The Next Generation Cloud:

The Rise of the Unikernel

A Xen Project publication
April 2015

xenproject.org

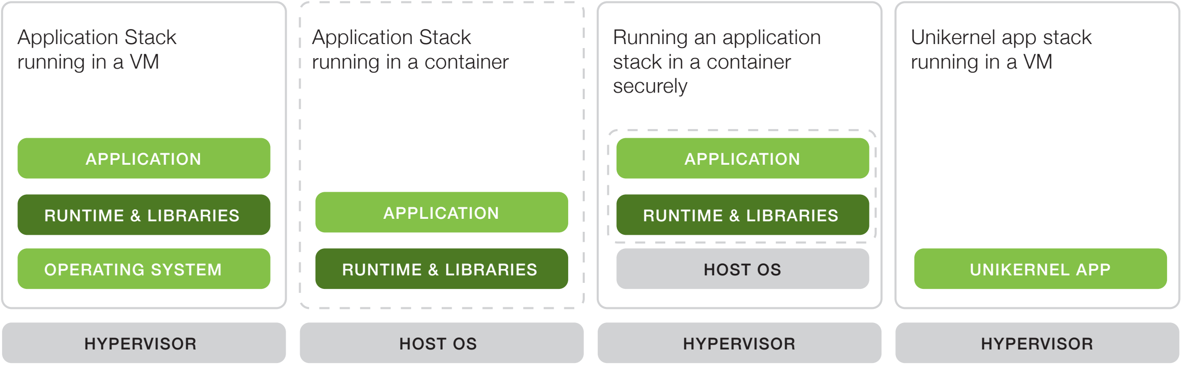
Docker and Linux container technologies dominate headlines today as a powerful, easy way to package applications, especially as cloud computing becomes more mainstream. While still a work-in-progress, they offer a simple, clean and lean way to distribute application workloads.

With enthusiasm continuing to grow for container innovations, a related technology called unikernels is also beginning to attract attention. Known also for their ability to cleanly separate functionality at the component level, unikernels are developing a variety of new approaches to deploy cloud services.

Traditional operating systems run multiple applications on a single machine, managing resources and isolating applications from one another. A unikernel runs a single application on a single virtual machine, relying instead on the hypervisor to isolate those virtual machines. Unikernels are constructed by using "library operating systems," from which the developer selects only the minimal set of services required for an application to run. These sealed, fixed-purpose images run directly on a hypervisor without an intervening guest OS such as Linux.

As well as improving upon container technologies, unikernels are also able to deliver impressive flexibility, speed and versatility for cross-platform environments, big data analytics and scale-out cloud computing. Like container-based solutions, this technology fulfills the promise of easy deployment, but unikernels also offer an extremely tiny, specialized runtime footprint that is much less vulnerable to attack.

There are several up-and-coming open source projects to watch this year, including **ClickOS**, **Clive**, **HaLVM**, **LING**, **MirageOS**, **Rump Kernels** and **OSv** among others, with each of them placing emphasis on a different aspect of the unikernel approach. For example, MirageOS and HaLVM take a clean-slate approach and focus on safety and security, ClickOS emphasizes speed, while OSv and Rump Kernels aim for compatibility with legacy software. Such flexible approaches are not possible with existing monolithic operating systems, which have decades of assumptions and trade-offs baked into them.



How are unikernels able to deliver better security? How do the various unikernel implementations differ in their approach? Who is using the technology today? What are the key benefits to cloud and data center operators? Will unikernels on hypervisors replace containers, or will enterprises use a mix of all three? If so, how and why? Answers to these questions and insights from the key developers behind these exciting new projects are covered in this paper.

Unikernels Can Improve Internet Security

Many industries are rapidly moving toward networked, scale-out designs with new and varying workloads and data types. Yet, pick any industry — retail, banking, healthcare, social networking or entertainment — and you'll find security risks and vulnerabilities are highly problematic, costly and dangerous.

Adam Wick, creator of the The Haskell Lightweight Virtual Machine (HaLVM) and a research lead at Galois Inc., which counts the U.S. Department of Defense and DARPA as clients, says 2015 is already turning out to be a break-out year for security.

"Cloud computing has been a hot topic for several years now, and we've seen a wealth of projects and technologies that take advantage of the flexibility the cloud offers," said Wick. "At the same time though, we've seen record-breaking security breach after record-breaking security breach."

The names are more evocative and well-known thanks to online news and social media, but low-level bugs have always plagued network services, Wick said. So, why is security more important today than ever before?

The creator of MirageOS, Anil Madhavapeddy, says it's "simply irresponsible to continue to knowingly provision code that is potentially unsafe, and especially so as we head into a year full of promise about smart cities and ubiquitous Internet of Things. We wouldn't build a bridge on top of quicksand, and should treat our online infrastructure with the same level of respect and attention as we give our physical structures."

In the hopes of improving security, performance and scalability, there's a flurry of interesting work taking place around blocking out functionality into containers and lighter-weight unikernel alternatives. Galois, which specializes in R&D for new technologies, says enterprises are increasingly interested in the ability to cleanly separate functionality to limit the effect of a breach to just the component affected, rather than infecting the whole system.

For next-generation clouds and in-house clouds, unikernels make it possible to run thousands of small VMs per host. Galois, for example, uses this capability in their **CyberChaff project**, which uses minimal VMs to improve intrusion detection on sensitive networks, while others have used similar mechanisms to save considerable cost in hardware, electricity, and cooling; all while reducing the attack surface exposed to malicious hackers. These are welcome developments for anyone concerned with system and network security and help to explain why traditional hypervisors will remain relevant for a wide range of customers well into the future.

Madhavapeddy goes as far to say that certain unikernel architectures would have directly tackled last year’s Heartbleed and Shellshock bugs.

“For example, end-to-end memory safety prevents Heartbleed-style attacks in MirageOS and the HaLVM. And an emphasis on compile-time specialization eliminates complex runtime code such as Unix shells from the images that are deployed onto the cloud,” he said.

The MirageOS team has also put their stack to the test by releasing a “Bitcoin pinata,” which is a unikernel that guards a collection of Bitcoins. The Bitcoins can only be claimed by breaking through the unikernel security (for example, by compromising the SSL/TLS stack) and then moving the coins. If the Bitcoins are indeed transferred away, then the public transaction record will reflect that there is a security hole to be fixed. The contest has been running since February 2015 and the Bitcoins have not yet been taken.

Linux Container vs. Unikernel Security

Linux, as well as Linux containers and Docker images, rely on a fairly heavyweight core OS to provide critical services. Because of this, a vulnerability in the Linux kernel affects every Linux container, Wick said. Instead, using an approach similar to a la carte menus, unikernels only include the minimal functionality and systems needed to run an application or service, all of which makes writing an exploit to attack them much more difficult.

Cloudius Systems, which is running a private beta of OSv, which it tags as the operating system for the cloud, recognizes that progress is being made on this front.

“Rocket is indeed an improvement over Docker, but containers aren’t a multi-tenant solution by design,” said CEO Dor Laor. “No matter how many SELinux policies you throw on containers, the attack surface will still span all aspects of the kernel.”

Martin Lucina, who is working on the Rump Kernel software stack, which enables running existing unmodified POSIX software without an operating system on various platforms, including bare metal embedded systems and unikernels on Xen, explains that unikernels running on the Xen Project hypervisor benefit from the strong isolation guarantees of hardware virtualization and a trusted computing base that is orders of magnitude smaller than that of container technologies.

“There is no shell, you cannot exec() a new process, and in some cases you don’t even need to include a full TCP stack. So there is very little exploit code can do to gain a permanent foothold in the system,” Lucina said.

The key takeaway for organizations worried about security is that they should treat their infrastructure in a less monolithic way. Unikernels allow for the careful management of particularly critical portions of an organization’s data and processing needs. While it does take some extra work, it’s getting easier every day as more developers work on solving challenges with orchestration, logging and monitoring. This means unikernels are coming of age just as many developers are getting serious about security as they begin to build scale-out, distributed systems.

Multiple Unikernel Uses Cases Emerging

Early adopters are using the technology to run web sites, critical systems infrastructure, cutting-edge research or to operate as a network appliance. MirageOS, for example, is serving as a successful testbed for cutting-edge research at the University of Cambridge and other academic groups, while Galois' clients use HaLVMs for a number of network services and functions.

"One of our clients used a combination of HaLVMs to provide a reliable, secure VPN solution for laptops. Internally, we have also used HaLVMs to implement a variety of network services, including encryption nodes, random number generators, and network sensors," Wick said.

OSv runs on AWS and is so popular that the beta program is already over-subscribed. Laor believes many uses cases will benefit from OSv, which offers superior I/O performance, manageability and ease of use. Caches, load balancers, NoSQL and other I/O intensive workloads are ideal targets for OSv, in his opinion.

At an even simpler level, many systems can be improved through the use of a few strategically placed unikernels, according to Galois. Why not insert a HaLVM that performs quick spot checks of all incoming data before passing it on to the server? If your system is sensitive to changes in load, why not insert a MirageOS unikernel that can perform rate limiting? Want to switch to SSL, but your server doesn't support it — why not add a LING converter?

"Many breaches start with a hacker sending invalid messages to a server that has not been properly implemented. All of the above situations are ones in which the flexibility and scalability of unikernels can really shine, and I believe we will start to see people taking advantage of them over the next year," said Wick.

Madhavapeddy's group is working on a new toolstack called Jitsu (Just-in-Time Summoning of Unikernels), which can start a unikernel in ~20ms in response to a network request.

"This lets us run millions of sleeping unikernels that awaken in response to a network request and live for a few seconds at a time. We're calling this sort of infrastructure 'dust clouds' and expect that it will dramatically change the economics of hosting on the cloud," he said. Jitsu will be presented at the USENIX NSDI conference this May in Oakland, California.

Amir Chaudhry, who leads the Nymote.org project based on MirageOS unikernels said, "The coming era of hyper-elastic clouds using MirageOS and Jitsu means that users do not have to run large, always-on VMs. Instead, users can provision services and applications only when there is demand, scaling out and back down automatically. This enables people to maintain a secure, personal online presence for a few dollars a year, adding additional services as desired. All without giving up their personal data to third-party services or having to become SysAdmins."

Containers and Unikernels — Friends or Foes?

Will enterprises deploy a mix of VMs, unikernels and containers? Or will unikernels eventually go mainstream and replace containers? Identifying the best set of technologies for an organization depends on the end goals, experts say. In some cases though, unikernels may very well be the technology of choice in the future.

“Docker is great when you want to put together a number of functions into a single component. If you want a LAMP stack, you’re probably better off just using a LAMP Docker instance and pressing ‘go.’ On the other hand, if you want a lightweight, single-service component that you can bring up and down quickly, or want to scale massively, then unikernels are going to be a clear winner,” Wick said.

New options for developers and SysAdmins are a certainty, as unikernels and container technologies are quickly evolving and hypervisors are branching into new areas such as embedded computing and ARM-based servers. This actually creates new opportunities across the board, according to Cloudius.

“Unikernels provide the best of all worlds – on the one hand they retain the rich hypervisor ecosystem and enable superior isolation, live migration and robust SLA. On the other, unikernels provide container-like properties such as sub-second boot time, density and simplicity,” Laor said.

Madhavapeddy believes unikernels and Linux container technologies are highly complementary to one another. In his opinion, numerous combinations will emerge with hypervisors still the technology of choice for securing multi-tenancy environments.

“I also expect to see a unikernel backend for Docker in 2015 that will enable developers to partition a particular workload across unikernels and Linux VMs,” Madhavapeddy said. “We will also see improved compatibility between the unikernel stacks as the interconnect standards settle down, enabling multiple language runtimes such as Java, OCaml, Go, Rust and Haskell to each run inside a VM and form a secure distributed system of unikernels.”

Lucina, who is focused on providing compatibility with existing applications, points out that Docker and hypervisors operate at different technology layers, so one will not replace the other. He sees Rump Kernels as a Docker alternative in the future.

“Rump Kernel-powered unikernels can run existing software — Nginx, PHP and MySQL were all ported with little effort,” Lucina said. “Once we work out the remaining challenges in usability, I envision Rump Kernels replacing Docker for deploying services in many scenarios.”

Laor acknowledges that some organizations will want to simplify and stick with a single technology. By following Docker’s format as closely as possible with OSv, he hopes sophisticated users won’t have to compromise on a single technology. With unikernel projects focused on finding a balance between security, performance and portability, unikernels will likely play an important role when deploying any future networked infrastructure. Here’s a closer look at key projects to watch in the coming months.

7 Game-Changing Cloud Technologies

ClickOS a high-performance, virtualized software middle box platform based on open source virtualization. Early performance analysis shows that ClickOS VMs are small (5MB), boot quickly (as little as 20 milliseconds), add little delay (45 microseconds) and more than 100 can be concurrently run while saturating a 10Gb pipe on an inexpensive commodity server.

Clive is an operating system designed to work in distributed and cloud computing environments.

HaLVM The Haskell Lightweight Virtual Machine (HaLVM) is a port of the Glasgow Haskell Compiler tool suite that enables developers to write high-level, lightweight VMs that can run directly on the Xen Project hypervisor.

LING is highly compatible with Erlang/OTP and understands .beam files. Developers can create code in Erlang and deploy LING unikernels. LING removes the majority of vector files, uses only three external libraries and no OpenSSL.

MirageOS Incubated by Xen Project, MirageOS is a clean-slate library operating system that constructs unikernels for secure, high-performance network applications across a variety of cloud computing and mobile platforms. There are now more than 60 MirageOS libraries and a growing number of compatible libraries within the wider OCaml ecosystem. With recent improvements to the toolchain and an increasing number of contributors, MirageOS makes it easier than ever to “compile your own cloud.”

OSv is a new OS designed specifically for cloud VMs from Cloudius Systems. Able to boot in less than a second, OSv is designed from the ground up to execute a single application on top of any hypervisor, resulting in superior performance, speed and effortless management. Support for C, Java, Ruby, node.js, and Scala application stacks available as well as future suport for Golang.

Rump Kernels provide free, portable, componentized, kernel quality drivers such as file systems, POSIX system call handlers, PCI device drivers, a SCSI protocol stack, virtio and a TCP/IP stack. These drivers may be integrated into existing systems, or run as stand-alone unikernels on cloud hypervisors and embedded systems.

Additional Resources

Tech Republic article “**Unikernels Offer a Striped Down Version of Linux**” by Nick Hardiman.

Recent presentation from Galois: “**Unikernels: Who, What, Where, When and Why**.”

Erlang on Xen — use cases for the new Erlang platform LING that is at the heart of the elastic cloud.

Slide deck from Puppet Labs’ Gareth Rushgrove titled “**The End of the General Purpose Operating System**.”

Silicon Angle article “**MirageOS: Platform for Launching Self-Contained Applications on Top of a Hypervisor**” by Saroj Kar.

Unikernel performance research from Ian Briggs, Matt Day, Eric Eide, Yuankai Guo, and Peter Marheine at the University of Utah. Their preliminary paper, which includes OSv performance data, is “**Performance Evaluation of OSv for Server Applications**.”

Xen Project Evangelist Russell Pavlicek’s recent presentation “**The Next Generation Cloud: The Rise of the Unikernel.**”

“**Containers vs Hypervisors: The Battle Has Just Begun**” article by Pavlicek.

Introduction to the **MirageOS Bitcoin Pinata** by Amir Chaudhry.

“**Rumprun for Rump Kernels: Instant Unikernels for POSIX Applications**,” presented at New Directions in Operating Systems by Martin Lucina.

To learn more about The Xen Project software please visit us at www.xenproject.org.

To learn more about MirageOS, please visit openmirage.org

The Xen Project is a Linux Foundation Collaborative Project that develops the Xen Project Hypervisor and its associated subprojects.