nanovms.com

nanovms

WHITEPAPER

NanoVMs IN DEPTH

INTRODUCTION

Security issues remain a key challenge in cloud adoption while the ever increasing need for more software drives cost and complexity up. Unikernels are widely acknowledged as the future of cloud infrastructure yet they remain inaccessible to most organizations.

"Unikernels might kill containers in five years."

SINCLAIR SCHULLER
CEO
APPRENDA

NanoVMs is the only unikernel platform available today. Unikernels have been widely acknowledged as the future of cloud infrastructure for a variety of reasons. Unikernels have security concerns architected from the ground up. Unikernels offer higher performance at a lower operational cost.

However, the existing legacy big public cloud providers are not a great fit for unikernels. Also compilation and orchestration tooling have been previously lacking.

NanoVMs provides an entire suite of tooling to compile and orchestrate unikernel web applications with a click of a button. NanoVMs negates the need to muck around with Makefiles, figure out what libraries need to be included or hire expert level systems engineeers.

SECURITY

One of the main benefits of using unikernels is the in-depth security model they provide. Unikernels come with an encompassing four point security model. They are single process systems. They don't have the notion of users. They don't come with shells. Lastly, they have a dramatically reduced attack surface.

Unikernels are single process systems. By design they can only run a single program per virtual machine. This means it is quite literally impossible to launch other programs on the same virtual machine. This immediately puts an end to so-called "shell code exploits" in particular and remote code execution attacks in general. Even the system calls to execute other programs do not exist. There is no security model quite as thorough as the unikernel model.

Unikernels do not have the concept of users. Rather than allowing end users to execute code on production machines configuration and administration tasks are moved to the compilation and deployment stage of an application life cycle. This prevents accidental configuration changes by administrators and also prevents rogue hackers from executing code as well.

Unikernels do not have shells either. This prevents hackers from logging into production machines. The mechanism does not exist. There is no way to install extra software on end production machines without re-deploying the software application. This is by design and prevents attacks like the Equifax Apache Struts attack.

Lastly, unikernels have a dramatically reduced attack surface. Compared to a Linux system there are millions of lines of code less. Thousands of libraries are not present. Hundreds of other processes are not present. The unikernel is comprised of only the end application and the necessary software to make it run and nothing else. This severely limits any potential attack from occurring as attackers are forced to attack the end application and not other parts of the system.

PERFORMANCE

NanoVMs unikernel platform can make applications go screaming fast. The architectural design of unikernels bring several advantages that are not possible in old outdated Linux systems. Unikernels have little to no systems calls, little to no context switching, faster boot times, and can use less system resources.

In Linux 4.X there are 399 system calls. Windows has close to 700. Some unikernel implementations have no concept of system calls while others have only a few. This is important because this dramatically affects the performance of programs. Every packet that comes across the network and every disk read or disk write is a hit.

Context switching can occur when the host operating system switches from process to process or from kernel to user mode. There is a heavy cost that is incurred on each context switch and it can happen quite a lot. Since unikernels are single process systems by definition they do not pay this penalty. This allows unikernels to perform more work faster.

Unikernels can boot incredibly fast. Compared to a normal virtual machine that can take seconds to boot unikernels can boot in 50ms. Some unikernel researchers are now booting unikernels in 2-3ms. That is only slightly slower than a fork system call on Linux and more than two orders of magnitude faster than Docker.

All of these performance improvements together allow your software to use less system resources leading to a direct reduction in your overall infrastructure operating cost.

COST REDUCTION

Unikernel platforms offer both capex and opex reduction. On the capital expenditure side fewer servers are needed to accomplish the same task as fewer resources can be consumed by a far greater number of virtual machines. This is simply not possible with traditional legacy virtual machines. Today's cloud environment is fundamentally different from the PDP-7 and PDP-11s that Unix was designed on. Even as developers go out of their way to isolate applications and design secure solutions they can not escape the forty years of sordid inefficiencies and vulnerabilities that plague servers today. Being able to run thousands of virtual machines on the same commodity hardware as you might run tens of today is a true game changer and can save your company a lot of money.

On the operational side of things fewer devops engineers are needed to configure, administer, and deploy unikernels as they have a lot less "moving parts". Since configuration is done at deployment time fewer mistakes are made and security issues become less of a concern. An incredible amount of time and energy is put into creating cloud environments that can provision hundreds to thousands of virtual machines. This is only necessary because instead of adding in only what you need to make your applications work as unikernels do developers have to guess and strip away at pieces of the operating system they think they don't need. At the end it usually becomes a compromise between time, security, and ease of understanding.

Unikernels explicitly declare their dependencies to alleviate the guesswork, reduce the time investment, and eliminate the security unknowns.

"Unikernel-based payloads don't have command shells to exploit. They don't have utilities which can be subverted. And they don't have full operating systems with documented risks which can be compromised."

RUSSEL PAVLICEK
CLOUD SOLUTIONS ARCHITECT
REDHAT

ORCHESTRATION

NanoVMs has an orchestration system built from the ground up to support unikernel clouds. This orchestration system is built with unikernel design concepts such as:

- Configuration Management Deploy Workflows
- Hot Volume Swapping
- Migration Workflows
- Sub 100ms boot times
- Server-less
- Functions as as a Service

Since unikernels are single process systems orchestrating applications can be different than normal VM or container environments. For example a typical web application might involve an application server and a database that while in a traditional cloud environment probably runs on different VMs has to run on different VMs in a unikernel environment.

Likewise organizations taking advantage of serverless and functions-as-a-service environments might have resorted to hacks in the past to build APIs that reflect that. Unikernels, being small , and fast to boot can readily provide the same functionality in a much more secure fashion using dramatically fewer resources. It's trivial to boot thousands of VMs in seconds and spin them back down immediately using unikernels - not so easy in the big public clouds.

COMPILATION

Unikernel compilation is extremely hard if you aren't using the correct tools. Unikernels are the composition of the application and the operating system together as one virtual machine. Developers find compiling and linking the correct libraries into a unikernel by hand extremely complicated using off the shelf tooling and guessing at the parts that need to be glued together.

Figuring out what libraries need to be linked into the application is another hard task as typically a system administrator or devops engineer will do this by hand in legacy Linux systems.

Unikernel compilation is very different from regular application compilation. Indeed many languages don't "compile" at all. NanoVMS not only compiles the software but "cross-compiles" the application into an end bootable machine that can be utilized by a unikernel platform.

NanoVMS looks at dependencies that traditionally come on a host operating system such as where to put the TLS certificates. NanoVMS looks at your application's native library dependencies. For instance libxlst and libxml are commonly found in every single interpreted language such as php and ruby. This is typically available as part of the base operating system or as an artificat that a devops engineer would install. Then NanoVMS looks at the language specific dependencies. Whether that involves maven or sbt with java and scala or bundler with ruby or npm with javascript we can install the langugage specific dependencies automatically ensuring that things like the underlying database library works.

NanoVMS simplifies the compilation and library linking into an easy to use automated tool that automatically discovers the application dependencies needed for a given application.

REDUCE DEVOPS

Devops teams are more in demand now than ever as organizations take on more digital initatives requiring more highly skilled engineers to automate and define software infrastructure. Unfortunately, high performing devops engineers are expensive. NanoVMS's unikernel platform reduces the need and demand for devops engineers as infrastructure is immutable by default and defined up front in the application lifecycle.

Configuration management software has always been a pain for devops and sysadmins. Whether it's older software such as chef,puppet, and ansible or newer container/cloud native software such as terraform the ability to make accidental changes to important production deployments is always such a common problem. Unikernels really embrace the concept of immutable infrastructure by baking in configuration to the VM at compile time and utilizing immutable volumes by default.

Application orchestration or application deployment systems continue to be a struggle for many organizations as more and more workloads are enabled. Options such as Kubernetes and Docker Swarm have attempted to solve this problem for so-called cloud native options but leave legacy applications with no solution. Having said that using a cloud native solution almost always requires a large competent team of devops engineers because of the constellation of available software and the necessary knowledge to glue it all together.

CLOUD AND INFRASTRUCTURE INTEGRATION

NanoVMs can easily slide into your existing on-premise cloud infrastructure or you can run it in the cloud as a managed service. This includes compute, networking, storage and more.

"Unikernels are small and fast and give Docker a run for its money, while at the same time still giving stronger features of isolation"

FLORIAN SCHMIDT
RESEARCH SCIENTIST
NEC LABORATORIES EUROPE

Unikernels today are not great fits for the big public clouds. Performance is dramatically reduced by having to choose between nested virtualization or waiting on the legacy cloud deployment patterns to provision unikernels. Cost scales much higher as the number of resources needed to deploy unikernels in a safe performant manner are much higher on the public clouds.

Thankfully unikernels fit in perfectly well with existing on premise cloud installations and managed unikernel solutions. Whether your organization uses VMWare ESX, Xen, KVM, or no hypervisor the NanoVMs unikernel platform can transparently slide in well. OpenStack and other private Infrastructure as a Service offerings work well with NanoVMs too.

ABOUT NANOVMS

NanoVMs provides the only unikernel platform available today. NanoVMS allows IT teams to run and assemble their various projects as unikernels using their existing infrastructure.

NanoVMs provides enterprise-ready solutions to secure your applications, lower cloud spend, and increase performance.

NanoVMs offers corporate unikernel training, development, migration, consulting and offers a fully turnkey managed unikernel platform.