profiling, instrumentation and performance testing, as well as worst-case performance analysis of real-time safety-critical or mission-critical software.

**About the Author****Tomáš Fiedor**

is currently a researcher at Oracle Labs, specializing in static and dynamic performance analysis and benchmarking. Previously, he was a member of the VeriFIT group at the Brno University of Technology, Faculty of Information Technology, and is the primary author of the Perun tool.

**About the Author****Jiří Hladký**

leads the Kernel Performance Engineering Group at Red Hat, based in Brno, Czechia. With 15 years of experience in performance engineering, he specializes in benchmarking and optimizing the Linux Scheduler for RHEL. His work focuses on ensuring that kernel performance meets the demands of enterprise workloads.

**About the Author****Tomáš Vojnar**

is the head of the Department of Computer Systems and Communications at the Faculty of Informatics of Masaryk University in Brno, Czechia. Prior to this appointment, he was Vice Dean for Science and Research at the Faculty of Information Technology of Brno University of Technology, where he continues to be involved in research projects.

# Making a research will: the human side of project migration

Have a project moving on to a higher plane?  
Make a plan to prevent getting stuck in limbo.

by Heidi Dempsey

As technology innovators, we get excited about ushering in new ideas and implementing new technologies. Sometimes, however, we might overlook the less flashy labor needed to retire old ideas and replace existing software. Those ideas and software may still be operating perfectly according to their original specifications, but the world has moved on around them, and they need some help to transition their users as well as the software to newer implementations. We also often are tempted to overlook the human side of this work, which means convincing the people who have been happily using this technology to accomplish their own work that there are advantages to be gained by redoing their workflows and projects.

One example of this we've experienced recently in the research group is migrating projects that were implemented years ago with VMs on OpenStack to a new, more flexible container infrastructure with Red Hat OpenShift and [OpenShift Virtualization](#). Infrastructure providers can clearly see the scaling and support advantages to using containerization in large datacenters, but the

advantages may be less clear to an individual project owner. In some ways, the human side of a project migration requires many of the same steps we follow for making a will.

## Choose executors

Determine who you trust to make high-level changes to your project in order to be able to support it in the new infrastructure. This may also mean retiring some parts of your workflow that are no longer needed (great!) or reimplementing parts that require some change (oh no!) to still accomplish your project goals. The people you trust with this duty will be your executors. You cannot lead your project and also be your own executor, because you probably don't have the objectivity or the detailed knowledge to watch over all the different aspects of the transition.

For example, a small database that was OK to run on a VM for your project may need to move to a different, more scalable database. Although this will require a migration effort, in the end your project data can grow faster, making it easier for you to collaborate with more researchers. At Red Hat Research, we



About the Author  
**Heidi Picher  
Dempsey**

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



often work directly with project leaders on tasks like this to help speed a transition, but sometimes the right person is an internal university programmer or an open source contributor who worked on the earlier versions of the project. The important thing is to ask the right people to evaluate where changes are needed, and let the rest of the team involved in the transition know who the executors are before the transition starts.

### **Choose inheritors**

Determine what you want to keep in the project and what you want to give away. Projects that run for years can collect features like barnacles. Every once in a while, a thorough scraping is needed to keep the project team moving efficiently. That one feature that was added a few years ago for a single collaborator who has since retired may not be necessary anymore. Look over your project's functions and workflows and decide what is still productive and essential. Name a person to inherit responsibility to watch over the transition or retirement of each function and workflow. These are the people who will ensure that the retained functions continue to operate as needed when the project migrates.

Sometimes the function has nothing to do with the actual software changes, for example the function of communicating to the rest of the scientific community what you are doing with your project. It is important that the people you choose to inherit your important functions are people familiar with the project who also appreciate the value of the function they are caring for

through the migration. Sometimes these people are representing an entire community for large projects, and sometimes they are individual developers who developed their modules on the original project with you. In either case, it is important that they are named specifically. If the migration is done right, they will become beneficiaries of the process.

### **Have witnesses**

Sign your project transition plan in front of witnesses. Okay, you don't need a notary public for this, but everyone involved in the project—the people on the infrastructure team and other teams supporting any transition—must be able to agree on a written description of how the project is changing and who is responsible for verifying that the project is working successfully and as expected after the transition. At Red Hat Research, we find that developers, SQA, and User Experience engineers are often extremely good at specifying what "successful" means for parts of a transition plan, but remember that other specialties can be needed here. For example, your project may have privacy or compliance requirements that require legal review for a migration.

### **Inform all beneficiaries**

Make sure you notify anyone who is affected by the project transition before it starts—in other words, before the reading of your project will. This includes giving your users adequate time to prepare for a transition, of course, but may also include less obvious things like your Identity Provider or the IT staff who bill for your services. When in doubt, overcommunicate, because overlooking


an impact of your project transition, even if it is not part of the core function, can delay and complicate it. That uncle who didn't know he was inheriting your stuffed weasel collection may not be pleased, just like that person who was going to run a workshop with your project on the day it is down for transition maintenance.

### **Make it findable**

Keep your transition plan in a place that the entire team can easily access, and make sure to notify everyone of the location periodically. You know best whether this is a Google Doc, a repository, or a shared folder on your department server, depending on your team. Make sure that you don't send out just one email when the first draft is finished and then stuff the plan in a drawer in your grandfather's roll-up desk.

### **Update as needed**

If that uncle who was getting the weasel taxidermy passes away, or your database developer decides to take a new job, you need to promptly update the plan to show the changes succinctly. Then notify everyone involved that the project plan has been updated (again).

Following these suggestions can help you prepare your project for a successful transition from the earthly plane to a new life in containerized nirvana. More importantly, it can help the people who work with you and those who depend on your project for research, education, and development enjoy moving their workloads to a faster, more flexible environment—one that they could find heavenly as well. 



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.





# 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