

RH RQ

Bringing great research ideas
into open source communities

Paul Cormier and Orran Krieger

*"That's what open source is all about":
a short history of collaboration,
innovation, and education in research*



Meet CCO: a scalable multicloud cost
optimizer for complex workloads

Tuning Linux kernel policies for energy
efficiency with machine learning

Open source education:
from philosophy to reality



Red Hat
Research Quarterly

Volume 5:1 | May 2023 | ISSN 2691-5251

5 years of research:
special section
pages
28-45

Years to build the team.
Months to build the app.
One moment to see them launch.

This is what connecting your clouds feels like.

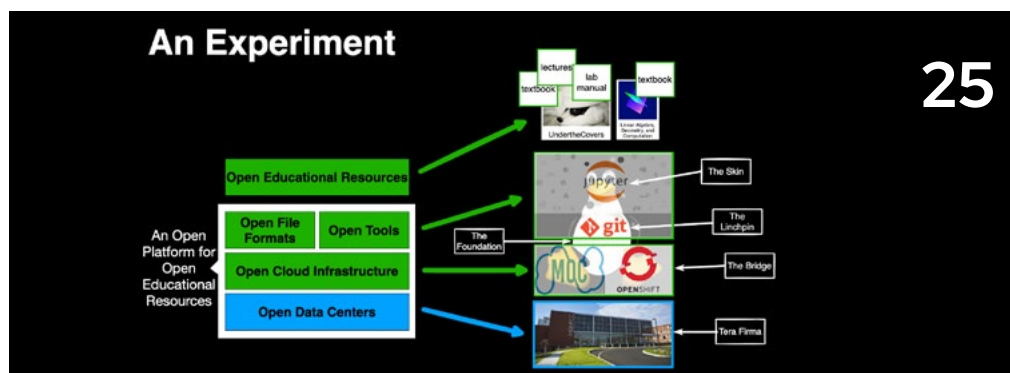
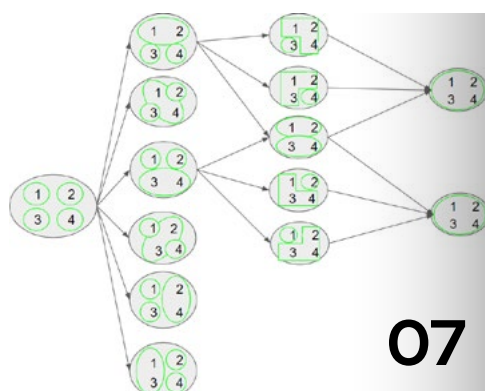


Red Hat

redhat.com/ourcode

Copyright © 2022 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.

Table of Contents



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

NORTH AMERICA
1 888 REDHAT1

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

ASIA PACIFIC
+65 6490 4200
apac@redhat.com

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



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

Departments

- 04** From the director
- 53** Research project updates

Features

- 07** Meet CCO: a scalable cloud cost optimizer
- 12** An interview with Red Hat Chairman Paul Cormier and Professor Orran Krieger
- 20** Tuning Linux kernel policies with machine learning
- 25** Jupyter books enable open source education platform
- 47** Using data to untangle public security certificates

Perspectives

- 28** RHRQ celebrates its fifth year with engineering leaders' thoughts on the past, present, and future of open hardware, research IT and open cloud, security, machine learning, and more.

From the director

**About the Author****Hugh Brock**

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

What have we done? Looking back at 5 years of forging ahead

by Hugh Brock

Things at Red Hat Research have been racing forward lately, as we begin seeing the results of work we started years ago. Already in 2023, we have successfully completed the OpenShift deployment at the New England Research Cloud, the Unikernel Linux paper was accepted at EuroSys 2023 (the premier conference on system software research), and many other exciting research results are approaching publication. So why pause now to reflect? Leaving aside the human temptation to group things in fives, RHRQ editor Shaun Strohmmer and I felt it was important not to let much more time go by without reviewing the accumulated work of the last five years to see what we could learn from our successes, and even our failures. We invite you, our readers, to join us for that review.

Before I get into the retrospective, however, let me review our regular technical section. In this issue, we feature three interesting research results and a piece on a new way of composing instructional material for the college classroom. This education project, called the Open Education Platform, has the potential to change the way we teach at the university level at least as much as the move from chalk to electronic slides—and hopefully with better effect. Developed by Boston University

professor Jonathan Appavoo and improved on by Red Hat Research programmers, this new method uses a set of extensions to an AI process definition tool called Jupyter notebooks to create interactive “slides” for teaching.

Much more than a simple slide deck, these presentations have text, graphics, and live terminal windows and editors to allow students to do their work right in the context of the lecture a professor is giving. They can also save that work directly to a git repository so that it can be retrieved and expanded on later. The entire set of course materials resides in a git repository as well, so it can be open sourced and added to or updated following standard open source practices. The potential for real collaboration around course materials is finally here with this work, and I strongly encourage you to check out both the article and the tools themselves.


Getting back to those research results though: we seem to be focusing on optimization in this issue, and we have two exciting developments to report. The first is about using machine learning models for performance tuning at the Linux kernel and device driver level. I’ve been told that a typical modern network interface card (NIC) has something like 5,000 tuneable “knobs,” of

which only a few hundred are even exposed for tuning in the Linux kernel—and those are mostly set by general-case heuristics that may be wildly inappropriate for the workload being run. In Boston University PhD Han Dong's thesis work on his new project BayOp, he attempts to apply Bayesian optimization via machine learning to the tuning of just two of those parameters in place of the heuristics that were there before. He found that dynamic optimized tuning can dramatically reduce a NIC's power usage without affecting its throughput, simply by smoothing out the incoming packet flow. While this is impressive in itself, the point is not simply that we can make NICs more efficient with this specific tool; rather, the work shows us that significant gains are available across the set of tuneable parameters in an OS when we bring machine-learning-based optimization to bear.

Continuing down the optimization path is a piece on a much higher level of abstraction: a model and a set of tools for optimizing an entire application's use of cloud VMs to provide the required performance for the lowest cost possible. The Cloud Cost Optimizer, detailed in Red Hat Research associate Ilya Kolchinsky's article of the same name, gathers updates on cloud pricing, including spot prices, and produces recommended deployment configurations for a given cloud app. For researchers, students, and even commercial users looking to keep their cloud spending under control and get the best possible performance out of their cloud application, the Cloud Cost Optimizer is quite useful.

It is also yet another interesting use of machine learning to solve an otherwise difficult systems problem.

I promised a second section in this special magazine issue, and I don't think you will be disappointed. Our Perspectives section, special for this anniversary issue, is a detailed look back at what we've done at Red Hat Research since we launched it in 2018. From security to reconfigurable hardware to our own production cloud, we've reviewed all the high points of the last five years and looked toward promising future developments. To be honest, I was astonished when I read through all the surveys and remembered everything we've done. It has been a very busy time filled with a lot of really interesting work, and I'm grateful to have been part of it.

On that note, it seems like a good idea to thank the people who got us to this point. They include the whole Red Hat Research team, who spend their time not only working on code and submitting it upstream, but also writing articles and papers so that people know about it; our editor Shaun Strohmer, who keeps this magazine on track at a very high standard; our research partners, in particular Dr. Orran Krieger of Boston University, who keeps us all inspired; and most importantly Paul Cormier, the Red Hat president, CEO, and now Chairman, who saw an opportunity to bring research ideas into open source and gave me the mandate and the funding to pursue it. Paul and Orran grace the cover of this issue, and deservedly so because without them, none of this would have happened. The whole RHR team and I will remain in their debt. 

Paul and Orran grace
the cover of this
issue, and deservedly
so because without
them, none of this
would have happened.



THERE ARE MANY UNIVERSITIES IN MASSACHUSETTS, BUT ONLY ONE FLAGSHIP FOR MASSACHUSETTS.

As the Commonwealth's flagship public research university, UMass Amherst is committed to pursuing progress for our great state in computer science, technology, engineering, and more. Learn why we've soared to #26 in *U.S. News & World Report* rankings of top-tier public universities and find your degree.

Learn more at umass.edu

University of
Massachusetts
Amherst

Meet CCO: a scalable multicloud cost optimizer for complex workloads

Cost optimization is a core challenge for users of cloud computing platforms. An open source tool is now available to solve it.

by Ilya Kolchinsky

The era of cloud computing has introduced endless possibilities through access to vast amounts of computing power, storage, and software over the internet. This growth has led to a shift towards remote work, collaboration, and the ability for small businesses to compete with larger ones. Cloud computing introduced greater scalability, flexibility, and cost-effectiveness in IT operations, facilitating innovation in data analysis, artificial intelligence, and internet-of-things applications.

However, actually migrating a business to the public cloud and ensuring that cloud resources are utilized in the cheapest and most efficient way has proven to be a highly challenging task. Enterprises must consider many technical, financial, and organizational factors to deploy a complex workload in the cloud, including technical complexity, security and compliance, business continuity, and availability of the necessary skills and expertise. In this article, I will demonstrate a solution for one particularly acute aspect of the above process: cost and resource optimization.

Optimizing the cost of running workloads on a public cloud involves many challenges. One

of the main difficulties is understanding cloud providers' various pricing models, as they vary significantly in their services and associated costs. In addition, workloads that fluctuate and change over time make it challenging to predict usage patterns and optimize resource allocation. Cost optimization requires continuous monitoring and analysis of usage data to identify and eliminate waste, which is often a time-consuming task. Managing contracts and negotiations with cloud providers can also contribute significantly to the overall cost of running workloads. All of these factors make cost optimization a complex and ongoing task requiring dedicated effort and expertise.

AN EXPLOSION OF OPTIONS

Let's start with the most fundamental problem. Suppose you have an application ready for deployment to the cloud. Such an application could be arbitrarily complex, contain multiple components with nontrivial dependencies, and involve diverse resource requirements. Assume for this example that resource consumption requirements (such as CPU and memory) are known in advance for all application components. (I will touch on resource requirement estimation later in this article.)



About the Author

Ilya Kolchinsky

is a research scientist with Red Hat Research and Technion, Israel Institute of Technology. He has a PhD and BSc in Computer Science from the Technion. Ilya's research interests span a wide range of topics in big data processing, such as distributed event-based systems, data stream mining, and AI and machine learning applications in stream processing engines.

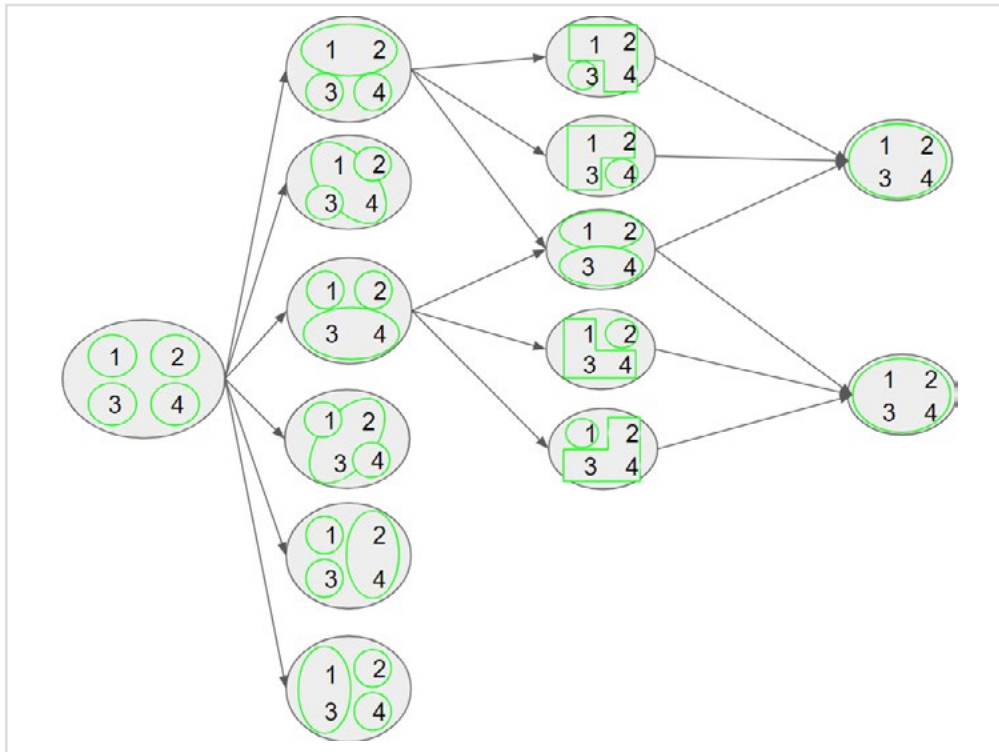


Figure 1. The combinatorial explosion of deploying a workload of four components. Only a part of the solution is displayed.

With all these assumptions in mind, how do you select the most fitting cloud provider for this application at the best price? How do you choose the number of virtual machines to acquire, their instance types, regions, and other crucial parameters? How do you compare the various offers from different providers and decide on the most economically reasonable one?

There is no simple answer to the above questions, for a number of reasons. The leading cloud providers offer a variety of instance types and configurations, each with distinct specifications. According to our estimates, for AWS alone there are over 9,000 different virtual machines available for purchase when

you take into account all combinations of instance type, region, and operating system. For real-life applications requiring a large number of VMs, the sheer number of possible combinations for deploying specific instances and colocating workload components quickly grows exponentially. It is highly impractical, and often outright impossible, to examine all possible alternatives with a brute force approach and simply pick the cheapest one.

Figure 1 illustrates the problem of estimating the cheapest deployment configuration on a toy example of an application with four components. Here, the numbers denote the components, the green circles represent the VMs,

and the black circles represent the different possibilities of allocating components to VMs. The range of allocation possibilities, from using a single VM for the entire application to putting each component on a dedicated instance, is exponential in the number of components. In other words, even for a moderate number of components, the time required to consider all possibilities grows very fast.

In many cases, decision makers simply stick to the same instance types and regions over and over again for all workloads. This strategy could lead to highly inefficient use of resources and significant financial losses, especially given the increasingly dynamic nature of the cloud market. The prices for specific instance types and regions could fluctuate rapidly and unexpectedly—or new, cheaper instance types could be introduced—and opportunities to save costs would be missed.

EFFICIENT SELECTION

This article proposes a different approach. Instead of either being limited to a single configuration or traversing the entire enormous space of possibilities, we can identify and extract a small set containing the most promising candidates. We do this using the advanced methods for combinatorial optimization developed in academia over the past decades. While this method does not guarantee the set will include the absolute cheapest solution satisfying the needs of an application, in the majority of cases it will be sufficiently close.

[Cloud Cost Optimizer \(CCO\)](#) is a project and an open source tool implementing the above paradigm for

Instance Parameters

Required parameters

name *	vCPUs *	Memory *	Pricing Option *	Size (GB)
Component1	8	16	Spot	
name of component	min number of vCPUs for the instance, i.e. 4,8,16 ...	min memory for instance (GB)		min storage size

Optional Instance Parameters

Interruption behavior	Interruption frequency	Network Performance	Affinity Component2, Component4	Anti-Affinity	<input type="checkbox"/> Burstable
		required network performance (Gbs)	example: 1.2	example: 1.2	

Optional Storage Parameters

☒ Choose by specifications

☐ Manual Storage selection

IOPS MIB I/O	Throughput	Storage Type
max iops for instance (GD)	max throughput for instance (MiB/s)	

Figure 2. A partial view of the input parameters accepted by the CCO

optimizing cloud deployment costs of arbitrarily complex workloads. The result of a long-term collaboration on cloud computing between Red Hat Research and the Technion, Israel Institute of Technology, CCO brings academic knowledge together with Red Hat expertise to provide a unique solution suitable for all kinds of clients and applications. CCO makes it possible to quickly and efficiently calculate the best deployment scheme for your application and compare the offerings of cloud providers or even the option of splitting your workload between multiple platforms.

Figure 2 illustrates the input received by the CCO for each component. First, the user provides the resource requirements of the workload in terms of CPU and memory. Other metrics, such as storage and network capacity, could be introduced in a future version. Additional, largely optional input parameters include the relations between the application components (for example, affinity and anti-affinity),

the maximum tolerated interruption frequency, client-specific pricing deals for varied cloud providers, and many more. In particular, CCO can be instructed to consider spot instances, allowing customers to save up to 90% of the instance cost while giving up only a small degree of stability and reliability.

After these details are specified, the optimizer analyzes the provided data and calculates the mapping of workload components to VM instances that minimize the expected monetary cost of deploying the application. The user can limit the search to a single cloud provider or choose a hybrid option that considers solutions deploying the workload on multiple providers for a better price. (As of May 2023, CCO supports AWS and Azure. An intuitive and well-documented plugin interface makes it possible to easily introduce support for additional public clouds.)

Figure 3 (on page 10) shows a sample result of running CCO on a simple application. Given a user query,

CCO produces a list of deployment configurations sorted in order of ascending cost. Each configuration contains a list of instances to be used with a full component-to-instance allocation map.

In addition to the graphic user interface, CCO exports an API and can be executed as a background task or incorporated into a CI/CD pipeline. This is especially useful for incremental deployment recalculations. As discussed above, pricing and availability of instance types are subject to change over time. The only way to ensure maximum cost savings is by periodically executing the cost optimization routine on the fly and making adjustments as needed.

The goal of the CCO project was merely to create a prototype of an innovative cloud cost optimization solution. However, even this prototype can help individual users and enterprises save money in several ways. By calculating and returning the cheapest combination of instances satisfying the

Results:

Price	Region
> 0.2224	us-east-1
▼ 0.2236	us-east-1

Type Name	Total_price	Cpu	Memory	Network	On Demand Price	Discount
> t3a.medium	0.035	2	4	Up to 5 Gigabit	0.0376	70
> t4g.xlarge	0.0699	4	16	Up to 5 Gigabit	0.1344	70
> a1.xlarge	0.0416	4	8	Up to 10 Gigabit	0.102	67
> a1.2xlarge	0.0771	8	16	Up to 10 Gigabit	0.204	67

> 0.2248	us-east-1
> 0.2272	us-east-1
> 0.2284	us-east-1

Figure 3. A sample output of the CCO on a simple application shows a list of deployment configurations in order of cost.

client’s specifications, the CCO allows users to minimize the unnecessary costs resulting from selecting a wrong instance type or unintentional overprovisioning. Further, by comparing the deployment options across cloud providers, the tool helps enterprises choose the provider that best fits their workload needs and budget. This could lead to significant cost savings, especially for organizations with many workloads running on multiple cloud providers. Finally, automating selection of the best instances and regions for a given workload reduces the need for manual monitoring and management.

FUTURE ENHANCEMENTS

While a [fully functional version of CCO](https://bit.ly/RHRQCCO) is available for use (bit.ly/RHRQCCO), there is no shortage of possible extensions and further improvements. Future versions of the cloud cost optimizer could take into account additional considerations such as availability, reliability, compliance, and regulations. In addition, the metaheuristic-based optimization

algorithm employed by the current version could be augmented with a machine learning approach such as deep reinforcement learning. State-of-the-art AI/ML tools have the potential to learn from previous usage patterns and make recommendations for future resource allocation, predict future prices of instances based on the market situation, estimate future interruption rates of spot instances, and so on. Incorporating these capabilities into CCO is an exciting and promising avenue for our future work.

One particularly interesting and relevant problem in this context is accurately estimating the resource requirements of cloud workloads. As mentioned above, the CCO requires per-component CPU and memory requirements as the input for its optimization algorithm. However, manually estimating resource consumption patterns is notoriously difficult for most real-life applications. To address this shortcoming, we are working on another tool, codenamed

AppLearner. AppLearner utilizes advanced ML techniques to learn the application behavior from past runs and predict future resource consumption, in terms of CPU and memory, over time. The forecasting horizon could lie between mere hours and multiple months, depending on data availability and the target application’s complexity. Ultimately, we intend AppLearner and CCO to work in tandem, with the former’s output serving as the latter’s input. In contrast with CCO, AppLearner is still a work in progress, and we expect the prototype to become available later this year.

Those interested in finding out more about CCO, AppLearner, and the rest of our cloud computing projects, or those looking for collaboration and contribution opportunities, are kindly invited to contact Dr. Ilya Kolchinsky at ikolchin@redhat.com. Details about all Red Hat Research projects can be found in the [Research Directory](#) on the Red Hat Research website, research.redhat.com. 



THE UNIVERSAL AI SYSTEM FOR HIGHER EDUCATION AND RESEARCH

NVIDIA DGX A100

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

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

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

A photograph of two men standing in front of a red brick wall in a modern hallway. The man on the left is wearing a dark green button-down shirt and dark jeans, with his hands in his pockets. The man on the right is wearing a dark blue long-sleeved shirt and blue jeans, also with his hands in his pockets. The hallway has light-colored wooden floors and white doors in the background.

"That's what open source is all about"

A short history of collaboration,
innovation, and education in research

An interview with **Paul Cormier** and **Professor Orran Krieger**
conducted by **Shaun Strohmer**

Interview

In 2017, Red Hat Chairman Paul Cormier and Boston University (BU) professor Orran Krieger helped spearhead a collaborative partnership between the two institutions that would come to include expanding Red Hat's participation in the MOC Alliance, the establishment of the Red Hat Collaboratory at BU for research incubation, and the creation of a Red Hat OpenShift Data Science environment at BU for open source education resources. The Red Hat Research team has been managing Red Hat's relationship with BU since the team's inception in 2018.

To celebrate the fifth year of Red Hat Research and the research quarterly, RHRQ editor Shaun Strohmer sat down with Paul and Orran to look back at how and why this partnership formed and its impact on education and the tech industry. They had plenty to say about changes in computing environments, what computer science education needs today, and why open source is the cornerstone of innovation.

Shaun Strohmer: What motivated each party to create the Red Hat-Boston University partnership? Let's start with Paul. What was the appeal of starting a collaborative relationship with the computer science folks at BU?

Paul Cormier: We first thought it was a way to grow a community around OpenStack through collaboration and connect with people early in their careers. From there, it has significantly expanded. The Mass Open Cloud (MOC) played a big part in it, too. It was very enticing to have a place to run all of our products together as a portfolio in a real-world, large-scale environment so we could understand the system aspects of it more deeply.

One thing we've always tried to do at Red Hat—and I've been here 23 years—is look at our product set as a portfolio. In the technology world today, we don't think enough about how things work together as a system. Joining BU in the MOC was a big opportunity to see how our combined products interact in a systems world.

Shaun Strohmer: From the BU side, Orran, what did you hope a Red Hat partnership could bring to the university or the MOC?

Orran Krieger: There were many reasons why building the cloud for academic users made sense, but I really wanted to base it on open source software so we could see everything going on. Red Hat brought a lot of individual technologies for the on-premises cloud, including Red Hat Enterprise Linux, OpenStack Ceph, and so on.

In the public cloud realm, there's increasingly this oligopoly of three or four players. Everything is closed, so you can't see what's going on. But Red Hat specifically is about openness. Openness eventually wins, and by having Red Hat as the anchor partner, when we bring in other partners who might be more focused on proprietary things, we can still impose openness on everything.

Shaun Strohmer: Did the partnership initially start with Red Hat providing certain resources?

Paul Cormier: I wouldn't put it that way. Resources weren't the first thought in our minds, and I don't think they were in Orran's mind either. For us, the beginning was about getting broader: a broader look, broader participation, a broader take on these technologies we knew were the underpinnings of what was emerging as cloud technology.



About the Author **Shaun Strohmer**

is the Red Hat Research Quarterly editor. 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.

For us, the beginning was about getting broader: a broader look, broader participation, a broader take on these technologies we knew were the underpinnings of what was emerging as cloud technology.

Shaun Strohmer: Then the relationship also broadened to include specific research projects. How did you hope Red Hat's engagement in individual projects would advance them?

Orran Krieger: I've had a foot in the academic research community and industry most of my career. I like doing research involving innovative things that lead to publications. But I also want the things I do to have a real impact, to be integrated into things that make a difference to people and society.

As systems are getting more complicated, no individual faculty member has the expertise on the details of different projects that are out there. You can't build the system from scratch. I also wanted to create a model where engineers engaged with students so the students could have a tangible impact on open source software that could host real users. It's a very innovative thing—I don't think there's anything like this anywhere else.

When I learned how to program, a lot of it was essentially apprenticeship, where people who were better than me were beating me up. I can't do that with a huge number of students, but I can involve engineers in projects and get real criticism and learn modern practices. That engagement has added a degree of expertise and professionalism that's allowed students and projects to have an impact they couldn't have otherwise.

Shaun Strohmer: Paul, what were your initial thoughts on getting Red Hat engineers engaged in university research projects?

Paul Cormier: We didn't want the university doing work to augment our products directly. Instead, we were interested in research that would feed into our products three, four, five years or more down the road.

Open source is such a big part of everything in the software world today. If you look at what Red Hat did to be successful, we started with pure open source, which is really a kind of research. And as Orran said, it's a place to get an apprenticeship. You put your ideas and code out there and get beat up on it. That's what open source is all about. What Red Hat did was take that model and develop the ability to build commercially consumed enterprise software. We had a head start in the research world just because we were based in open source.

Orran Krieger: One of the things we talked about then—and I'm super excited because it's actually happened—is accelerating the impact or the transition to product. So much in the open source community still goes through a release cycle. What we started talking about, even in that first meeting, was whether we could take things from research and expose them at a very early stage, in a CD (continuous delivery) kind of way.

There are three ways of doing radical innovation. There's the research community, which has been doing radical, out-of-the-box thinking for a long time. There's the open source community, which radically changed all the software we're using today. And then there's that cloud world, where they often innovate even before releasing, testing a change at scale and allowing for rapid evolution. The open source community

hasn't had that, and collaboration seemed like a way to get all three things—the cutting-edge thinking of research, the impact of open source, and accelerated innovation—tied together.

Paul Cormier: I agree. The open source community is big, but it's also very individualistic. They don't have that big testbed. But I also remember we talked about the skill sets needed to operate a cloud at scale: we realized early on that it would be a hybrid, multicloud world. One cloud by itself was never going to be sufficient. So what are the skill sets? The only people that knew how to operate clouds at this scale were the Big Three cloud guys.

So we looked at research from that perspective. What skills, tools, and processes do people need to operate clouds at scale? In a hybrid cloud world, that's what many companies have to figure out—and that was a really attractive thing too. That led to building almost a whole new profession of site reliability engineers (SREs). That doesn't sound new today, but it was back then.

It's a very innovative
thing—I don't think
there's anything like this
anywhere else.

Orran Krieger: That part is sometimes forgotten. We have this whole high-performance computing (HPC) community, and we have the Massachusetts Green High-Performance

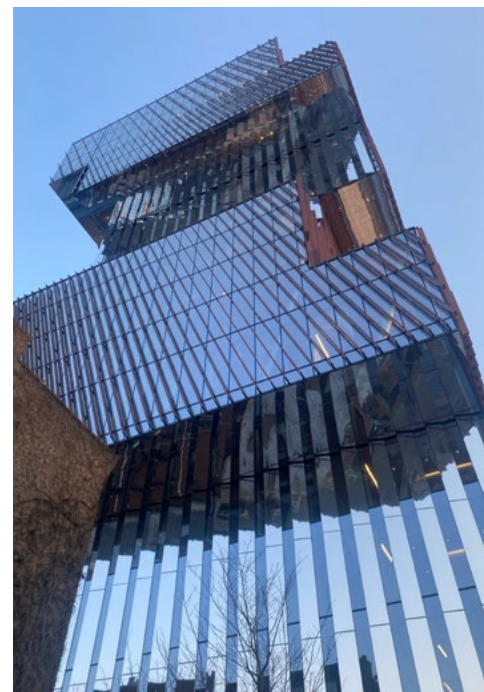
Computing Center (MGHPCC). They're now transitioning to operations on the MOC for much of what they do. We have 500,000 cores of compute, 500 petabytes of storage, and tens of thousands of users just on the HPC clusters, and increasingly that demand is shifting to cloud. We have people here doing operations that are used to scale, so we already have a community of operators that are in a position to say, "This is what you're going to need to support production operations."

Shaun Strohmmer: I want to discuss a factor you mentioned earlier: the ability to bring in other industry partners. What are the benefits of that? And what are the challenges?

Paul Cormier: It is a huge benefit. Sometimes people get confused about what Red Hat is. One thing I always say is that Red Hat is not an open source company: Red Hat is an enterprise software company with an open source development model. It's a different thing, right? We do all of our development in open source because we think that open source is a better way to get to a better outcome.

Some of our commercial competitors look at us as a software company and not necessarily as an open source company. Some may not know what open source means. But working in collaboration with universities has been a powerful demonstration of how we do open source. Now this open source-based technology around the cloud is driving where the whole industry's going.

We're not doing this just to make Red Hat products better, though certainly that is an outcome we'd love



The Red Hat Collaboratory has moved to the brand new BU Center for Computing and Data Sciences.

to see. The first order of business was to solve problems in a completely open world with anyone, especially from the academic community, that wanted to participate. That really helped other commercial software companies and even some of our competitors better understand what Red Hat is all about. Before, they just saw us as a competitor, winning or losing deals with each other. This was a great way for us to illustrate what open collaborative development truly means. When you've got university partners, by their nature, what they do is open, collaborative development.

Orran Krieger: I can't expand on what Paul said except to say that I violently agree. Also, no one company has the power to do everything.

Collaboration seemed like a way to get all three things—the cutting-edge thinking of research, the impact of open source, and accelerated innovation—tied together.



BU PhD Ali Raza presents his thesis work on Unikernel Linux. His related paper on the subject will appear at top systems conference EuroSys 2023 in Italy in July.

As for the challenges, I get nervous about some of the benefits being lost if we aren't mindful of them as we grow. One of our tasks initially was finding a model for aligning the incentives of engineers and researchers, so they want to work together. If people's incentives are purely publications or purely how many lines of code they commit to open source or get into a product, it doesn't work. It's a lot to [Red Hat Research director] Hugh Brock's credit that we've been so successful in bringing them together.

Shaun Strohmer: Paul, how has this partnership benefitted Red Hat over time, whether in terms of talent or innovation or something else?

Paul Cormier: Red Hat is a software vendor by heart—we like to code things. But we realized we would need people with specific skills as we got into the work.

Now we have our own managed services, where we manage our products and our partner's clouds for our customers. It's one of our fastest-growing segments. And it grew from the knowledge and tools we got from working with BU. One of the reasons why managed services are one of our fastest-growing segments is that the industry doesn't have enough talent for customers to do that themselves. This collaboration was the beginning of identifying and meeting that need from both a skills perspective and a tools perspective.

Orran Krieger: That's really exciting. We've experienced the flip side of that. We brought vanloads of students down to MGHPCC when we racked 500 servers that had arrived, which was so unfamiliar to them. The computer is sort of this magic box for a lot of students. This partnership



BU professors Vaisha Kalavri and Jonathan Appavoo present their Red Hat Collaboratory project "Towards high performance and energy efficiency in open source stream processing".

engages students in something real, which they then get to use.

Paul Cormier: I totally agree with you. People who come up through the Linux world often play with this independently. But when we recruit from universities, we see some people, even in computer science, who don't understand how a computer works underneath. It's fantastic that you get that benefit on the student side.

Shaun Strohmer: Let's move from getting students to understand systems to the value of systems research more broadly.

Paul Cormier: One of the questions I often ask is, "What problem are we trying to solve?" It's so valuable to have that big-picture understanding of the problem and have the skills

to drill down into the piece you're trying to solve at the moment.

I see many people who only understand the big picture. And I see another set of people trying to solve their little piece of the problem without understanding where it fits in the bigger problem. This is what systems are: if you can traverse that system up and down, you're a much better problem solver. And systems are just getting bigger every day.

Orran Krieger: The assumptions are changing. Before, Moore's Law essentially said, "Don't worry about adding more stuff, because things are going to get faster anyhow," so we—as an industry—added layers and layers of crud. And the networks have gotten hugely faster, solid-state drives have changed, and persistent RAM accelerators have changed things. As

we see the end of Moore's Law and Dennard scaling, systems are more critical, and they're critical at scale, because problems occur at scale that don't occur when it's not at scale.

Shaun Strohmer: Let's talk about computing environments. We've talked a lot about cloud, but what about on-premises computing? Will that continue to be relevant?

Paul Cormier: It's as relevant as cloud computing. It's a hybrid world. On-premises is still relevant because there's a whole set of applications that, for a hundred reasons, aren't going to move to the public cloud, whether it's security, data, privacy, location, or what have you.

The problem is I don't think much of the IT world understood that until they got into moving applications to the cloud. Big businesses, back when Amazon Web Services (AWS) was just taking off in the mid-2000s, were planning to have 90% of their apps in Amazon within five years. Now we're helping them build private clouds and hybrid environments, and they're only about 25% in the cloud. They're probably going to grow that, but the point is they didn't understand why they had to be a hybrid shop, or what it meant, until they started getting into the work of cataloging their applications and understanding what they do, how they interact with a thousand other applications, the cost of where they're going to run it and where they're not going to run it, and what it was going to take from a software development perspective to get out to the cloud.

Once they understood that, they became hybrid believers. Over the last three or four years, we have

run into that in spades every day. It's a very different world than six or seven years ago when people were thinking, "We're just going to the cloud because it's easy."

Shaun Strohmer: Orran, what about from a university or research perspective?

Orran Krieger: A decade from now, I don't believe this question will even be relevant. The question is, how do we make a system so it's elastic and we can shift computing between them? What we've been doing here at BU is moving stuff between HPC clusters and OpenStack and OpenShift with different compliance regimes. Ultimately, it's about clusters of computers—datacenters—shifting computing around and building services that can span all that.

Paul Cormier: I couldn't agree more. From a system perspective, we just expanded that big picture to include whether it will run in multiple clouds, on-premise, and so on. That's now the new big design point to design down from. Look at microservices. That's about breaking up a monolithic application into a set of services that can potentially run everywhere. That's just good software design.

Orran Krieger: University research is a good example. You have multiple universities, multiple clusters deploying all kinds of things. If you've got data sets for gene sequencing or data sets from NASA, they will be used by thousands of universities that want to access the data set in many different places. We want those places to collaborate in terms of hosting the data efficiently,

not having it redundantly stored across many different things. It's actually systems of systems.

Shaun Strohmer: Before we finish, I'd like to hear from you about the research work you think will be important in the next few years and how the Red Hat-BU partnership has unique opportunities to engage in it.

Paul Cormier: It might not sound new, but the multi- or hybrid-cloud world is still new in every aspect. Ninety-nine percent of the software I see does not run in a hybrid cloud world. We have security issues across that, and data, privacy, and management issues. For a lot of software today, the clouds and the on-premises are just a bunch of islands. We have a ton of work to do before we're truly running that as one system. Before we can expand, for example, to use AI and machine learning and automation to do hybrid better, we need to understand this new world better.

Don't get me wrong: I'm not saying we don't understand hybrid. I'm saying that there is still so much to learn there. We will have to ask exactly the questions we discussed earlier. Where does it fit in the big picture? And what are the things I need to do with the lower picture? How does what I'm doing affect the layer above me or three layers above me?

Orran Krieger: So systems matter—they really matter. On one side, you're taking things to a massive scale; on the other, you're hyper-optimizing things for what they're actually doing. Take Kubernetes. It says, we're going to make the kernel standardized, and it doesn't matter what the application runs on.


We're figuring out how to take systems not just as a pool of things but actually optimize them for what they're doing, all across the stack. There's a very cool project at the Collaboratory now [see BU PhD Han Dong's article, "Tuning Linux kernel policies for energy efficiency with machine learning," in this issue] using AI to analyze and optimize a specific workload, then cut its energy use by one half. I think the role of the operating system and the hardware—especially as we've gotten away from a world where we're getting performance improvements just by the clock cycle doubling every few years—will be increasingly important.

Paul Cormier: That goes back to what we were talking about earlier. We do those optimizations in all layer levels, but now how do they fit back into the big picture? And what are the tools we now need to develop to manage and run it?

Orran Krieger: We've got our work cut out for us—we'll have job security in the system space for a while!

Paul Cormier: It's a great time to be a systems engineer.

Orran Krieger: When I entered the field and built my first system, it was this dismal world where operating systems were proprietary, OS innovation could only happen within companies, and performance didn't matter anymore. And today, open source and systems are at the center.

Shaun Strohmer: That's a perfect summary. Thanks so much to both of you for taking the time to talk with us today and share these insights on where we've been and where we need to go. 



MOC ALLIANCE

MAKING THE CLOUD LESS, WELL, CLOUDY

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

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



[@mass-open-cloud](#)



www.massopen.cloud



contact@massopen.cloud

Housed at:

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

**BOSTON
UNIVERSITY**

Feature

**About the Author****Han Dong**

is a postdoc in the Computer Science department at Boston University. His research interests lie in distributed systems, high-performance computing, and operating systems. He is interested in research addressing the growing energy needs of our modern systems.

Tuning Linux kernel policies for energy efficiency with machine learning

Presenting BayOp, a generic ML-enhanced controller that optimizes network application efficiency by automatically controlling performance and energy trade-offs.

by Han Dong

As global datacenter energy use rises and energy budgets are constrained, it becomes increasingly important for operating systems (OS) to enable higher efficiency and get more work done while consuming less. Concurrently, the environmental footprint of hardware manufacturing is [increasing](#), further underscoring the importance of extracting more value from existing resources. The complexity of modern systems software and hardware and the end of Dennard scaling and Moore's Law only add to the difficulty. Meeting these challenges will require solving the problem of how to specialize different kernel policies and hardware configurations that are often written to support only general use cases.

Moreover, as latency-critical applications, such as key-value stores (i.e., memcached), become ubiquitous across datacenters, cloud service providers more frequently choose to deploy them on dedicated hardware. Motivated by the rise of this software-hardware dedication, this article presents BayOp. This generic controller optimizes the efficiency of network applications by taming and controlling the system's performance

and energy trade-offs automatically. BayOp uses an established machine learning (ML) technique, Bayesian optimization, to exploit two hardware mechanisms that externally control interrupt coalescing and processor energy settings. The key insight behind BayOp's dynamic adaptation is that controlling interrupt coalescing induces batching to stabilize application latency periods, making it easier to control performance-energy trade-offs and magnifying the benefits of batching with processor energy settings.

Our team of Boston University researchers and Red Hat engineers are making the following contributions toward realizing BayOp:

- We conducted a novel performance and energy study of two network applications by sweeping up to 340 static combinations of (1) a network interface controller's (NIC) interrupt delay setting (ITR) to control interrupt coalescing and (2) the processor's Dynamic Voltage Frequency Scaling (DVFS) to control processor energy settings.
- Our study found that performance improvements of over 74% are possible

in Linux for a simple ping-pong network application by using a static ITR. We investigated the performance and energy trade-offs for a memcached server and found that tuning both ITR and DVFS can yield 76% in energy savings.

- Our data also reveal that tuning ITR and DVFS results in stable OS behavior, which implies that this structure can be captured formally. Based on these findings, we developed the BayOp controller, which can dynamically adjust the settings of a memcached server to adapt to changing offered loads and performance and energy goals while meeting different service-level agreement (SLA) objectives.

ITR PERFORMANCE
STUDY IN NETPIPE

A common feature of modern high-speed NICs is the ability to control batching via interrupt coalescing. In this work, we explore this mechanism on an Intel 82599 10GbE by using its ITR register, which is exposed by Linux ethtool. Software typically uses the ITR register to configure a delay in 2-microsecond (μs) increments from 0 to 1024 μs . If the spacing of interrupts is less than the ITR value, the NIC will delay interrupt assertion until the ITR period has been met. Linux’s network device driver typically contains a dynamic policy that seeks to performance-tune the ITR value to better reflect the current workload.

However, our study reveals that Linux’s default dynamic ITR policy can result in performance instabilities even in a simple network ping-pong application such as NetPIPE. **Figure 1**

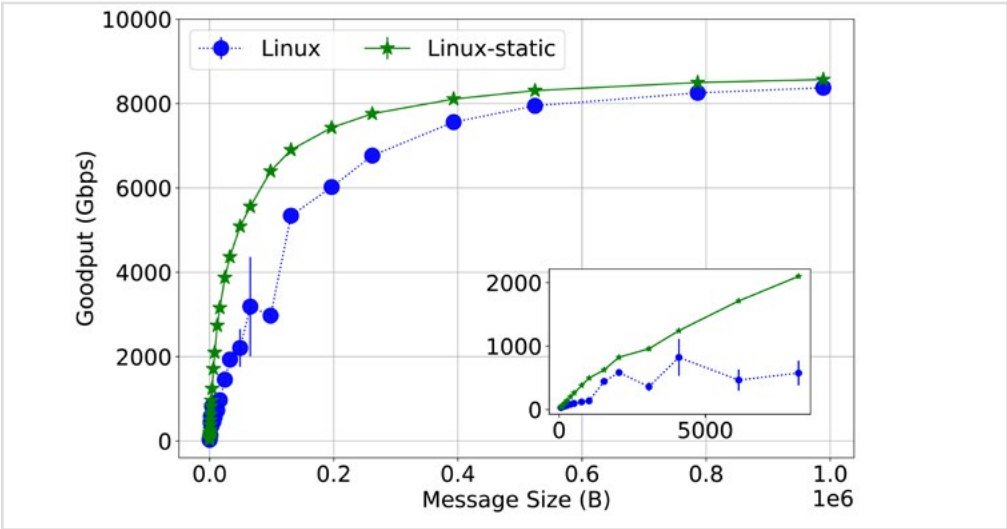


Figure 1. Goodput measurements for NetPIPE across different message sizes from 64 bytes to 1 MB. The inset is zoomed in on message sizes less than 8 KB. The error bars on each point show the standard deviation of measured performance.

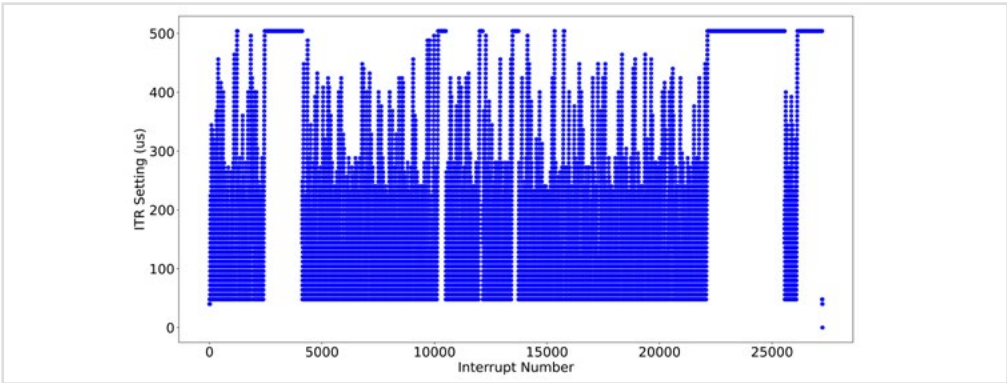


Figure 2. ITR values set by Linux’s dynamic ITR algorithm. This is captured during a live run of NetPIPE at 64 KB message size.

illustrates the measured performance, or Goodput, differences for a range of message sizes between Linux, which uses its dynamic ITR policy, and Linux-static, where we disabled its dynamic policy and selected a single fixed ITR value instead. This figure illustrates that using a static ITR was able to achieve higher Goodput in all message sizes. For example,

at 64 KB messages, Linux-static improved its performance by 74%.

This performance difference can be traced to the behavior of Linux’s dynamic ITR policy. **Figure 2** shows a snapshot of every updated ITR value captured in Linux’s network device driver during a single run of NetPIPE using 64 KB message sizes. This figure

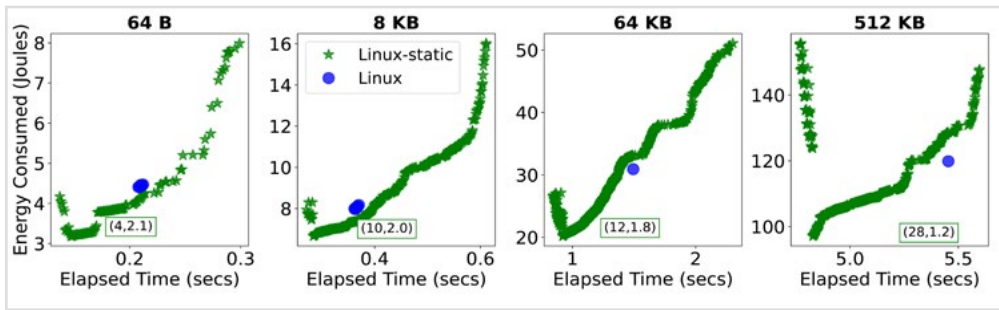


Figure 3. NetPIPE performance and energy results for different message sizes. Every Linux-static datapoint is the result of a single experimental run with a unique (ITR, DVFS) combination. The X-axis is a measure of performance (lower is better), and Y-axis shows total energy consumed. The labeled (ITR, DVFS) pair are experimental Linux-static values that resulted in lowest energy use. The number of round-trips is fixed at 5000 for each message size.

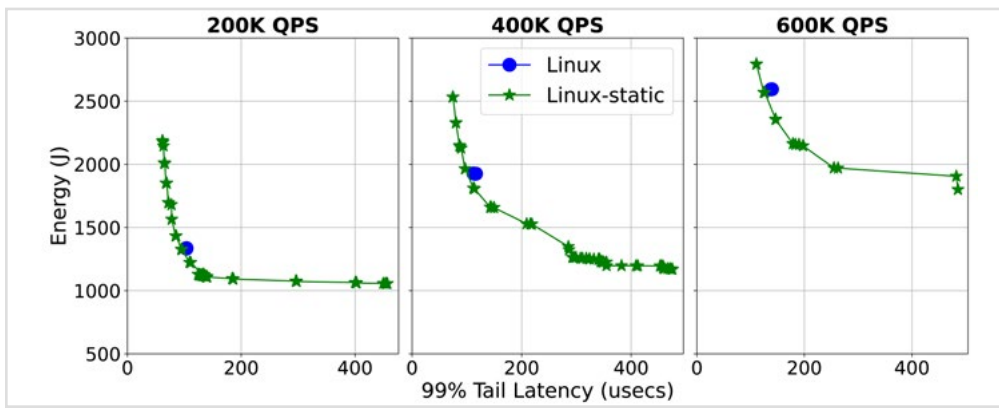


Figure 4. Each point represents a single experimental run of memcached at different QPSes. Each Linux-static data point uses a unique (ITR, DVFS) pair, and we only illustrate data that lie on the Pareto-optimal curve. The X-axis is a measure of performance (lower is better), and Y-axis shows total energy consumed.

illustrates the extreme variability (up to 500 μ s) at which ITR is updated on a per-interrupt basis. This variability suggests that the current dynamic policy, designed to support general use cases, is operating at the wrong timescale for an application such as NetPIPE and that further specialization can yield significant advantages.

PERFORMANCE AND ENERGY

We then expanded our study to uncover the performance and energy

trade-offs within this space. Toward this end, we explored the novel use of ITR and DVFS by statically configuring up to 340 unique combinations of each across both NetPIPE and a memcached server. In **Figure 3** and **Figure 4**, we compare Linux-static, which statically sets both, against Linux, which uses the dynamic ITR policy and dynamic DVFS powersave governor. Memcached is an example of an application with an external request rate that can largely be considered independent from the

time required to service a request. Service providers often set an SLA target for these types of applications, such as some percentage of requests to be completed under a stringent time budget (99% tail latency < 500 μ s for Figure 4). At the same time, there is a constant stream of requests-per-second (QPS) arriving at the server.

Figures 3 and 4 illustrate a rich performance and trade-off space between Linux and Linux-static, whereby tuning both ITR and DVFS can result in dramatic energy savings of over 50% and even improve performance in the case of NetPIPE. This study also reveals characteristic shapes of Linux that differ depending on the application. For NetPIPE, there is the V shape. The lowest point in this V shape represents a setting that uses the lowest energy while being competitive in performance; the vertical points above represent other configurations that sacrifice energy for better performance. Note that Linux always lies to the right of the V curve, indicating the value of doing such a static search. In contrast, memcached reveals an L shape. While this L shape differs in absolute performance and energy, the underlying Linux response to changes in (ITR, DVFS) combinations remains stable across the offered loads, which suggests one can capture these behaviors formally.

BAYOP DESIGN AND RESULTS

To operationalize these energy savings and stabilize OS response behaviors, we built BayOp, an application- and OS-agnostic controller using a sample-efficient ML technique—Bayesian optimization—to automatically probe for efficient (ITR, DVFS) settings within a running system while under changing

offered loads. In particular, we targeted applications such as memcached, where the SLA space enables a rich set of performance and energy trade-offs.

We used BayOp to automatically tune a memcached server while servicing a publicly available [memcached trace from Twitter](#). Twitter's trace reveals that these services often maintain a mean demand curve that changes slowly over periods of 24 hours or more. These can be attributed to either diurnal access patterns or can be induced through service admission control layers such as load balancers. These curves suggest that specialization of a single application at a specific offered load can be a realistic form of optimization to exploit the stable regions of these demand curves.

Figure 5 illustrates the BayOp controller design. In phase 1, a live system running memcached is currently servicing various QPSes from an external source. BayOp will then periodically trigger a set of performance and energy measurements of the live system in phase 3. For each measurement in phase 3, the Bayesian optimization process is used to compute a reward penalty of its current configuration and then, in phase 4, recommend and update a new (ITR, DVFS) pair on the live system such that it minimizes the reward penalty. Once this process is finished, the memcached server is set with a static (ITR, DVFS) setting until the next set of measurements is triggered.

Figure 6 illustrates results from two different SLA objectives: 99% latency < 500 μ s, and a stringent 99% latency < 200 μ s. The figures to the left illustrate the system's energy

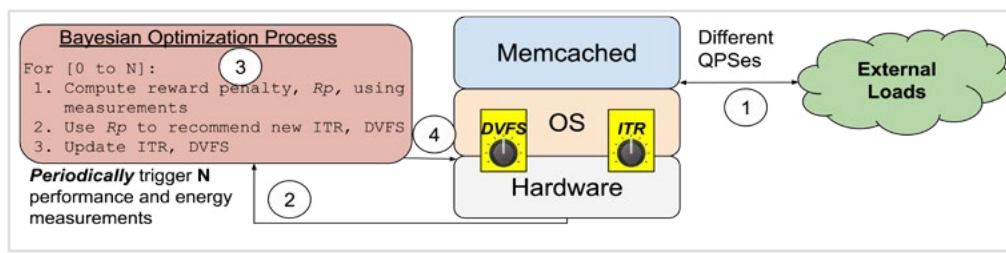


Figure 5. BayOp controller for a memcached server

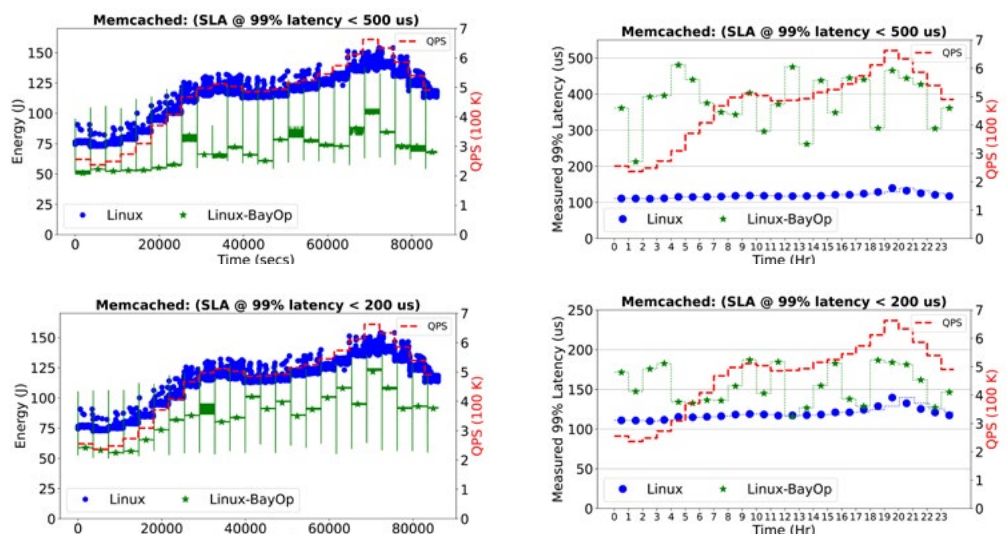


Figure 6. BayOp applied to cache-trace QPS rates over a 24-hour period for memcached.

Each row represents a different SLA objective and shows the measured energy-per-second as QPS changes across the five system configurations studied.

usage on a per-second basis over a 24-hour period, and the change in hourly QPS rate is also indicated. At the beginning of each hourly QPS rate, there are spikes in energy usage from Linux-BayOp, the result of the Bayesian optimization process as it dynamically searches through (ITR, DVFS) settings on the memcached server to meet its corresponding SLA objective. After this initial energy spike, the system settles to a steady energy consumption state until the next hourly trigger. The figures to the right illustrate the latency trade-

offs BayOp makes to maintain SLA objectives while saving energy.

In the case of an SLA objective of 99% latency < 500 μ s, we find that Linux-BayOp can result in energy savings of up to 50% over Linux. Even at a more stringent SLA of 99% latency < 200 μ s, Figure 6 shows that Bayesian optimization can adapt to this new requirement while saving up to 30% energy. These results demonstrate the generality of the BayOp controller. They also reveal the limitations of Linux's existing dynamic policies, as one cannot

DEVCONF.cz

open source community conference

June 16-18, 2023

 Brno, Czech Republic

REGISTER NOW




www.devconf.cz

use its current ITR and DVFS algorithms to express these rich performance and energy SLA trade-offs.

FUTURE DIRECTIONS FOR BAYOP

BayOp's design as an external controller creates the potential to integrate with load balancers to exploit and optimize a dynamic fleet of servers that direct incoming offered loads to one or more servers configured with specific ITR and DVFS. Additionally, if servers are added or removed from the fleet, BayOp re-optimization can then be coordinated with re-balancing.

In our use of BayOp to optimize memcached servers, we make certain simplifying assumptions, such as the hourly trigger to run the Bayesian optimization process. While we have demonstrated that these simple assumptions can result in significant advantages, there is considerable room for improvement. BayOp's architecture enables the integration of more advanced policies for deciding when to trigger the Bayesian process, such as in response to dramatic changes in QPS rates or exploiting historical patterns in service loads. The Bayesian optimization package can also be improved to reduce the cost of sampling.

Our work is currently being extended through the Red Hat Collaboratory at Boston University to improve performance and energy efficiency for open sourced stream processing applications as well. Reach out to handong@bu.edu with questions, see the data collection infrastructure at github.com/sesa/intlog, or visit the project page on the [Red Hat Research website](https://www.redhat.com/research). 

Open source education: from philosophy to reality

Researchers, interns, and industry engineers have joined forces to create an open education platform using Red Hat OpenShift Data Science.

by Danni Shi

Open source technology has transformed many industries, and education is now poised to be the next frontier. Open Education (OPE), an innovative project initiated by Boston University professor Jonathan Appavoo, is revolutionizing how education is delivered and consumed. OPE aims to put education on a path to open source, empowering educators in any discipline to create, publish, and collaboratively develop high-quality educational materials that students can access with just a web browser.

In collaboration with Red Hat Research, OPE leverages modern open source technologies to create an open environment and platform for education. Education has traditionally been a closed system, with students required to pay hefty fees for access to materials and classes. This has created significant educational barriers. OPE is helping to break down these barriers by making learning resources freely available to all, enabled by changes in cloud computing, open source technology, and education trends:

- The widespread adoption of cloud computing platforms provides scalable

and cost-effective infrastructure for running data science workloads.

- The increased availability of open source technologies like Jupyter Lab, GitHub, and machine learning tools has facilitated the development of educational content and resources that can be created and shared in a cost-effective way.
- The demand for remote online learning resources is increasing, motivated by the COVID-19 pandemic and the drive for educational equity.

In addition to increasing access, OPE facilitates a more collaborative and efficient approach to learning. By sharing resources, teachers and students can learn from each other and build on each other's ideas, leading to a better educational experience. The platform also prioritizes ease of use and simplicity in its interface and functionality, so people with varying levels of technical expertise can use it.

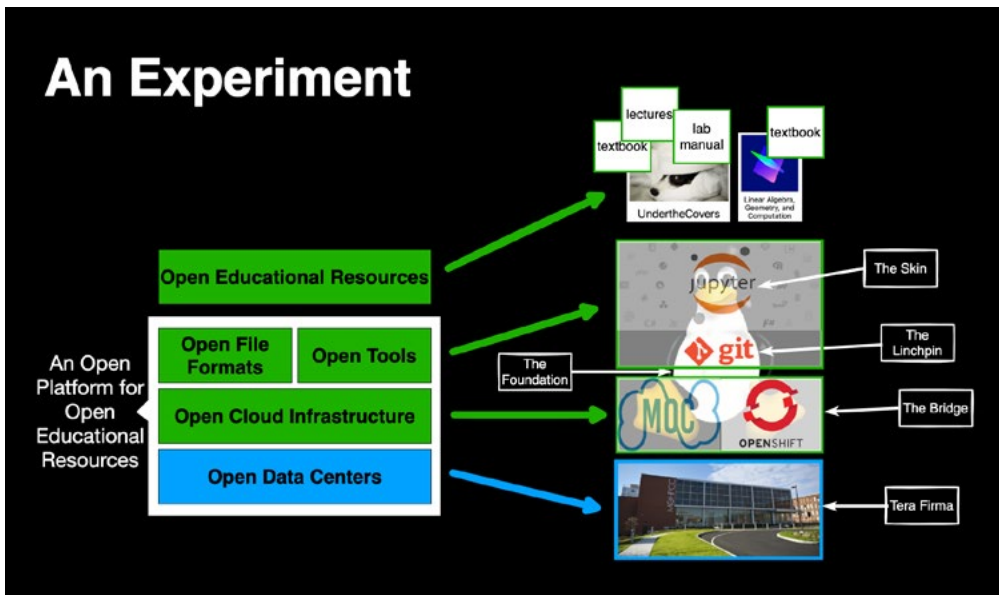
The open ownership model of OPE starts with high-performance, open datacenters that provide the necessary hardware resources. The project exploits Linux to enable the use



About the Author

Danni Shi

is a senior software engineer at Red Hat, leading the development effort for the OPE project. She is dedicated to advancing open source education and improving the accessibility of technology for all.



This diagram shows how the open ownership model utilizes open datacenters and Linux to create a rich environment of tools and services that support a new approach to educational material.

of cloud platforms, serving as the foundation for a rich environment of tools and services that support a novel approach to educational material. The current framework uses container infrastructure, Git, and JupyterLab, and is hosted on a Red Hat OpenShift Data Science (RHODS) platform. Using RHODS allows OPE to leverage the platform's scalability and security to provide a reliable, consistent environment. This allows OPE to focus on creating educational content while RHODS handles the underlying infrastructure and platform management.

The OPE Template consists of several independent core branches:

- A base container branch that provides an interface for specializations
- A tooling branch that offers support for building course content,

publishing content, customizing containers, and publishing containers to a repository

- A test book branch that tests various components to ensure the preservation of compatibility across updates

THREE TRACKS TO SUCCESS

The OPE project team, consisting of BU professors, teaching assistants, and Red Hat engineers and interns, has been working on several simultaneous tracks to make OPE a success:

The first track investigates generating engaging educational experiences using technologies produced by open source communities centered on JupyterLab. Arlo Albelli, a BU graduate student, led the summer 2022 OPE effort. He is the author and editor of the [Under the Covers](#)

open source textbook and a course instructor. Arlo is directly engaging with BU Computer Systems 210 course content and identifying prioritized tasks and features based on class feedback.

As part of the OPE project, the team also integrated RISE (Reveal.js Jupyter/IPython slideshow extension) in the JupyterLab environment and developed a slide layout extension. RISE is a Jupyter extension that transforms Jupyter notebooks into interactive presentations with live code, visualizations, and narrative text ([see a demo and documentation](#)). The slide layout extension allows users to create custom slide layouts, enabling more complex or customized presentations. Yiqin Zhang, an software engineer intern from BU, led the development of these extensions.

The second track is building custom Jupyter container images to provide more infrastructure options. Abirami Dhayalan, an intern from the University of Texas–Dallas, focused on the development of base OPE container images with the necessary software and packages installed to support the Jupyter environment and authoring tools. She standardized the container-source-to-image (S2I) building process and developed Fedora and Ubuntu-based multistage image builds.

The third track focuses on developing an automated test framework for OPE content that could potentially become part of the supported Red Hat OpenShift Data Science platform (RHODS). Xinyi Wu, a co-op

master's student from Northeastern University, worked on the test book, designing it to be a sampler book for the current and future courses. She also incorporated a RHODS testing framework with OPE test cases. Her work has enabled us to automate the testing process, making it more efficient and improving our development cycle. In the spring of 2023, intern Griffin Heyrich is improving test coverage and adding new test cases to OPE. Our goal is to make the OPE test framework usable and ensure that the builds and run processes are reliable.

Danni Shi is the Red Hat Research regular staff responsible for mentoring and communication work. She provides guidance and support to interns on each of the three tracks, helping them build the skills and knowledge they need to succeed in their tasks. This includes providing feedback on their work, offering suggestions and recommendations, and helping troubleshoot any issues.

OPE GOALS FOR 2023

With new interns joining the team, OPE has set new goals for 2023 to develop its tooling and systems further. These enhancements aim to improve the user experience and make it easier for individuals and organizations to use and extend the open education project and test framework. One of the primary tasks for OPE in 2023 is to make container image builds more reliable and faster. The OPE team is focused on reducing the container image size by using lightweight base images and multistage builds. Intern Isaiah Stapleton is leading the effort to

minimize container image size, which will help speed up the image-building process and ensure that builds are completed quickly and efficiently.

OPE is helping to break
down barriers by making
learning resources freely
available to all, enabled by
changes in cloud computing,
open source technology, and
education trends.


Another important goal for OPE in 2023 is to concentrate on book content and interactive elements, ensuring that books have integrated assignments and that the content is well-structured and easy to follow. In addition, OPE will make information about templates easier to find and use. This includes providing a basic content template for users just starting to create books. Ke Li, an intern from BU, is working on developing command-line tools to integrate OPE features. This will normalize the process of creating, publishing, and working on a book. It will also provide templates for features such as tables, making it easier for users to incorporate these elements into their books. As OPE continues to evolve and grow, these new goals and tasks will play a crucial role in ensuring that the framework remains accessible, user-friendly, and effective

for those who want to use it to create and publish open education resources.

CONTINUAL IMPROVEMENT

By providing the necessary tooling and support, OPE makes it easy for educators to utilize the power of open source to build and publish their materials. The project also enables community contribution, review, and verification of the educational content. This ensures that the materials are constantly improved and updated based on the collective efforts of the community. With OPE, educators have first-class support for collaboration, replication, and continual improvement of open source educational content.

Dr. Appavoo has already used the OPE framework to develop lecture notes and lab guides for the BU CAS CS210 course. The response from students and fellow educators has been overwhelmingly positive. The team is now working on making the framework more universal to encourage more lecturers to participate, offer courses, and assist students.

The OPE project is making significant strides in transforming education and making it more accessible and convenient for both educators and students. The combination of open source technologies and a community-driven approach to development and verification ensures that OPE remains at the forefront of educational innovation. With the continued development of OPE, we may see a future where education is more accessible, engaging, and inclusive for all. 

May 2019
Volume 11

RESEARCH QUARTERLY

Bringing great research ideas into open source communities

IN THIS ISSUE:



- Daniel Gruss – What can we do to improve security and resistance to the Spectres and Meltdowns of the future?

Also in this issue:

- IoT: Building an open test automation framework
- The ROSE project: Using open source to bring students together



August 2019
Volume 12

RESEARCH QUARTERLY

Bringing great research ideas into open source communities



How To Train Your Model:
E. Ugur Kaynar's Research Adds
Object Store Caching to Ceph,
Speeds Machine Learning

Building a Linux Unikernel
Has Open Source Made Patents Obsolete?
Where We Came From: Red Hat Research in Brno, Czechia



November 2019
Volume 13

RESEARCH QUARTERLY

Bringing great research ideas into open source communities



Rolling your own processor:
Ahmed Sanaullah builds an open source
toolchain for an FPGA

Keyline: Securing the edge, one slice at a time
The Isolation of Time and Space: Partitioning Hypervisors
The post general-purpose CPU world is upon us



February 2020
Volume 14

RESEARCH QUARTERLY

Bringing great research ideas into open source communities



Finding Flipper
Newcastle PhDs Georgia Atkinson
and Cameron Trotter use deep
learning to identify and count
marine mammals

Using new tools to analyze old data and improve
our picture of the universe
Finding patterns in data, on the fly
Red Hat's Mark Little looks into the future



RH RQ

Bringing great research ideas into open source communities



Kit Murdock
an open source
swashbuckler

Fuzzing hypervisor
virtual devices
Hardware is back
+
Research Day
Europe update



AIOps
Hema
Veeiradi on
Prometheus
Anomaly
Detection



RH RQ

Bringing great research ideas into open source communities



VOYAGE
into the open
dataverse

Open source
cloud operations
Sharing hardware safely
with Elastic Secure
Infrastructure
+
Don't blame the
developers

James Honaker and
Merce Casas on the
privacy balancing act



RH RQ

Bringing great research ideas into open source communities



Václav Matyáš
open source cybersecurity
and the next generation

Machine learning
meets big data
Finding bugs in
parallel programs
+
Mental models



A thread
model:
Daniel Bristo de
Oliveira on the
formal analysis
and verification
of the real-time
Linux kernel



RH RQ

Bringing great research ideas into open source communities



Kate Saenko
minimizing
dataset
bias in AI

Sequential
Monte Carlo
Efficiently verifying
Linux behavior
+
Adaptive learning



Teaching teachers
Changing the world,
one research at a time



RH RQ

Bringing great research ideas into open source communities



Anat Bremner-Barr
when one plus one
makes more than two

Translation layers
for the cloud
Planting research
seeds
+
Demystifying
scheduling latency



New
mentorship
program:
Irit Gollman and
Liora Milbaum
realize potential
when experience
meets passion



RH RQ

Bringing great research ideas into open source communities



Anna Brunström
the right idea at the right time

BigDataStack
delivers
User authentication
+
Faster hardware
through software



Making the most of
research mentorships:
the building blocks of productive
industry-university relationships



RH RQ

Bringing great research ideas into open source communities



Barbora Buhnová
On founding Czechitas and
opening the doors of tech

Constant-time cryptography
The elastic bare metal cloud
Linux-based unikernels
Machine learning &
accessibility



RH RQ

Bringing great research ideas into open source communities



Michael Zink
On shared cloud
computing resources
making research more
accessible and powerful

Optimizing Kubernetes
Ops is the new code
+
Where will we find
the data scientists?



RH RQ

Bringing great research ideas into open source communities



Ayşe Coskun
On machine learning for
operations and how AI can
push analytics to the speed
of software deployment

Adaptive streaming
RISC-V for FPGA
+
Horizon EU: funding
open source research



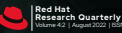
RH RQ

Bringing great research ideas into open source communities



Tomáš Černý
On cross-cultural exchange,
microservice evolution, and
quality assurance

Can streaming data and machine
learning build better communities?
How open data standards
make Brno a better city
Lessons from an upstream
hypervisor fuzzer
+
Verification of a
Linux distribution



RH RQ

Bringing great research ideas into open source communities



**Daniel Gruss and
Martin Schwarzl**
When is it secure enough? Vulnerability research
and the future of vulnerability management

Meet osnoise
+
Preserving privacy in the cloud
Open source research
opportunities abroad



RH RQ

Bringing great research ideas into open source communities



Abhimanyu Gosain
Where are we with wireless? How
researchers are pushing forward
the state of the art, and what that
means for industry

Testing critical IoT systems
+
Measuring open source success
Yuga: A tool to help Rust developers
write unsafe code more safely



Open source through the research lens

As RHRQ starts its fifth year in print, it’s impossible to resist the temptation to look back at all we’ve done so far. The result is this collection of Perspectives.

deas introduced in Volume 1 have made their way from concept to widely available open source solutions, and some of the PhD candidates we mentored in their early research careers are now valued members of the Global Engineering team. We asked a selection of folks associated with Red Hat Research to weigh in on the research and collaborations they considered most interesting over the past five years of its existence.

Each made clear that they were sharing their individual perspective on the activities they know best—there’s no promise of exhaustive coverage or accurate predictions. Nevertheless, together they paint an inspiring picture of the innovative work that can be accomplished when engineering know-how and bold research questions come together in open source environments.

– Shaun Strohmer, Ed. RHRQ

30	Focus on open hardware	By Ulrich Drepper and Ahmed Sanaullah
33	Focus on clouds and research IT	By Heidi Picher Dempsey and Gagan Kumar
37	Focus on testing and operations	By Daniel Bristot de Oliveira and Bandan Das
39	Focus on security, privacy, and cryptography	By Lily Sturmann
41	Focus on AI and machine learning	By Sanjay Arora and Marek Grác
43	Focus on education	By Sarah Coghlan and Matej Hrušovský

Perspectives

Focus on open hardware

by Ulrich Drepper and Ahmed Sanallah

When the research interests that eventually coalesced into Red Hat Research first started, open hardware innovation was not a central feature on our roadmap. We asked Distinguished Engineer **Ulrich Drepper** and Senior Data Scientist (FPGAs) **Ahmed Sanallah** to explain how and why that changed. Uli leads Red Hat's research and future vision on artificial intelligence, machine learning, and hardware innovation, and he first wrote about hardware for RHRQ in "[Hardware is back](#)" (May 2020). Ahmed's current focus is building open source tooling for FPGAs. He received his PhD from Boston University in 2019, and his dissertation—referenced below—won the Outstanding Computer Engineering Dissertation Award. He's written about FPGAs for RHRQ in "[Roll your own processor: building an open source toolchain for an FPGA](#)" (Nov 2019) and "[RISC-V for FPGAs: benefits and opportunities](#)" (May 2022).



Ahmed Sanallah has been writing about FPGAs and open hardware since RHRQ's third issue in November 2019.

FPGAS AND OPEN HARDWARE

When Red Hat Research first started working with Boston University, we were not planning on doing the kind of hardware work we are doing now. Our interests were more focused on what we could do in the realm of container deployment and acceleration. Ahmed proposed his research to Red Hat back in October 2016, and we were trying to use hardware based on FPGAs (field-programmable gate arrays) to do computation on the networking path for cloud nodes. By building support that would allow cloud tenants and providers to develop and deploy FPGA workloads, we were aiming to enable a number of projects, such as network packet processing, metering and telemetry, middleware offloads, and even high-performance computing (HPC) application acceleration such as molecular dynamics and machine learning.

Our use of FPGAs for this was motivated by the additional flexibility, performance, and power/performance ratio FPGAs could give

us. ASICs (application-specific integrated circuits) freeze you into a specific set of features that improves performance and efficiency at the cost of flexibility and upfront cost, while deploying such workloads on CPUs can result in greater flexibility at the cost of reduced efficiency. Moreover, running these workloads on a CPU also reduces resources that can be leased out to tenants. FPGAs, by contrast, basically give you CPU-like flexibility and ASIC-like performance by allowing you to reconfigure the hardware. This results in a persistent state of innovation where you can modify your workloads and consistently update things, all while reducing power requirements and freeing up other datacenter resources.

The problem with FPGAs, specifically, is that the tooling today depends on proprietary software that is also probably 30 years behind the rest of the software world. There was simply never the competition needed to drive improvement in how these tools could be used. Having these tools as part of a larger infrastructure—one where you can press a button and it generates everything in a way that a software developer can understand—was simply not possible. That's why we started to look into the idea of building out systems and specifying applications or supporting application deployment on custom hardware in a way that is useful, not a one-shot design that only you can use and nobody else can benefit from. We founded this effort on Ahmed's thesis work, which explored building infrastructure around an FPGA to support deploying applications in a standard way. Done properly, this

infrastructure will permit the FPGA to be a kind of offload engine in a way that is transparent to the host. All these ideas taken together were the genesis of our hardware research.

To actually achieve this, we had to start from the bottom. There were many open source projects we could piggyback on to do some of these things, but we also needed the building blocks to interface with the system users and also the host system, which we had to build up. You need to communicate from the host system with the FPGA—for instance, through PCIe (Peripheral Component Interconnect-express), through USB, through Wi-Fi— and we are building all these blocks out. Our intention was always that the deployment and application development could be done by someone not steeped in the world of hardware. We are building the tooling to transform inputs from regular software developers, in an easily understandable way, into the appropriate configuration files for all the involved tools without losing any of the efficiencies the low-level tools are granting us.

Where we, as Red Hat Research, can have the most impact is in the area of edge devices. I mean real edge devices—not edge compute at the cell tower, but something that is living next to your machine control system or doing some monitoring at an entry system. These systems are going to be doing more of the work they are supposed to do locally, in a more intelligent way, but in a resource-constrained way. For instance, say you have a door-entry system that takes a photo, sends it back to a

This results in a persistent state of innovation where you can modify your workloads and consistently update things, all while reducing power requirements and freeing up other datacenter resources.

server, does image analysis there, and then sends a signal back. What if, instead, we can deploy the models on an FPGA, which is in the door-entry system itself, and because we are programming it in a highly efficient way, it can make the decisions locally?

This type of system is going to be extremely important, but we can't possibly do it in the current ecosystem of proprietary clouds. Companies like Amazon, Google, and Microsoft provide similar functionality, but then you are exclusively tied into their services. We have to provide an ecosystem outside that.

Our intention was always that the deployment and application development could be done by someone not steeped in the world of hardware.

DRIVEN BY OPENNESS

What we're doing is not so much providing a solution to something as really working on infrastructure and tools the way they should have been designed from day one. The need for an open ecosystem apart from the walled gardens of the proprietary cloud vendors compels us to take a new bottom-up approach toward a completely changeable platform. Once we accomplish that, we'll have a world where more innovation is

possible, and devices are working better with less centralization, so every single device out there doesn't have to phone back to a set of datacenters to make a decision.


Every single day there are more devices out there to do different things. For example, in a monitoring system, you're always adding things, like a new sensor. When you want to do something different, you might have to design a new system. But with an FPGA, you have the flexibility to keep the same system and just connect a couple of wires differently on the board, connect the different sensor, and reprogram. You can make it work—and do it with a lot less energy consumption than even the microcontroller can afford you.

It has also helped that over the last five years it's become evident to more people that FPGAs are not merely implementation and testing devices for a later ASIC design. Now they are recognized as deployments for applications. That was really a prerequisite for this work, that people realize FPGAs are something the general public should be able to utilize by themselves. This started with the first toolchains to program FPGAs to become fully open source, which happened in only the last ten years.

MOVING TOWARD MASS ADOPTION

Looking ahead, we want to advance the state of specialized hardware, such as FPGAs, in a manner that maximizes its practical value to the community. This will require research towards reducing hardware development

overheads to match those of software development, identifying and providing mechanisms for software stacks to leverage specialized hardware, and exploring novel use cases and hardware deployment configurations. To enable and accelerate this research, we are driving an on-premise co-design research lab called CoDes. This lab co-locates hardware, software, researchers, and engineers in a unique collaborative space focused on supporting a completely configured, drivable deployment of arbitrary applications on devices that encompass FPGAs but can also have microcontrollers, sensors, and so on. In the longer term, we are also working on problems that CPU technologies and even GPU technologies are not well suited for.

We are also working on various techniques around confidential computing with BU. So using some of the abilities of FPGAs – for instance, direct communication between multiples of them – you can implement multiparty computation (MPC) that goes much faster than it would on traditional hardware, without the interference of an operating system. If you imagine that you have two applications running that are implementing tasks like MPC, and they have high communication needs, you can cut out any kind of latency if your application talks through a wire directly to the neighboring application. We've already demonstrated this, even on the software side, where we have cut out a normal communications line. Similarly, in the [Unikernel Linux work](#) we can already achieve dramatic speed-ups. With FPGAs, we expect to achieve even better results. Our hope is that the tools we are building can become widely available and achieve mass adoption. 

Focus on clouds and research IT

by Heidi Dempsey and Gagan Kumar

The open cloud has been both cornerstone and North Star for Red Hat Research. Our relationship with the Mass Open Cloud (MOC) and its more recent iteration, the MOC Alliance, has been critical to advancing our understanding of open cloud architecture and the many possibilities it opens for research. (Look no further than [our interview with Red Hat Chairman Paul Cormier and Boston University professor Orran Krieger](#), in this issue, for an illustration.) RHRQ asked US Research Director **Heidi Picher Dempsey** and Red Hat Project Manager **Gagan Kumar** to dive into Red Hat's role in the growth of the MOC and related projects.

INFRASTRUCTURE IN THE MOC: A CASE HISTORY IN COLLABORATION

Heidi Picher Dempsey seeks out and grows research and open source projects with academic and commercial partners. As a network engineer and operation leader, she designed, built, integrated and operated many nationwide suites of prototype cloud infrastructure for academic, government, and industry use. For RHRQ, Heidi has interviewed [Abhimanyu Gosain](#) of the Northeastern Institute for the Wireless Internet of Things and [Dr. Michael Zink](#) of UMass-Amherst and the Open Cloud Testbed and maintains a [column](#) about university-industry research collaboration.

Production New England Research Cloud (NERC) OpenShift container services launched in early 2023, marking a major milestone in a process that began with a research question at Boston University in 2014: would it be possible to build an infrastructure that disaggregates the virtual machines (VMs) in a shared computing environment from the software components that create and manage them, allowing for mix-and-match development of software, hardware, and services that will enable more innovation and new service marketplaces in the cloud?

At the time, VM system components were usually proprietary and so tightly coupled that a single term (e.g., VMware) referred interchangeably to the collection of VMs, the software that managed VMs, and the company that sold VM licenses. BU found ideal collaborators at Red Hat,



RHRQ February 2022 featured UMass Amherst professor Michael Zink on how shared cloud computing resources make research more accessible and powerful.

where engineers were building early production OpenStack cloud computing infrastructure components designed to be open from the operating system up through the stack to the application. The idea became a driving force behind Massachusetts Open Cloud (MOC) development and systems engineering efforts that continue to this day.

Initial goals

From the beginning, the MOC aimed to create an improved computing resource for cloud and big data users and a new model of cloud computing that would enable research and technology companies to innovate and profit in the cloud and big data sectors. This emphasis on building open infrastructure to support both research and industry allowed the MOC to develop a new model where university researchers, open source projects, and research IT groups could collaborate on real systems work with a clear path to transition to active use in industry. What's more, students, researchers, and engineers could work together on building and supporting the infrastructure as a combined team, getting everybody's hands dirty and providing a much better understanding of the challenges involved in transitioning ideas to practice.

The first year of building infrastructure for the MOC at the [Massachusetts Green High Performance Computing Center](#) (MGHPCC) saw the creation of the hardware-as-a-service concept, along with early efforts at automation, a service directory, cloud client libraries, and the first of many front-end Graphic User Interface (GUI) tools. Sixteen Dell servers along with networking and storage hardware

from Intel, Cisco, and Mellanox were jointly built out and operated by the newly formed DevOps team, which, in addition to the original BU proposers, now included engineers from Red Hat, the US Air Force, and Cisco, as well as Harvard and BU's professional research IT groups. The team was building the user community of systems engineers, as well as the community of students, researchers, and big data developers who used tools like Hadoop, PIG, Spark, Mesos, and RabbitMQ on the infrastructure. Although the term "DevOps" had only recently come into use, the MOC had already taken the idea much further, collaborating on all phases of research, development, deployment, and support with a combined team of engineers, academics, and students from each discipline working together toward the same goals. Engineers from Red Hat's Research group were dedicated to working with the team to get a production version of OpenStack software up and running on the new infrastructure, and this became the basis for the first researcher VM service deployed at the MOC.

Although researchers, PhD candidates, and professors were already using the MOC for projects, theses, and courses by 2016, the infrastructure had to expand and grow more reliable for the MOC to become a major mid-scale testbed infrastructure. A storage research cluster, using hardware from Lenovo and Brocade, added SSDs and 500 TB of storage capacity to the MOC. The D3N project started in 2017 and showed it was possible to significantly improve data storage performance by application of caching

changes at multiple levels of the systems stack, in a successful test of the original mix-and-match concept. The DevOps team reached out to security experts to conduct penetration testing and strengthen the overall security of the MOC. The expanded MOC (called Kaizen from the Japanese for "good change") was serving peak loads of 80 VMs to over 100 users.

Enter containers

A funny thing happened while the team was busy building VMs: Kubernetes, which was first released by Google as a seed technology in 2015, grew quickly in popularity. By 2018, Amazon, Azure, and companies like Digital Ocean were offering low-cost managed compute services as an alternative to VMs. Container solutions were quickly taking over market share in managed computing, but once again, the containers were not implemented with open interfaces from the OS through to the application layers of the stack. Unchecked, these closed systems would also block innovation and research across services.

The MOC began to investigate OpenShift as a means to add clusters of containers to their existing VM compute resources, while Red Hat doubled down on their commitment to the idea of open clouds, launching the Red Hat Collaboratory with BU. Work to expand and improve both MOC production testbeds proceeded in parallel. The DevOps team created separate Ops and Prod clusters to enhance reliability and make scaling out more worker nodes in the productions clusters easier. In addition to the bare metal and VM services previously offered,

the team introduced containers and helped develop OpenShift operators for researchers. Production MOC OpenShift container services finally launched in early 2023.

Meanwhile, the MOC organization also expanded, and became the MOC Alliance. Intel and IBM joined, with IBM adding Power9 servers with NVIDIA GPUs to the mix and Red Hat donating 26,000 OpenShift production licenses. MOC researchers proposed an Open Cloud eXchange (OCX), enabling hybrid cloud connections between different providers, and won research funding from the National Science Foundation to implement the new infrastructure. The DevOps team engineered and deployed connections between MOC compute resources and another national US research testbed called CloudLab, joining forces with a Cloud Lab research team from UMass Amherst. This work included building and deploying open source software to allocate and provision resource connections and VLANs dynamically for hybrid clouds. The Hardware Isolation Layer (HIL), Bare Metal Imaging (BMI), and Elastic Secure Infrastructure (ESI) software emerged at various stages from these efforts. MOC teams contributed code back to public open source communities including Ironi, Keylime, IPXE, and TrustedGRUB2 as the hybrid solutions evolved.

Facilitating services

The scope and technology of MOC infrastructure advanced steadily to the point where the total number of MOC users had quadrupled by 2020. Most of those users came from

computer science and engineering backgrounds and could easily patch their own images and write software for the missing bits of a system that they might need. Providing at-scale services that would be useful to any art or science researcher without specialized computer skills required new collaborations with the BU and Harvard research IT groups, who had already faced this challenge with services such as earthquake forecasting, predicting the spread of diseases, and analyzing star formation.

Unchecked, these closed
systems would also block
innovation and research
across services.

The [New England Research Cloud](#) combined with the MOC Alliance to create production cloud resources based on the MOC, using standard deployments and automation that made it possible for other institutions using this as a template to create a full suite of services based on NERCs open methods. The DevOps team added support for Single Sign On (SSO) through the Federated InCommon Identity management service and software already in use at most US universities. The ColdFront GUI allows users to request and manage their own resources, change resource allocations, and raise helpdesk tickets when needed. MOC accounts are integrated with the Keycloak MGHPCC Shared

Services Account portal and use a common MGHPCC OSTicket system to make it easier to track and close issues with multiple teams and users. The team used Red Hat's Advanced Cluster Management to standardize monitoring and alerting, and Ansible playbooks to make deployments more easily repeatable. Finally, they created reporting and billing software that sent cloud usage metrics from OpenShift and OpenStack into XDMoD software from the National Science Foundation, creating reports that most researchers were already familiar with from previous projects.

The NERC team provides hands-on facilitation and technical expertise for research end users, and the DevOps team works together with NERC to resolve issues, provide needed new features, and improve future services. Using this model, the MOC Alliance and NERC can now support projects such as the Newspapers Database Project, which makes 20 million articles from newspapers.com and the US Library of Congress available to history and political science researchers. No one really calls this big data research anymore: researchers now take as a given large-scale data and the computing infrastructure necessary to process it.

The ability to create open hybrid clouds that allow people to create and add their own software and innovate anywhere in the software stack using open interfaces has evolved from an idea to a practical reality. New technology and services that demand new infrastructure are evolving even more quickly now than when the MOC first came into being, as evidenced by


the topics in the latest [MOC Alliance Workshop](#). We're working on new infrastructure to support machine learning, use core-to-edge protocols for computing with wireless endpoints, collect data from sensors in the wild, and help grade school students learn to read. The future dreamers are still out there, and they're certainly welcome to come build with us!

DO MORE WITH LESS

Gagan Kumar focuses on projects related to the MOC Alliance, bare metal sharing systems, and metrics collection for OpenShift instances. Gagan provided an update on the multiyear Elastic Secure Infrastructure (ESI) project, "[The elastic bare metal cloud is here](#)" (Nov 2021), and is a Senior Product Manager with Project Curator, an infrastructure consumption analysis project for the OpenShift platform.

When Red Hat Research started, distributed systems, cloud computing, security, and operating systems were our primary focus. These are fields where many Red Hatters have significant experience and can help university researchers fast-track their ideas in frontier technologies to the real world in an open source way. Since then, we've achieved three significant milestones in building the relationships necessary to make this happen. First, we've helped develop community research clouds through the MOC Alliance (MOC-A) and the New England Research Cloud (NERC). Second, Red Hat Research has established its presence in many prestigious universities in the United States, Europe, and Israel. Boston University, Masaryk

University, Newcastle University, and Technion have labs and office spaces dedicated as collaboration spaces for researchers in those universities and Red Hat engineers to come together to discuss ideas and implement projects. As part of the MOC-A and NERC partnership, we also work with researchers from Northeastern University, Harvard, MIT, and the University of Massachusetts. The third milestone is the research interest community the Red Hat Research team has built, which gives Red Hatters the opportunity to engage with research ideas and research projects at many levels. As a result, Red Hat engineers are working closely with researchers to accelerate progress in areas of critical interest.

Since most computation is moving toward cloud computing, research is now focused not only on cloud services' efficacy but also their efficiency. Certain research domains like intelligent cost management, on-demand resource sharing, and remediation techniques such as AIOps and MLOps are coming to the forefront of the research field. This is reflected in a number of projects managed by Red Hat, including [ESI](#), Project Curator, [OS-Climate](#), the [Cloud Cost Optimizer \(CCO\)](#), and [AI for CloudOps](#). As another field of growing importance, edge computing is an opportunity to extend the open hybrid cloud all the way to the data sources and end users. Data might have traditionally belonged in the datacenter or cloud, but many important decisions need to happen out here—on the edge. This technology will open opportunities for many advancements. 

Researchers now
take as a given
large-scale data
and the computing
infrastructure
necessary to
process it.

Focus on testing and operations

by Daniel Bristot de Oliveira and Bandan Das

Red Hat Research has fostered work on testing and analysis that started as open source explorations and ended as valuable upstreamed resources for anyone to use. We asked two engineers who've worked on highly successful projects, **Daniel Bristot de Oliveira** and **Bandan Das**, to share some of the biggest research accomplishments so far and let us in on what we can expect in the next three to five years.

REAL-TIME LINUX

Red Hat engineer and researcher **Dr. Daniel Bristot de Oliveira** has delivered several practical improvements to the Linux kernel, including a [Real-Time Linux Analysis \(RTLA\) toolset](#) in the Linux 5.17 kernel release and a [runtime verification subsystem](#) in the Linux kernel 6.0. Daniel published work on the formal analysis and verification of the real-time Linux kernel in a series of three articles, "[A thread model for the real-time Linux kernel](#)" (Oct 2020), "[Efficient runtime verification for the Linux kernel](#)" (Feb 2021), and "[Demystifying real-time Linux scheduling latency](#)" (May 2021). Recently, Daniel wrote about the development of osnoise, in "[Meet osnoise, a better tool for fine-tuning to reduce operating system noise in the Linux kernel](#)" (Nov 2022).

Five years ago, the vision of Linux as a real-time operating system for safety-critical systems was nothing more than a motivation idealized by researchers in academic papers. Not that real-time Linux did not exist; indeed, the vast majority of features composing the real-time Linux kernel were already there, for example, the PREEMPT_RT and SCHED_DEADLINE. The use of Linux in embedded systems was also a reality. The primary obstacle was the challenges imposed in the certification of Linux for safety-critical applications.

The rise of edge computing helped drive the development of the community around Linux for safety-critical systems, mainly motivated by the automotive industry. This initiative was led by the Linux Foundation's ELISA (Enabling Linux In Safety Applications) group and industrial players such as BMW, Bosch, and Red Hat. This



Daniel Bristot de Oliveira began a series of research articles about the real-time Linux kernel in the October 2020 issue of RHRQ.

trend leveraged the research and development of methods and tools to aid in the analysis of Linux, which is the missing link between embedded and safety-critical Linux. Over these years, Red Hat Research actively helped in this field, motivating researchers to improve the safety aspects of real-time Linux by using sophisticated analysis of the Linux kernel. For example, academic research developed together with Scuola Superiore Sant'Anna (IT) and Universidade Federal de Santa Catarina (BR) led to the creation of the runtime verification subsystem and the RTLA toolset, both integral parts of the Linux kernel.

We expect to see growth in the number of publications that tackle safety aspects in the Linux kernel. We foresee research involving languages that include safety as a native aspect, such as Rust and eBPF; the application of AI in the creation of models to be used in the verification of Linux properties; and the use of more complex formal languages to verify the timing properties of real-time Linux schedulers.

FUZZING THE LINUX KERNEL

Bandan Das is a software developer in the virtualization group at Red Hat. He worked on the project *"Fuzzing device emulation in QEMU"* at the Red Hat Collaboratory at Boston University, with a team that included Red Hat engineers Stefan Hajnoczi and Paolo Bonzini and BU professor Manuel Egele. The project sought to develop a novel method for fuzzing virtual devices and implement it in the popular open source QEMU hypervisor packaged in most Linux distributions. (Fuzzing is a powerful technique for dynamically generating and executing randomized test cases.)

Bandan mentored PhD candidate and research associate Alex Bulekov (BU 2023), who documented the team's successes in two articles for RHRQ, "Fuzzing hypervisor virtual devices" (May 2020) and "Applying lessons from our upstream hypervisor fuzzer to improve kernel fuzzing" (Aug 2022).

In the early days of Red Hat Research, we encouraged engineers to approach PIs at BU and other universities to brainstorm ideas for collaboration. Systems, FPGAs, testing, and education were areas of focus, and several of these projects have stabilized with concrete research goals, upstream contributions, and academic publications. While cloud computing was still the talk of the day, when outlining our goals for the QEMU fuzzing project, we never thought we would deeply integrate our fuzzing infrastructure in the cloud. Today, we extensively use Google's oss-fuzz project, which runs fuzzing on upstream QEMU in the cloud. Additionally, we took advantage of fuzzing and sanitizer improvements to LLVM to improve our fuzzing framework. VM Snapshot fuzzing emerged as a powerful approach to fuzzing complex software, which encouraged us to develop our snapshot fuzzer for kernel fuzzing—a key difference with existing kernel fuzzers such as Syzkaller.

Among our successes, we developed and upstreamed the current state-of-the-art fuzzing approach for hypervisors. The upstream fuzzer has continued to identify bugs across a wide range of virtual devices (including virtual I/O devices often used in the cloud). This fuzzer **identifies and patches serious**

bugs before they make it into a release—a capability that benefits all downstream QEMU users. The novelty of our approach led to our paper's acceptance at Usenix Security 2022, which has an 18% acceptance rate. We also developed FuzzNG, a kernel fuzzer that is competitive even when compared with the large, established Syzkaller project. We can fuzz most of the Linux subsystems that Syzkaller can fuzz, with virtually no human effort. As far as I am aware, no other public fuzzer has this capability. Our paper was accepted to the NDSS Symposium 2023, another Tier 1 security conference, which has about a 15% acceptance rate.

Kernel and hypervisor fuzzing techniques are evolving with active work ongoing at companies such as Google, Microsoft, and Apple. In addition to our collaboration with BU, we have kept track of the recent **US Executive Order** on increasing static analysis and fuzzing of software used by the US federal government and setting minimum standards for code verification by developers. We also have interest from companies such as Yandex, Oracle, and Google that have helped shape the fuzzing project to the state it's in today.

In the next three to five years, we expect to see better tooling to run fuzzers continuously for open source software. As opposed to typical applications, operating systems and hypervisors have complex interfaces that are event driven and are encapsulated deep within multiple layers of abstractions. We aim to efficiently fuzz these interfaces and increase coverage. **RH**

Focus on security, privacy, and cryptography

by Lily Sturmann

RHRQ asked **Lily Sturmann**, a senior software engineer at Red Hat in the Office of the CTO in Emerging Technologies, to look back at the past few years of research in the area of security and privacy research and share her perspective on the future. Lily has primarily worked on security projects related to remote attestation, confidential computing, and securing the software supply chain. She has contributed frequently to the Red Hat Next blog, and her most recent article for RHRQ was [“Preserving privacy in the cloud: speeding up homomorphic encryption with custom hardware”](#) (Nov 2022).

Security and privacy have moved further to the forefront of technical concerns over the past several years, as more organizations make use of hybrid cloud environments encompassing machines and devices on-premises, on-cloud infrastructure, and at the edge. The reliability of cryptographically backed integrity and confidentiality guarantees on infrastructure owned by third parties is critical for those running sensitive workloads in these environments, and cloud providers have responded by making technologies like confidential computing and trusted platform modules (TPMs) available on their infrastructure.

Similarly, the privacy challenges involved with interorganizational collaboration and computation on large, sensitive data sets have motivated innovative work on multiparty computation and fully homomorphic encryption. Throughout its history, RHRQ has highlighted the strides made by researchers in these privacy-enabling fields, as in an [interview with Harvard researchers James Honaker and Mercè Crosas](#) (Aug 2020) on making open

source solutions for storing and sharing richly detailed information about experiments, software, and systems more available to all.



In the August 2020 RHRQ, James Honaker and Mercè Crosas spoke to us about differential privacy and secure multiparty computing.

High-profile vulnerabilities and attacks related to software dependencies have been manifesting ever more frequently.


RHRQ's very first issue in May 2019 featured the work of side-channel attack researcher Daniel Gruss of Graz University of Technology (Austria). With Research Director Hugh Brock, Daniel wrote about "[Taking a proactive and holistic approach to cybersecurity](#)." Calling for a more vibrant exchange between security researchers and engineers, Daniel discussed Spectre and Meltdown, two CPU vulnerabilities that brought attention to the widespread practice of viewing security only at the level of individual components. Daniel and collaborator Martin Schwarzl were featured again in an interview with Red Hat Vice President of Product Security Vincent Danen. "[When is it secure enough?](#)" (Nov 2022) elaborated on the importance of taking a system-wide view of security, especially at the intersection of software and hardware. Daniel and Martin's latest research on side-channel attacks is introduced in the "[Research project updates](#)" section of this issue.

Another critical area is supply chain security in software, as demonstrated by the series of [Executive Orders](#) from the US government. This is especially apt with the high-profile vulnerabilities and attacks related to software dependencies that have been manifesting ever more frequently, the most notable being the vulnerability in [Log4j](#) discovered in 2021. Masaryk University PhD candidate Agáta Kružiková's article on "[User authentication for open source developers](#)" (Aug 2021) in public open source repositories describes one of the many challenges in the complex software supply chains that we all depend on. The [Sigstore](#) project has gained recognition for helping to make



The [November 2022 RHRQ](#) included an interview conducted by Red Hat Vice President of Security Vincent Danen and topics including vulnerability management and homomorphic encryption.

signatures in the open source software supply chain standard practice, with the first-ever Sigstorecon held in 2022 as a co-located event at KubeCon Detroit.

Looking ahead, we will continue to face challenges like the growing power of quantum computing and Shor's algorithm potentially making much of the cryptography we rely on today obsolete. But we also have many new engineering tools available, from wider adoption of the security-oriented [Rust programming language](#) to [WebAssembly](#) to the novel ways we can protect code and data integrity and confidentiality across heterogeneous environments. While security is by definition a constantly evolving field, the power of open source software and open research to meet these challenges and create opportunities remains constant. 

Focus on AI and machine learning

by Sanjay Arora and Marek Grác

Red Hat Research focuses on accelerating the practical applications for artificial intelligence and machine learning by combining academic approaches and industry use cases. Rather than focusing purely on advancing AI/ML techniques, we identify research collaborations where they can play a central role in solving computing problems. The AI/ML projects we've highlighted in past issues drive innovation across multiple disciplines, from optimizing hybrid cloud operations to enhancing education, developing technology for autonomous vehicles, and automating methods for detecting visual disinformation online.

We asked **Sanjay Arora**, a Red Hat Research data scientist, and **Marek Grác**, a Red Hat senior software engineer and lecturer in machine learning at Masaryk University (Brno, CZ), to share their perspectives on trends in AI/ML, past, present, and future. Sanjay has contributed to RHRQ in the articles [“When good models go bad: minimizing dataset bias in AI”](#) (Feb 2021) and [“Yuga: a tool to help Rust developers write unsafe code more safely”](#) (Feb 2022). Marek has contributed to the RHRQ stories [“‘When one teaches, two learn’: making the most of technical research mentorship”](#) (Aug 2021) and [“Making machine learning accessible across disciplines”](#) (Nov 2021).

One of our tasks at Red Hat Research is persuading engineering managers and business units that it is worth engineers' time to participate in research projects, internships, and mentoring students working on a thesis. Cooperation with universities has led to collaboration in research and demonstrated its applications for products and the open source community, making persuasion easier. The range of topics is broad, from online learning and community management to cloud technologies and testing.

AI/ML is an excellent example of how that relationship works. The learnings we gain from research and teaching at universities are something we can also share internally. Our



The February 2021 issue featured Kate Saenko, Boston University professor and consulting professor for the MIT-IBM Watson AI lab, on minimizing dataset bias in AI.


The Red Hat OpenShift Data Science (RHODS) team is providing the infrastructure for IBM efforts to train large models.

efforts to work on AI/ML projects with university partners have led to a broader understanding of the role of AI/ML in industry. The Applied Machine Learning course we teach at Masaryk University has content we want to teach in-house as well, so software engineers, quality assurance teams, and other roles know how to work with machine learning people to get the best possible results.

We foresee several different directions for further work in the field of AI/ML itself and in the many domains where it plays a key role:

- **Secure multiparty computing and differential privacy:** The capacity to use sensitive datasets without revealing information about individuals in the dataset is critical for expanding the use of AI/ML systems. A good example is the recently launched BU Collaboratory project “[Co-ops: collaborative open source and privacy-preserving training for learning to drive](#).” The project is building out a privacy-preserving platform for sharing data collected from cars and videos that can be used for distributed, large-scale training of models for self-driving.
- **Large ML models and their applicability data,** including telemetry, logs, and error messages: Training large language models (LLM) on this unannotated data could enable better predictions. Generating natural language responses from a knowledge base could also support helpdesk technicians and automate a portion of their work. Current developments

in this arena include GPT-4 and Chat GPT, which are closed source. The Red Hat OpenShift Data Science (RHODS) team is providing the infrastructure for IBM efforts to train large models, which could lead to work on applying these models for better searching, querying Red Hat Insights, or integrated development environments (IDEs).

- **Replacing heuristics with learned policies in systems software:** Systems software like operating systems and compilers have a lot of heuristics that are used to guide decision making. These decisions affect performance and resource consumption. Red Hat Research is engaged in several projects exploring the replacement of these heuristics with learned policies using techniques like reinforcement learning and Bayesian Optimization. Two projects—“[Automatic configuration of complex hardware](#)” and “[Toward high performance and energy efficiency in open source stream processing](#)”—involve learning network policies governing packet batching and processor voltage and frequency settings to enable substantial energy savings while maintaining performance guarantees. (See also Han Dong’s article “[Tuning Linux kernel policies for energy efficiency with machine learning](#)” in this issue.) The project “[Practical programming of FPGAs with open source tools](#)” focuses on searching for the optimal ordering of compiler passes to maximize performance for the compiled code. 

Focus on education

by Sarah Coghlan and Matej Hrušovský

Enabling hands-on, experiential opportunities for students at multiple learning levels has been a mainstay of the Red Hat Research mission. Mentoring students in open source development, teaching classes, creating curriculum, and contributing to education infrastructure are all ways of growing a robust open source research community. That in turn benefits students, the companies that hire them, universities, and—ultimately—the end users of open source technologies. **Sarah Coghlan** is the university program manager for Red Hat in Boston, and **Matej Hrušovský** is the university program manager for Red Hat in Brno, Czech Republic, home to Red Hat's largest engineering office. RHRQ asked them to discuss the wide range of activities in this vital area.

UNIVERSITY-BASED PROGRAMS

When Red Hat Research launched, university programs were largely local and designed around community needs. That is evident in the variety of programs we have supported, whether by supplying funding or contributing expertise. While not an exhaustive list, this sampling of past and present educational programs shows the depth and diversity of initiatives Red Hat Research has participated in:

Undergraduate research

- **GROW: Greater Boston Research Opportunities for Young Women:** Young women entering their senior year of high school participate in collaborative research at Boston University.
- **SoarCS and RAMP (Research, Academic, and Mentoring Pathways):** [UMass Lowell students](#), particularly first-generation college students and students from groups underrepresented in STEM fields, get mentoring in research and develop programming and other skills the summer before their freshman year.

- **Collaboratory student research projects:** [Boston University undergraduate students](#) contribute to computer systems research



Masaryk University professor Barбора Buhnová explained why diversity leads to better problem solving in the November 2021 issue of RHRQ.

The pandemic highlighted the need to augment open source learning resources and target a broader remote audience.

at the Red Hat Collaboratory in projects related to Unikernel Linux, practical programming of FPGAs, and security detection.

Open source and programming

- **Red Hat Beyond:** Red Hat engineers launched [this project](#) to promote open source and DevOps concepts among students in Israel. Beyond courses have been taught at Reichman University and other regional universities, and for the Israeli Navy.
- **Red Hat open source education (ROSE):** Arab and Israeli students learn about Linux, open source, and Python programming together in this long-running program.
- **Red Hat Summer Camp:** This IT-focused camp teaches Brno high school students about [open source values](#) and technical skills including coding, UX design, and Git.
- **Social Innovation hackathons:** Partnerships with groups including UNICEF, the Southern Coalition for Social Justice, and Boston's Children's Hospital inspired community coding efforts to use the power of open source to address global issues.

Diversity in STEM careers

- **Czechitas thesis award:** This non-profit organization, hosted by Masaryk University and [co-founded by professor Barbora Buhnová](#), encourages the participation of young women in tech. The thesis award provides mentorship for young women passionate about working in IT.
- **FIT summer school for girls:** Red Hat Czech is a partner in this program that encourages

high school girls interested in IT, based at the Brno University of Technology.

- **Leadership academy:** This program, [launched by UMass Amherst](#) in collaboration with several other Massachusetts colleges and universities, develops professional skills with students of color and women interested in careers in technology and engineering.
- **TechTogether hackathons:** Hackathons in major US cities encourage people of marginalized genders to become part of the hackathon community, contributing to an increase in the gender diversity of hackathons from 18% in 2017 to 46% in 2022.

Many Red Hatters also teach for-credit classes at universities where Red Hat Research has a presence, and others teach Linux Kernel Development through the Linux Foundation. Red Hat has sponsored undergraduate internships, awarded scholarships, and funded research opportunities for undergrads at BU, Northeastern, the University of Massachusetts, and several other US universities for many years.

RESOURCES FOR FACULTY AND INSTRUCTORS

With the formation of a global research team, our educational activities started to align. The COVID-19 pandemic also had a significant impact: many of our university and high school activities at the time were hands-on and in person, and not all of them could be run remotely. On the other hand, the pandemic highlighted

the need to augment open source learning resources and target a broader remote audience.

Red Hatters and faculty frequently collaborate on developing course materials related to open source and core research areas of interest. In 2022, we began aggregating the resources that Red Hatters and our university partners have used to teach courses on topics including cloud computing, Linux administration, and technical writing. A [growing online database](#) representing universities from North America and Europe makes these materials easy to find and available to anyone for reuse.

In the next few years, we'll likely hear more about the Open Education project (OPE) on the Red Hat OpenShift Data Science (RHODS) environment at BU. OPE aims to empower educators in any discipline to create, publish, and collaboratively develop high-quality educational materials that students can access with just a web browser. See Danni Shi's article "[Open source education: from philosophy to reality](#)" in this issue to learn more about where this initiative is headed.

TRAINING AND WORKFORCE DEVELOPMENT

Especially in the United States, the traditional path to careers in app-dev, data science, IT, and site reliability engineering (SRE) has been through college. Many of these jobs can be done by skilled people who do not have a college degree, and the typical college education does not always prepare candidates well for real-world work. Additionally, a college education is not easily




attainable for many talented candidates because of social barriers in our education system.

In January 2023, Red Hat Research launched an apprenticeship program with the Massachusetts Green High-Performance Computing Center (MGHPCC). The program provides an entry point into IT for talented individuals who either would not take the traditional college path or are switching from other, non-technical subjects. This program sources student interns from Springfield Technical Community College (STCC), a public, not-for-profit institution in an underserved community in western Massachusetts.

Students who have excelled in the Computer Systems Engineering Technology program have the opportunity to complete an internship with TechSquare, a high-performance

computing (HPC) Linux System Administration consulting group.

The Red Hat Research team engages with students who successfully complete the apprenticeship. While at Red Hat, students work to gain experience as SREs, directly working on Red Hat projects.

Red Hat/MGHPCC apprentices will participate in the integration and operations of open source software and systems in distributed hybrid clouds, learn how to transition research and prototype systems to production environments, and learn how to support real production environments in research IT and enterprise datacenters. Students in the program will be able to support multiple collaborative projects with academic and industry partners now underway in research environments worldwide. 

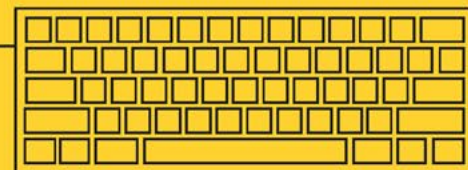
Your research,



projects



and education partner.





A data-driven approach for analyzing Common Criteria and FIPS 140 security certificates

Seccerts is a much-needed tool for data scraping and analysis of security certificates, but creating it was harder than expected. Here's why.

by Petr Švenda and Jaroslav Řezník

Security certification documents from certification schemes like Common Criteria (CC) and the National Institute of Standards and Technology (NIST) Federal Information Processing Standard (FIPS) contain valuable, detailed information. Most of it, however, is not machine readable. Extracting value from these loosely structured PDF documents is a tedious, labor-intensive, error-prone manual process.

If only the documents were properly structured and automatically processable!

Our laboratory at the Center for Research on Cryptography and Security (CRoCS) aimed to write a suite of processing tools to extract basic information from public certification documents and allow for a more automated search for potentially vulnerable ones. We were soon exposed to the intricate complexities



About the Author
Petr Švenda

is a security researcher and teacher at Masaryk University, Czech Republic.

He first touched the domain of security certifications in 2002 while working on side-channel analysis of cryptographic devices for the Czech government and has kept his passion for cryptographic smartcards ever since.



About the Author
Jaroslav Řezník

is a Principal Program Manager responsible for Red Hat's government certifications under the Product Security Compliance and Risk team. In his almost 15 years at Red Hat, he has touched many different areas from very different angles, from the community work on Fedora that is still his passion to compliance with government standards like Common Criteria and FIPS.

A QUICK PRIMER ON CERTIFICATION

Most consumers are familiar with Payment Card Industry (PCI) data security standards, which ensures that credit/debit card payments are processed according to agreed-upon security requirements. EVM specifications (Europay, MasterCard, and Visa) fulfill a similar purpose for cards with smart chips.

CC certification was developed to provide an international computer security standard for products used in defense and government intelligence. In the United States, NIST developed FIPS to establish guidelines for the security of federal computer systems, and many industries in the private sector also voluntarily use these standards. For example, FIPS 140-2 establishes the security requirements for cryptographic modules.

of CC and FIPS certification schemes and realized what we thought would be a quick task would take much longer. We also needed industry expert knowledge and insight.

This was the genesis of the seccerts project, now available at the website seccerts.org for easy browsing and github.com/crocs-muni/sec-certs for more advanced users wanting to self-host and utilize a Python-based API.

FUNCTIONS OF THE SECCERTS PORTAL

The CRoCS laboratory first looked into CC certification documents out

of necessity rather than curiosity. We found a serious vulnerability allowing factoring of RSA keys generated by cryptographic smartcards certified to levels as high as CC Evaluation Assurance Level 6+¹ and a timing side-channel attack allowing extraction of private ECDSA (elliptic curves-based digital signature algorithm) keys.² The certification documents were the most detailed source of the information we needed without signing non-disclosure agreements, but they were difficult to use

The vulnerability disclosure process highlighted another issue: when using a composite product, will users be notified when a single component is found to be vulnerable? The implications of this problem are substantial. For example, the Estonian government learned of the vulnerability in their electronic citizenship cards less than two months before national elections despite the vulnerability in the underlying chip being privately communicated to major customers for half a year.

To mitigate these problems, CRoCS set several goals for developing a suite of tools to analyze security certification documents. Some goals were clear from the beginning; others gained importance as we dug deeper into the ecosystem. These are the most important ones:

Make existing information more

available: The portal uses artifacts from existing certification schemes and the National Vulnerability Database. It provides additional insight by processing, connecting, and overlaying these data sources. Processed data is visualized and available in JSON format for further processing. Python-based API and example Jupyter notebooks are prepared for instant analysis. The seccerts portal is extensible to other schemas or databases in the future.

Provide deeper insight into certification ecosystem trends:

The certification process evolves over time, with different actors adopting potentially different strategies during the certification procedure. The data-driven approach may provide insight into how items are certified, which certification claims are used more or less frequently, the type of items certified, which security and cryptographic mechanisms are used, and other factors. As a more lightweight and agile certification scheme requires us to change or omit some existing steps, understanding the security impact of existing steps is crucial.

Utilize open data and tools for better transparency and

accessibility: Open source, freely available tools, and a data-driven

¹ ROCA vulnerability, CVE-2017-15361. See Matus Nemec, Marek Sys, Petr Svenda, Dusan Klinec, and Vashek Matya, "The return of Coppersmith's attack: practical factorization of widely used RSA moduli," in 24th ACM Conference on Computer and Communications Security (CCS'2017), p. 1631–1648, 2017.

² Minerva vulnerability, CVE-2019-15809. See Jan Jancar, Vladimir Sedlacek, Petr Svenda, and Marek Sys, "Minerva: The curse of ECDSA nonces (systematic analysis of lattice attacks on noisy leakage of bit-length of ECDSA nonces)," in IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020(4): 281–308.

approach provide better accessibility for end users by extracting the most relevant data otherwise hidden in the certification documents. Open availability also increases the transparency of the certification process by making it easier to verify claims and compare different certificates.

Facilitate better end user verifiability of the purchased certified product: In many cases, end users have limited options for verifying whether the product is genuine. Standard shallow identifiers, such as serial numbers, can easily be tampered with. To mitigate this, the seccerts portal may instead host authenticated forensic profiles based on harder-to-manipulate behavioral properties, such as detailed performance profiling or power consumption traces of certified devices created by a trusted authority using open tools. The user will later collect the same properties from the purchased product using the same open tools and compare them to the expected forensic template. Increased end user verifiability should increase product scrutiny by performing repeated checks over time and performed by many different users. Even new product tests can be performed and shared with others, increasing end user confidence.

Enable faster notification to end users in case of a new potential vulnerability: Pairing certified items and their dependencies (referenced certificates) with the platform identifier in the vulnerability database allows push notification for relevant changes, like the occurrence of a new vulnerability, in the user-selected set of certificates. The

seccerts portal provides notifications thanks to an extracted graph of references and National Vulnerability Database (NVD) mapping.

Aid vulnerability research:

Information aggregated from multiple sources allows a more exhaustive and efficient search for clues about related and potentially vulnerable (certified) products and provides better insight into a specific vulnerability's impact. A discoverer of the vulnerability quickly identifies other companies for targeted responsible disclosure.

While not all documents and certification artifacts are publicly available, those that are provide a trove of interesting technical data commonly used in domains listed above, but in a rather ad-hoc and often labor-intensive way. Our goal is to systematize and make these efforts easier while learning something along the way.

Open availability also
increases the transparency
of the certification process
by making it easier to verify
claims and compare
different certificates.

RESULTS ACHIEVED, ISSUES ENCOUNTERED

To tackle these issues, we had to process available source documents and build large sets of regular expressions

for topics of interest (e.g., certificate IDs, cryptographic algorithms, protection profiles, security standards, security functional and assurance claims, defenses used, and many more). We then had to devise heuristic methods to pair related documents, including records from different databases. During the process, we obtained some insight into the pain points of the certification process.

To determine the accuracy of the automated extraction and analysis, we had to rely on some ground truth and expert knowledge. We utilized many years of our own experience with security analysis of cryptographic smartcards, which constitutes a significant fraction of all devices certified under the CC scheme.³ Our collaboration with Red Hat engineers allows us to get deep insight and possibly verify data for several other categories, including operating systems. We also manually labeled large testing and validation datasets to pair certified products and records in the NVD.

These are the most critical systematic issues we identified:

The human element: Certificate reports are written by humans for humans, and it shows. Documents are in different languages with human-made typos. Typos hinder automatic processing, forcing the use of heuristics to correct the errors—always missing some and possibly introducing new

³ The category "ICs, Smart Cards, and Smart Card-Related Devices and Systems" contains around 35% (568 out of 1606) of all currently active certificates. Common Criteria, Certified Products List—Statistics, commoncriteriaportal.org/products/stats, June 06, 2022.



The Red Hat lab at the Faculty of Informatics, Masaryk University (Brno, CZ), was the first of several on-campus labs Red Hat maintains in Brno and Boston.

ones. Even the format of a fundamental piece of certificate information—unique certificate IDs—is left to separate national schemes. Some have well-structured but mutually different naming schemes containing the year of certification, incremental number, and optional revisions, while others adopt only the incremental number. This makes it hard to unambiguously reference and automatically extract these references.

Fuzzy boundaries of certified items:

The Target of Evaluation (ToE) – that is, the part of the item that actually undergoes certification – is specified in a somewhat informal way, making it difficult for automatic processing to establish the ToE boundaries. The software equivalent of the Bill Of Materials (BOM) was proposed as a partial solution, but the task is more complicated: ToE covers not only the parts of the certified product but also the circumstances and environment

of use. Fuzzy boundaries make it difficult for end users to verify whether the product used is in the certified configuration or to properly pair and evaluate the vulnerabilities reported.

Vulnerabilities: A connection of certified items to external sources, especially vulnerability databases, is ambiguous at best and almost impossible to establish at worst. We manually labeled thousands of certificates just to train the mapping heuristic classifier of a certified item to the Common Platform Enumeration (CPE) record used in the National Vulnerability Database, still resulting in only 90% accuracy.

Design for replicability: Evaluation labs function as trusted mediators that allow assessment of certification claims based on proprietary (non-public) documents. The steps taken,

tools used, and results obtained by evaluation labs are frequently confidential. Moreover, the level of expertise found in specialized evaluation laboratories is not available to all end users. Public documents, like the maintenance reports, also frequently lack the data used during issuance. As a result, certification is primarily a one-off exercise not independently verifiable later. And because vendors are sponsoring the evaluation of their own products, conflicts of interest may arise.

**PROBLEMS FOR PRODUCT
RELEASE**

Overall, security certifications help raise the bar of product quality. However, there are at least two ways that they can make product releases more difficult. First, frequent changes in standards and requirements in otherwise rigorous certification processes make keeping certifications up to date challenging, both from a technical perspective and for planning a release timeline. In the case of FIPS 140-2/140-3, the problem is how slow the cryptographic module validation process became. It's not uncommon for modules to stay on the so-called Modules in Process (MIP) list for many months, leading to many new uncovered vulnerabilities while the module is in the review queue. For example, Red Hat's strategy is to keep CC certificates updated with every Extended Updates Support (EUS) release. The default lifecycle of the non-EUS release does not align with the certification process, as the release is supported only for six months. A product must receive a certificate within six months; however, testing has to be performed on the final General Availability code level. The standard six-month lifecycle

is not long enough to finish the certification within these boundaries. Any delays in the certification process increase the risk of important vulnerabilities and can jeopardize the whole project.

The second problem is that many certification schemes essentially require products to contain no known vulnerabilities. The National Information Assurance Partnership (NIAP) requires a vulnerability search no older than 30 days, which may cause last-minute changes in the evaluated product.

Just as significant, many found CVEs can be justified in one of several ways:

- CVE is already fixed.
- CVE could be mitigated by the evaluated configuration.
- CVE might not be applicable; for example, affected hardware is not available in the evaluated configuration.

Hence the naive interpretation of this data is almost impossible and may lead to many false positives. For example, the ToE for a product may be a tiny subset of the standard product offering. But as ToE boundaries and CVE record scope are difficult to establish automatically, the seccerts project outputs some false positives. Similarly, a human evaluator must filter the vulnerabilities report during the certification period.

MAKING CERTIFICATION BETTER, FASTER, AND CHEAPER

Addressing these issues will be difficult but necessary to make certification usable for frequently changing products. We have a few suggestions:

- Provide data in standardized, automatically processable formats—sanitized, normalized, and directly available.
- Document boundaries of certification with a clear, human- and computer-readable structure that enables running and verifying a system in the same environment as the certified one.
- Include a fully automated installation process in guidance documentation to ensure the system is remediated into the appropriate configuration.
- Proactively assign a CPE record in the NVD to every issued certification, using unique and robust identifiers.
- Make available tooling (ideally open source) and complete documentation of the configuration and parameters used by evaluation labs.


Automation of certification testing has already proven to be a viable means of improving the certification processes. FIPS's Automated Cryptographic Validation Testing System (ACVTS) is one of the first real, production-ready attempts to automate security certifications. It replaced the semi-manual Cryptographic Algorithm Validation System (CAVS) in June 2020. A client-server infrastructure using the ACVP protocol (JSON-based) has been added on top of the old-style CAVS test harness, and algorithm certificates are issued automatically. ACVP (the protocol) only covers the algorithm-testing part of FIPS 140-2/140-3 validations and even alone can speed up the validation process. Turnaround to get the algorithm certificates required for module validation is almost


instant; module validation is still the biggest portion of validation.

CONCLUSIONS

We remain convinced that our goals are worth pursuing further, despite being significantly more challenging than anticipated. Interest from national certification bodies, industry, security researchers, and the portal webpage has grown, resulting in discussions that explain some of the problems and open new questions

No silver bullets exist for such a complex environment. Still, based on our experience, we believe data analysis provides compelling insights and highlights issues that, when solved, will help improve the studied certification schemes. We believe more transparent and available certification data are helpful to all parties involved—especially vendors, regulators, and end users—despite their different interests. We hope the seccerts project provides useful base data to facilitate improvements to existing certification schemes to make them faster, cheaper, and more accessible. It can also help end users using certified products right now by getting more from the promised benefits of security certifications.

Finally, and maybe most importantly, certification bodies should conduct periodic assessments of the impact of certification on the security of products certified, just as we planned to do as academic researchers. The recommendations above would then become a natural prerequisite for completing such evaluation and hopefully result in a more transparent certification process with more value added. 



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.



Project updates

Research project updates

Each quarter, *Red Hat Research Quarterly* highlights new and ongoing research collaborations from around the world. This quarter we highlight collaborative projects from Graz University of Technology (Austria), Masaryk University (Czech Republic), and Karlstad University (Sweden). Contact academic@redhat.com for more information on any project described here, or explore more research projects at research.redhat.com.

PROJECT: Researching and mitigating the exposure of modern and efficient technology to side-channel attacks**ACADEMIC INVESTIGATORS:**

Prof. Daniel Gruss, Dr. Martin Schwarzl, Jonas Juffinger (Graz University of Technology)

RED HAT INVESTIGATORS:

Wade Mealing, Andrea Arcangeli

Modern systems implement numerous optimizations related to data structure and content directly. This yields an increase in performance and efficiency. In this project, we investigate how modern and efficient technology introduces side-channel vulnerabilities and how to mitigate these vulnerabilities, with a focus on remote attack scenarios. Recent results include discovering a novel side channel on AMD processors, exploiting scheduler queue contention. The root cause is the design of the scheduler queue in AMD processors, which is optimized for a higher degree of parallelism with schedulers per execution unit. An attacker can exploit the

contention on the scheduler queues to infer the contention level per execution unit and, thus, what other workloads on the system are doing (published at IEEE Symposium on Security and Privacy 2023).

We also collaborated with the University of Virginia and Cornell University to find side channels in modern persistent memory technology (published at USENIX Security 2023). At the overlap area between hardware and software, we developed a novel templating technique to localize side-channel leakage in software. Our technique, layered binary templating, can scan even large binaries in a reasonable time. We identified previously unknown leakage in Chrome and Chrome-based apps that enable leaking any keystroke performed in Chrome or Chrome-based apps with a hardware-based side channel such as Flush+Reload or a software-based side channel such as page cache attacks. The specific leakage has been



Red Hat works with TU Graz to research and mitigate the exposure of modern and efficient technology to side-channel attacks.

patched, and the concrete attack is now mitigated. However, layered binary templating can still find new leakage in other software binaries (published at ESORICS 2023).

On the pure software level, following up on our earlier work on remote page deduplication attacks, we analyzed the exposure of compression algorithms to remote timing attacks. Our attacks specifically exploit timing differences in compression and decompression but do not exploit the compression ratio or other metrics previous works identified as security-critical. Our work sheds light on the necessity of shielding compression algorithms against side-channel attacks or avoiding compression of sensitive data altogether (published at IEEE S&P 2023).

Another outcome of this research was Martin Schwarzl's successful, and excellent, defense of his PhD thesis. Martin will now move into industry, and his role in the project will transition to Jonas Juffinger, who has collaborated with Martin and will continue the research in this project, potentially with a higher focus on security and side channels caused by efficiency-related optimizations.

PROJECT: Side-channel attacks on embedded devices and smartcards

ACADEMIC INVESTIGATORS:

Tomáš Jusko, Ján Jančár
(Masaryk University)

Elliptic curve cryptography (ECC) is difficult to implement securely, especially regarding various side-channel attacks in which an attacker observes side channels such as power

consumption or timing of an ECC implementation while it computes. These attacks often require the attacker to have precise knowledge of the implementation choices made by the target, which they might not have in practice. To better understand these attacks, we built an open source toolkit, `pyecsca`, for side-channel attacks on ECC, focusing on reverse-engineering ECC implementations. [Pyecsca is available on GitHub.](#)

There are two notable ongoing efforts on the toolkit. The first is adding support for CPU emulation to the toolkit, enabling it to produce simulated side-channel traces for target ECC implementations and allowing easier attack prototyping and implementation evaluation. The second is to implement selected trace processing algorithms using GPGPU methods (e.g., CUDA) to speed up lengthy trace processing when operating on very large datasets of traces.

PROJECT: Verifying constant-time cryptographic algorithm implementations

ACADEMIC INVESTIGATORS:

Ján Jančár, Vashek Matyáš
(Masaryk University)

Timing attacks are among the most devastating side-channel attacks, allowing remote attackers to retrieve secret material, including cryptographic keys, with relative ease. In principle, avoiding these attacks is not hard, as it means developing constant-time code or using tools to verify constant-timeness. Yet, these attacks still plague popular cryptographic libraries 25 years after their discovery, reflecting a dangerous gap between academic

research and cryptographic engineering. To understand the causes of this gap, we surveyed 44 developers of 27 prominent open source cryptographic libraries. The survey aimed to analyze if and how the developers ensure that their code executes in constant time.

To follow up on the survey, we conducted a user study into the usability of constant-timeliness verification tools to better understand where their usability issues lie. We are currently in the process of analyzing the results of this user study.

PROJECT: Building the next generation of programmable networking—powered by Linux

ACADEMIC INVESTIGATORS:

Prof. Anna Brunstrom, Dr. Per Hurtig, Frey Alfredsson, and Simon Sundberg (Karlstad University)

RED HAT INVESTIGATORS:

Toke Høiland-Jørgensen, Jesper Dangaard Brouer, and Simone Ferlin-Reiter

Developing continuous passive network monitoring using BPF is progressing well. A tool for inferring TCP (and ICMP) RTTs directly from application traffic named evolved Passive Ping (ePPing) has been developed, and an initial version is available at [the XDP project repository on GitHub](#). Using BPF to monitor the packets directly in kernel space, ePPing avoids the packet capture overhead of similar solutions, such as Wireshark and PPing. The paper “Efficient continuous latency monitoring with eBPF” (presented at the Passive and Active Measurements Conference [PAM] 2023) evaluates ePPing and shows that it has much

lower overhead than the PPing tool it was inspired by, allowing it to scale to high-speed network links.

Additional work on aggregating the numerous RTT measurements

from ePPing in an efficient and informative way is ongoing. We are also looking to evaluate ePPing from an ISP vantage point. Feedback on the tool or suggestions for other use cases are welcome. [RH RO](#)

NEVER MISS AN ISSUE!

Available
in PDF and
printed
version



SUBSCRIBE NOW

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

red.ht/rhrq



AI ON INTEL®



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

Learn more at ai.intel.com