unikernel

The next big little thing ?

Docker Grenoble Meetup, 19 Jan 2017

Mike Bright, @mjbright

Viktor Farcic, senior consultant at CloudBees

… One of the most exciting areas that will become prominent in 2017 will be unikernels.

While the majority of the industry is still trying to wrap their heads around containers, we will start seeing unikernels taking over the stage.

They will, in a way, unify functionalities provided by VMs and containers.

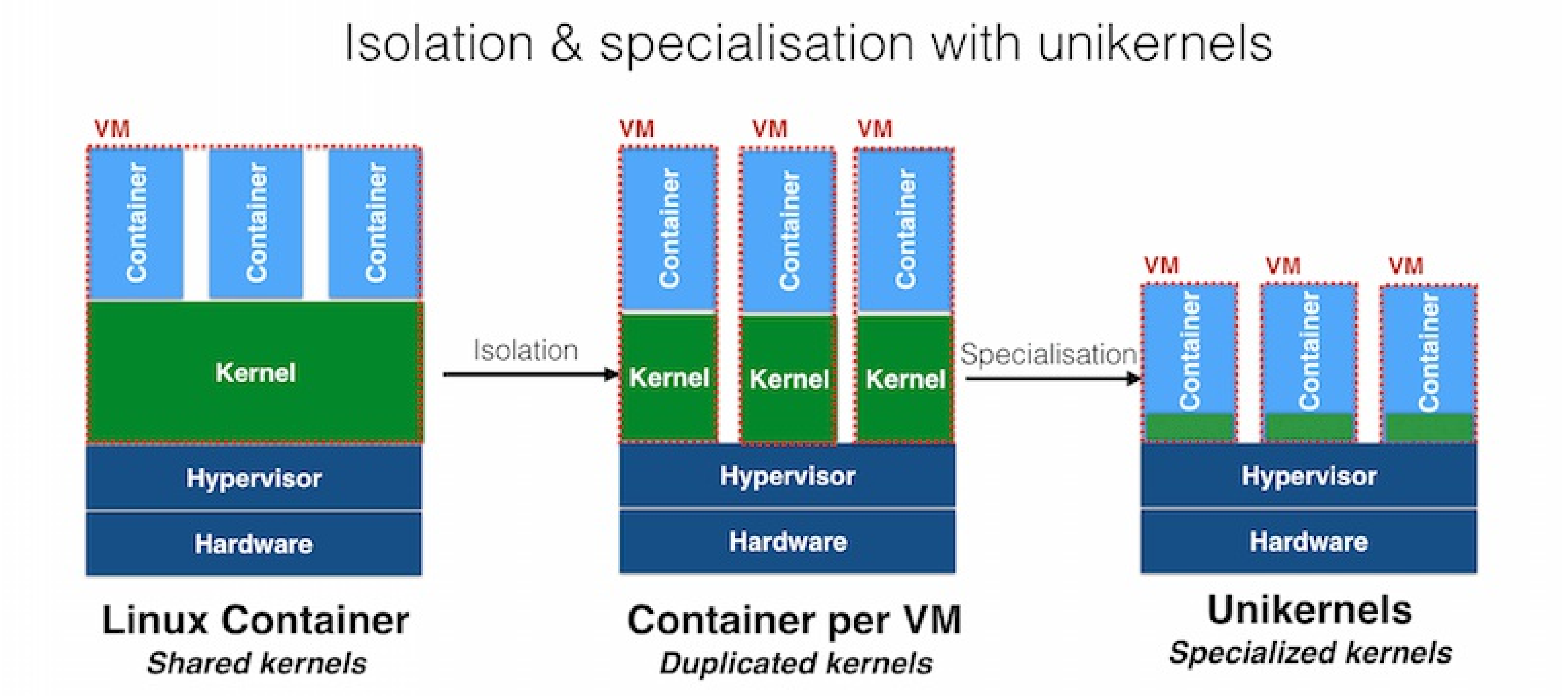
http://sdtimes.com/whats-horizon-2017/

Unikernels - Overview

- What are Unikernels?
- How did we get here?
- Unikernel implementations
  - Clean Slate
  - POSIX compatible
  - Tools
- Future
- Resources

What are Unikernels? "Library OS"

Unikernels are applications images built with only the Operating System components they actually require, e.g. TCP Stack, Disk access.

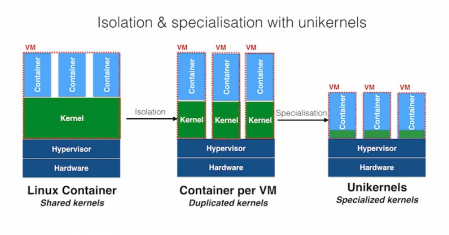


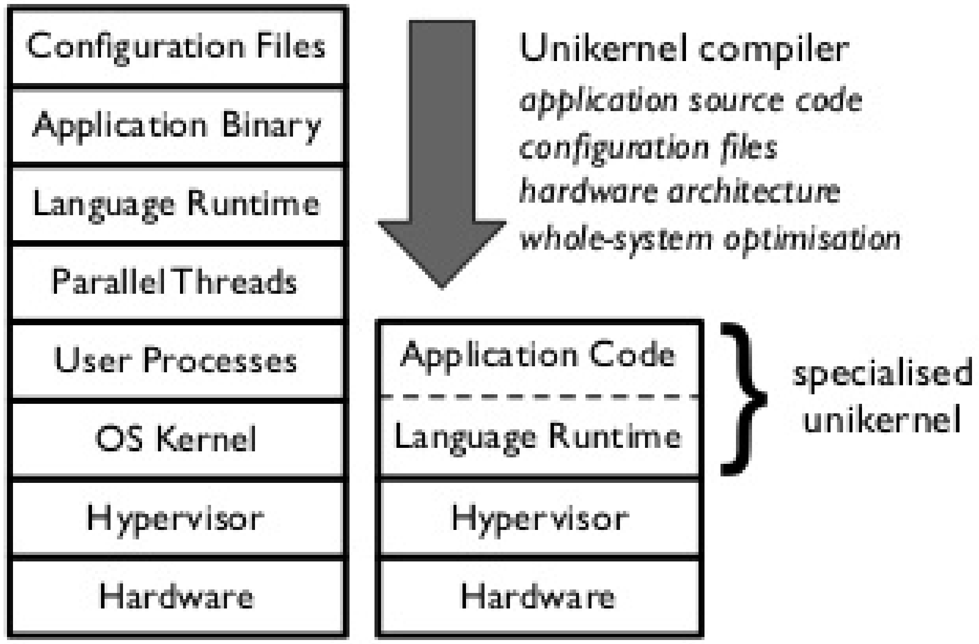
Single process applications (no threads, forking or multi-user) with very small size -> high performance, fast boot and small attack surface (secure).

@mjbright

What are Unikernels - how did we get here?

"Library OSes"





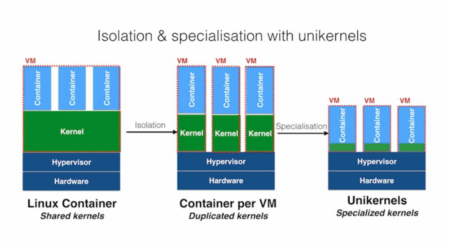
A Unikernel is built by the compiler linking only the OS components needed by the application.

The OS becomes a "Library OS"

Unlike "normal" applications which sit atop a generic monolithic Linux kernel (or even µ-kernel) which has many unneeded features, e.g. floppy driver.

What are Unikernels - how did we get here?

"Library OSes"

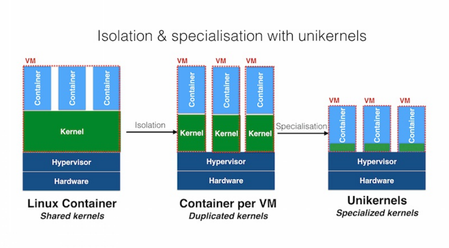


Unneeded features consume resources unnecessarily.

Unneeded legacy features represent a security risk - especially in the cloud.

What are Unikernels - how did we get here?

"Library OSes"



Unneeded features consume resources unnecessarily.

Unneeded legacy features represent a security risk - especially in the cloud.

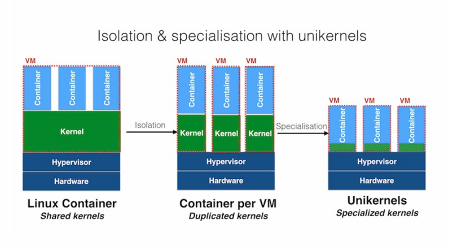
At October's "Docker Distributed Summit", Docker even talked of minimizing the Hypervisor also.

▶ "Unikernels The rise of the library hypervisor in MirageOS"

(ACM: Unikernels: Rise of the Virtual Library Operating System , Jan 2014)

What are Unikernels - how did we get here?

"Library OSes"



Unneeded features consume resources unnecessarily.

Unneeded legacy features represent a security risk - especially in the cloud.

At October's "Docker Distributed Summit", Docker even talked of minimizing the Hypervisor also.

"Unikernels The rise of the library hypervisor in MirageOS"

(ACM: Unikernels: Rise of the Virtual Library Operating System , Jan 2014)

These minimal systems can take ~ 200msec to boot.

This opens up the possibility of services being spin up on demand (MirageOS jitsu).

Where will Unikernels be used?

"Library OSes"

Application domains

- Cloud, e.g. serverless
- IoT (Embedded)
- HPC
- NFV (▶ Unikernels meet NFV Ericsson Research)

Where will Unikernels be used?

"Library OSes"

Application domains

- Cloud, e.g. serverless
- IoT (Embedded)
- HPC
- NFV (▶ Unikernels meet NFV Ericsson Research)

From the NFV Containers White Paper (2.3. Unikernels):

```
Unikernels are essentially single-application virtual machines
based on minimalistic OSes. Such minimalistic OSes have minimum
overhead and are typically single-address space (so no user/kernel
space divide and no expensive system calls) and have a
co-operative scheduler (so reducing context switch costs).

Examples of such minimalistic OSes are MiniOS [MINIOS] which
runs on Xen and OSv [OSV] which runs on KVM, Xen and VMWare.
```

https://datatracker.ietf.org/doc/draft-natarajan-nfvrg-containers-for-nfv/?include_text=1

Unikernel implementations

Unikernel Families

Many Unikernel implementations exist, there are two main classes of Unikernels

Unikernel implementations

Unikernel Families

Many Unikernel implementations exist, there are two main classes of Unikernels

Some take a clean-slate approach and emphasize safety and security. These tend to use the same language for the application and the Library OS components.

- MirageOS (Ocaml)
- HalVM (Haskell)

Unikernel implementations

Unikernel Families

Many Unikernel implementations exist, there are two main classes of Unikernels

Some take a clean-slate approach and emphasize safety and security. These tend to use the same language for the application and the Library OS components.

- MirageOS (Ocaml)
- HalVM (Haskell)

Others favour backward compatibility of existing applications based on POSIX-compatibility.

Many applications have been ported

- OSv (Tomcat, Jetty, Cassandra, OpenJDK, ...)
- Rumprun (MySQL, PHP, Nginx)

Unikernel Implementations

Technology	Description
ClickOS cnp.neclab.eu	For embedded network h/w. ~5MB images, boots <20ms, 45 μs delay, 100 VMs => 10Gbps
Clive lsub.org	Written in Go. For distributed and cloud.
Drawbridge MS	Research prototype. Picoprocess/container with minimal kernel API surface, and Windows library OS.
Graphene graphene	Securing "multi-process" legacy apps - adds IPC.
HaLVM galois.com	Port of GHC (Glasgow Haskell Compiler) suite. Write apps in Haskell to run on Xen.
IncludeOS includeos.org	Research project for C++ code on virtual hardware.
LING erlangonxen.org	Erlang/OTP runs on Xen.
MirageOS mirage.io	Clean-slate library OS for secure, high-perf network apps. More than 100 MirageOS libraries plus OCaml ecosystem.
OSv osv.io Cloudius	Run Linux binaries (w. limitations), supports C/C++, JVM, Ruby, Node.js
Rumprun rumpkernel.org	FreeBSD - Runs POSIX s/w on BM or VM (Xen).

Unikernel implementations - MirageOS/Ocaml

Clean-Slate

MIRAGE OS

https://mirage.io/

OCaml-Based

OCaml

MirageOS "Library OS" components are written in Ocaml .

ML-derived languages are best known for their static type systems and type-inferring compilers.

OCaml unifies functional, imperative, and object-oriented programming under an ML-like type system.

OCaml has extensive libraries available

(Unison utility)

Unikernel implementations - MirageOS-2

Clean-Slate

https://mirage.io/

OCaml-Based

MirageOS Unikernels are based on the Mirage-OS Unikernel base (OS library).

The mirage tool is used to build Unikernels for various backends:

- Xen Hypervisor (PV)
- Unix (Linux or OS/X binaries)
- Browser (via Ocaml->JS compiler !!)
- Even an experimental BM backend for Raspberry Pi

Unikernel implementations - MirageOS-2

Clean-Slate

MIRAGE OS

https://mirage.io/

OCaml-Based

OCaml

MirageOS Unikernels are based on the Mirage-OS Unikernel base (OS library).

The mirage tool is used to build Unikernels for various backends:

- Xen Hypervisor (PV)
- Unix (Linux or OS/X binaries)
- Browser (via Ocaml->JS compiler !!)
- Even an experimental BM backend for Raspberry Pi

Building applications for unix or xen

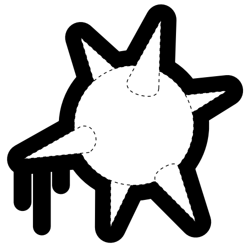
```
mirage configure -t unix
make
./mir-console

mirage configure -t xen
make
****xen create ./mir-console.xen
```

Unikernel implementations - MirageOS - Use Cases

Clean-Slate

MIRAGE OS

https://mirage.io/



- BNC Pinata: http://ownme.ipredator.se/
- Networking applications
  - e.g. CyberChaff "false network hosts"
- PayGarden, Sean Grove

  ▶ "Baby steps to unikernels in production"

  - Too painful to create/configure AMI images on AWS
  - Solo5 allows to create KVM images deployable on GCE

Unikernel implementations - HalVM

Clean-Slate

HalVM - The Haskell Lightweight Virtual Machine: GHC running on Xen

- https://github.com/GaloisInc/HaLVM
- HalVM3 is reconsidering it's Unikernel base http://uhsure.com/halvm3.html
    - Use rumpkernel (NetBSD base)
    - Shift to Solo5?

Unikernel implementations - OSv

POSIX-based

http://osv.io
http://blog.osv.io

OSv - Capable of running POSIX binaries

- can run JVM
- Cassandra: https://www.penninkhof.com/2015/05/minimalist-cassandra-vm-using-osv/

- Used in Mikelangelo (EU Project) The MIKELANGELO project aims to bring High Performance Computing (HPC) to the cloud. HPC traditionally involves bleeding edge technologies, including lots of CPU cores, Infiniband interconnects between nodes, MPI libraries for message passing, and, surprise—NFS, a very old timer of the UNIX universe.

- Building OSv Images Using Docker: http://blog.osv.io/blog/blog/2015/04/27/docker/

- SDI: ODL + OSv: http://blog.osv.io/blog/blog/2015/03/31/sdi/

Unikernel Tools

Tools

- Unik : tool for compiling apps to unikernels (various technologies)
- Solo5 : An alternative unikernel-base for MirageOS
  - Provides qemu/KVM support for MirageOS
- ukvm: An alternative VM Monitor
  - a "library hypervisor"
- capstan : OSv build tool

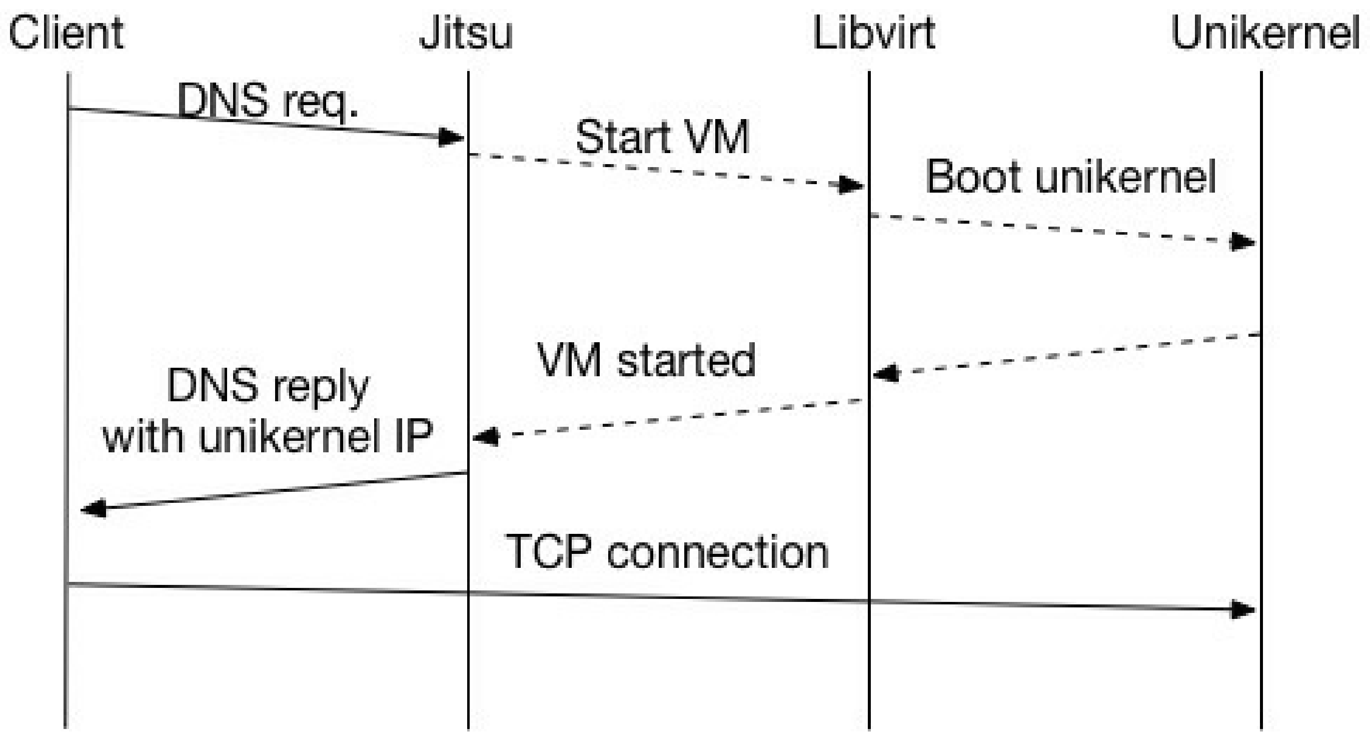
Unikernel Tools

Clean-Slate

MIRAGE OS

MirageOS jitsu : "Just-In-Time Summoning of Unikernels"

A DNS server that starts unikernels on demand.



Tested with MirageOS and Rumprun unikernels.

https://github.com/mirage/jitsu

@mjbright

Unikernels and Containers

Unikernels or Containers?

So what has this got to do with Containers?

Why did Docker buy Unikernel Systems (Jan 2016)?

Info.Q / Amir, Aug 2016

Unikernels and Containers

Unikernels or Containers?

So what has this got to do with Containers?

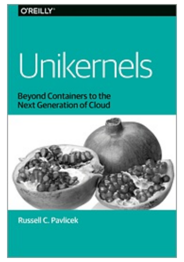
Why did Docker buy Unikernel Systems (Jan 2016)?

Info.Q / Amir, Aug 2016

- Unikernel Systems are involved in MirageOS/Xen
- Use of Unikernels in Docker for Mac
    - VPNKit, DataKit
- To provide build/run/ship tools for Unikernels?
- To secure Container deployments
    - Running Unikernels in containers????
    - Secure front-ends in hybrid solutions made of unikernels and containers
        - e.g. for OCaml MediaWiki (http2https, tls, ...)

Demo

Resources

Scoop.it!	Scoop.it Unikernels	www.scoop.it/t/unikernels
WIKIPEDIA The Free Encyclopedia	Wikipedia	en.wikipedia.org/wiki/Unikernel
unikernel	unikernels.org	unikernels.org
MIRAGE OS	mirageos.io	mirageos.io mirage.io/docs/papers
	OReilly "Unikernels"	Free download
	@unikernel	@unikernel
	github.com/ocamllabs	ocamllabs
	github.com/mirage	MirageOS

Thank you

Q&A