

CPOSC 2014

The Next Generation Cloud: The Rise of the Unikernel

Russell Pavlicek
Xen Project Evangelist
Russell.Pavlicek@XenProject.org

About the Old, Fat Geek Up Front

- Linux user since 1995; became a Linux advocate immediately
- Delivered many early talks on Open Source Advocacy
- Former Open Source columnist for Infoworld, Processor magazines
- Former weekly panelist on “The Linux Show”
- Wrote one of the first books on Open Source: Embracing Insanity: Open Source Software Development
- 30 years in the industry; 20+ years in software services consulting
- Currently Evangelist for the Xen Project (employed by Citrix)
- Over 75 FOSS talks delivered; over 150 FOSS pieces published

The Cloud We Know

- Field of innovation is in the orchestration
 - The Cloud Engine is paramount (OpenStack, CloudStack, etc.)
 - Workloads adapted to the cloud strongly resemble their non-cloud predecessors
 - Some basic adaptations to facilitate life in the cloud, but basically the same stuff that was used before the cloud
 - Applications with full stacks (operating system, utilities, languages, and apps) which could basically run on hardware, but are run on VMs instead.
 - VMs are beefy; large memory footprint, slow to start up
 - It all works, but its not overly efficient
 - 10s of VMs per physical host

The Next Generation Cloud

- Turning the scrutiny to the workloads
 - Should be easier to deploy and manage
 - Smaller footprint, removing unnecessary duplication
 - Faster startup
 - Higher levels of security
 - 1000s of VMs per host

The New Stuff: Docker & Containers

- Makes deployment easier
- Smaller footprint by leveraging kernel of host
- Less memory needed to replicate shared kernel space
- Less disk needed to replicate shared executables
- Really fast startup times
- Higher number of VMs per host

Docker Downsides

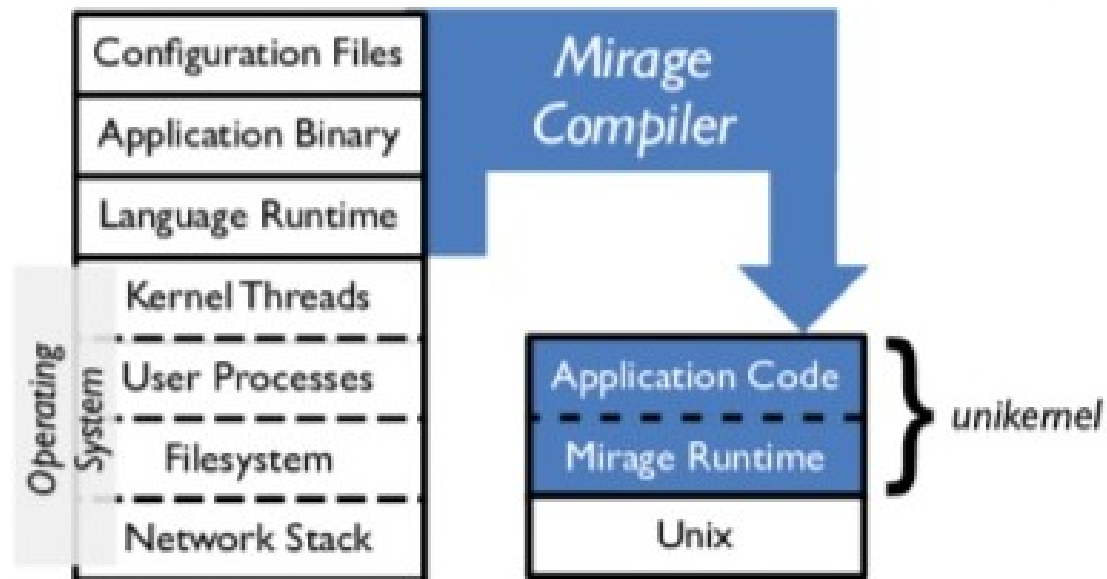
- Improvements, yes; but not without issues
 - Can't run any payload that can't use host kernel
 - Potential limits to scalability
 - Linux not really used for 1000s of processes
 - Security
 - Security is a HUGE issue in clouds
 - Still working on real security; someday...
 - At LinuxCon North America, Docker CEO doesn't even identify security as one of the top priorities
 - Google & others run Docker in VMs when they need security

The Unikernel: A Real Cloud Concept

- Very small
- Very efficient
- Very quick to boot
- And very, VERY secure!
- Many unikernels already exist, including Mirage OS, a Xen Project Incubator Project

Unikernel Approach: Mirage OS

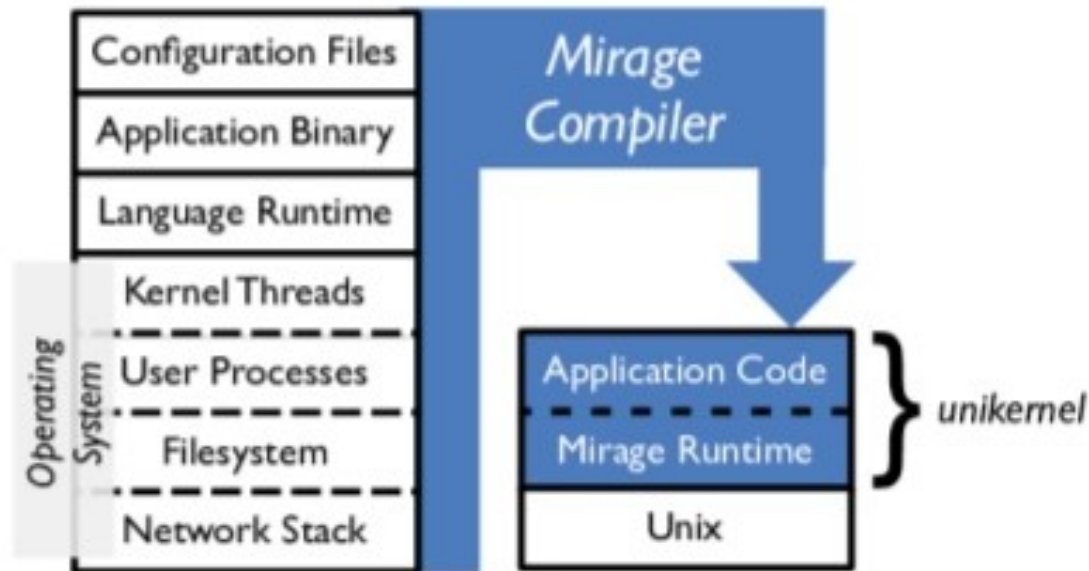
Swap system libraries to target different platforms:
develop application logic using native Unix.



Unikernel Approach: Mirage OS

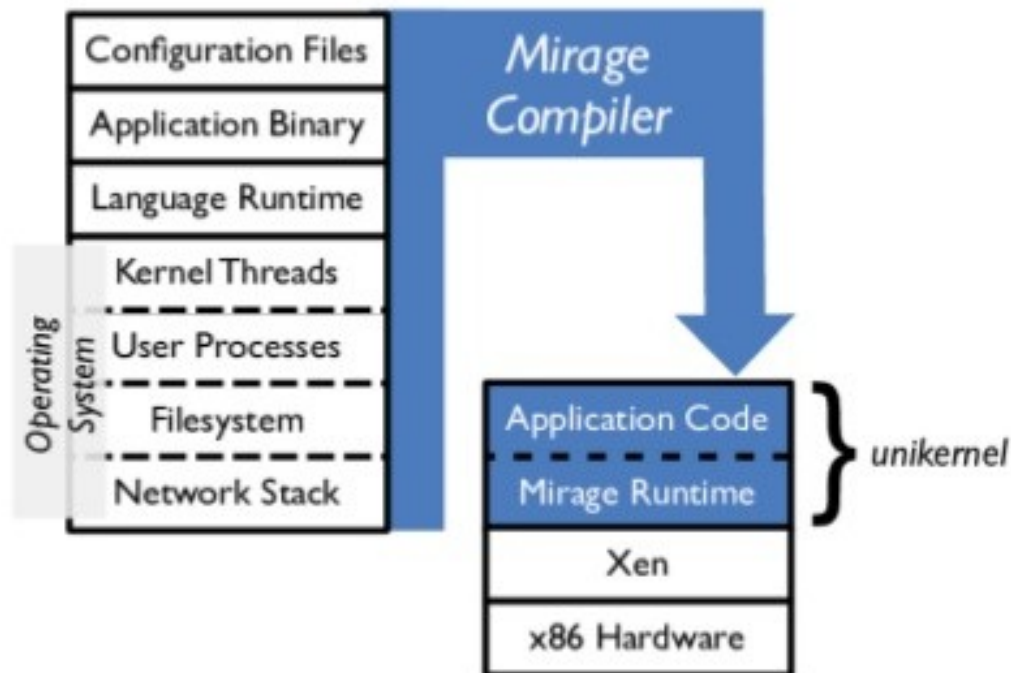
Swap system libraries to target different platforms:

test unikernel using Mirage system libraries.



Unikernel Approach: Mirage OS

Swap system libraries to target different platforms:
deploy by specialising unikernel to Xen.



Unikernel Concepts

- Use just enough to do the job
 - No need for multiple users; one VM per user
 - No need for a general purpose operating system
 - No need for utilities
 - No need for a full set of operating system functions
- Lean and mean
 - Minimal waste
 - Tiny size

What Do the Results Look Like?

- Mirage OS examples:
 - DNS Server: 449 KB
 - Web Server: 674 KB
 - OpenFlow Learning Switch: 393 KB
- LING metrics:
 - Boot time to shell in under 100ms
 - Erlangonxen.org memory usage: 8.7 MB
- ClickOS:
 - Network devices processing >5 million pkt/sec
 - 6 MB memory with 30 ms boot time

What's Out There Right Now?

- Mirage OS, from the Xen Project Incubator
- HaLVM, from Galois
- LING, from Erlang-on-Xen
- ClickOS, from NEC Labs
- OSv, from Cloudeus Systems
- And that's just the beginning...

Mirage OS

- From the Xen Project Incubator
- Language support: Ocaml
- Hypervisor support: Xen Project
- V2.0 released this year
- General purpose devices
- <http://www.openmirage.org/>

HaLVM

- Galois, Inc.
- Language support: Haskell
- Hypervisor support: Xen Project
- Originally designed to prototype operating system components
- Now suitable for creating network devices
- <https://galois.com/project/halvm/>

LING

- Erlang-on-Xen project
- Language support: Erlang
- Hypervisor support: Xen Project
- Use cases include Zero-Footprint Cloud
- <http://erlangonxen.org/>

ClickOS

- NEC Labs
- Language support: C, C++, Python
- Hypervisor support: Xen Project
- V0.2 released this year
- Suited for Network Function Virtualization (NFV) devices
- <https://cnp.neclab.eu/clickos/>

OSv

- Cloudeus Systems
- Language support: C, C++, Java
- Hypervisor support: Xen Project, KVM, VMware
- Slightly different from “standard” unikernels
 - Kind of “fat”
 - Full Java JVM stack, minus multi-processes (threads yes, forks no)
 - Can run almost any JAR file
- NFV optimized
- <http://osv.io/>

Are Unikernels a Panacea?

- Nope!
 - There will always be large databases and beefy apps which won't fit in this mold
 - The truth is that different problems are likely to require different optimal solutions for the foreseeable future
 - It is likely that the solution spectrum of the next few years will include a blend of unikernels, containers, and standard virtualization
 - But the arrival of unikernels means that the bar to efficiency has been raised to new heights

Open Source Leading the Way

- This is an example of how Open Source is working to expand horizons of the cloud
 - The closed source cloud just isn't the way to go
 - The real innovation in cloud is in Open Source
 - Xen Project is at the forefront of new cloud thinking, incubating and facilitating new technologies, including unikernels
 - Friends don't let friends go closed source in the cloud!

Questions?

Russell.Pavlicek@XenProject.org
Twitter: @RCPavlicek

Thanks to the Mirage OS team for the use of their images

This presentation will be available in the Presentations Section of
XenProject.org