unikernel

Technologies

3 juillet 2017, RMLL St-Etienne, Michael Bright

@mjbright

Agenda

- What are Unikernels ?
  - What they are not.
- Why Unikernels ?
  - Advantages / Characteristics
  - Application domains
- Implementations & Tools
- Demos
- Usage: Baremetal anyone ?
- Where’s it all heading ?

@mjbright

What's it all about ?

@mjbright

What are Unikernels?

"Unikernels are specialized, single-address-space machine images constructed by using library operating systems"

"What are Unikernels", unikernel.org

What are Unikernels?

“Unikernels are specialized, single-address-space machine images constructed by using library operating systems”

“What are Unikernels”, unikernel.org

“VMs aren't heavy, OSes are"

Alfred Bratterud, #includeOS

What are Unikernels? - They are "Library OS"

Specialized applications built with only the "OS" components they need.

A Unikernel is an image able to run directly as a VM

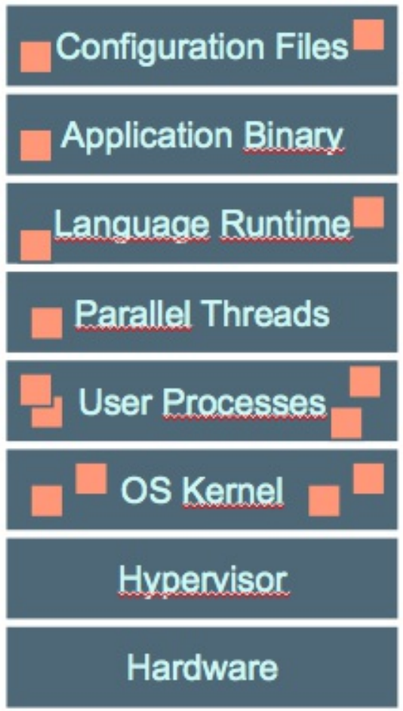
(on bare metal?)

"OS" components such as Network stack, File-system, Device drivers are optional

typically, there is no filesystem.

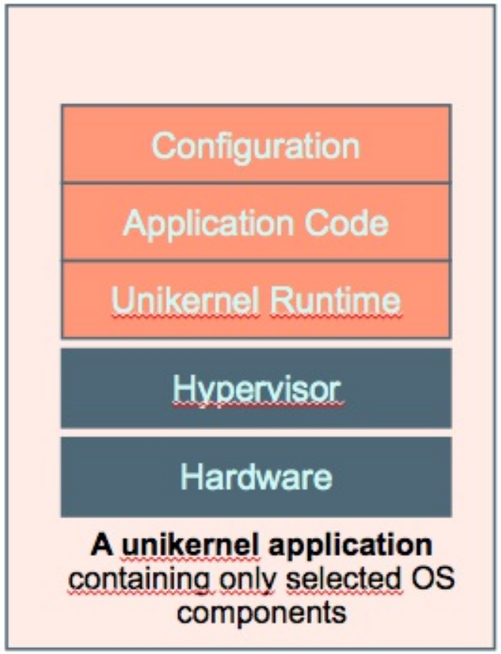
So configuration is stored in the unikernel application binary

@unjuggler



Typical application

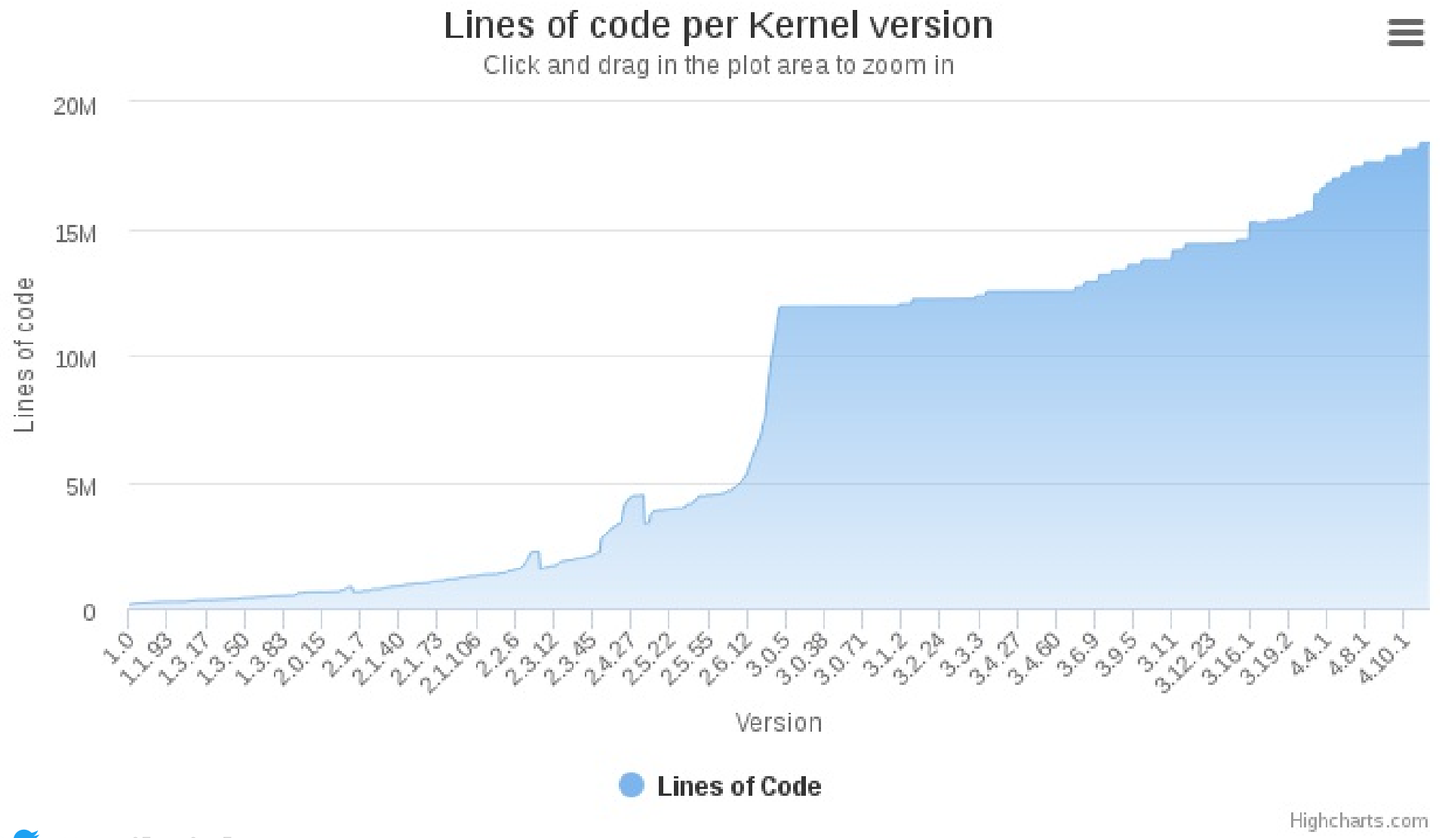
running above an OS



A unikernel application containing only selected OS components

Unikernels: What they are not ... General Purpose

OS kernels with unneeded features e.g. floppy drivers, designed to run any software on any hardware are huge - lines of code



@mjbright

Unikernels are not "top-down" minified versions of General Purpose OSes ...

Unikernels: What they are not ... minified OS

Container hosts

Minimal Linux distributions have been created with similar goals to Unikernels, aimed to be minimal host OS for container engines, e.g.

- CoreOS Linux
- Project Atomic
- RancherOS

They aim to be

- Secure
  - Less features/lines of code : reduced attack surface
  - Atomic updates of system (not quite immutable)
- Fast to boot : Small binary size
- Specialized to run containers

But these are still reduced versions of general purpose OSes and so have many unnecessary features.

@mjbright

Unikernels: What they are not ... minified OS

LINUXKIT

In April 2017 Docker open sourced LinuxKit a way of building minimal Linux distributions for hosting containers.

Unikernels: What they are not ... minified OS

LINUXKIT

In April 2017 Docker open sourced LinuxKit a way of building minimal Linux distributions for hosting containers.

LinuxKit is also a specialized Container Host with

- declarative specification of the system components to include
- services and applications encapsulated in containers
- MirageSDK ... looks interesting ...

Unikernels: What they are not ... minified OS

LINUXKIT

In April 2017 Docker open sourced LinuxKit a way of building minimal Linux distributions for hosting containers.

LinuxKit is also a specialized Container Host with

- declarative specification of the system components to include
- services and applications encapsulated in containers
- MirageSDK ... looks interesting ...

LinuxKit is still based on a General Purpose Linux Kernel but allows for much customization of the base system.

It's just one step closer to Unikernels ...

Unikernels: What they are not ... minified OS

LINUXKIT

In April 2017 Docker open sourced LinuxKit a way of building minimal Linux distributions for hosting containers.

LinuxKit is also a specialized Container Host with

- declarative specification of the system components to include
- services and applications encapsulated in containers
- MirageSDK ... looks interesting ...

LinuxKit is still based on a General Purpose Linux Kernel but allows for much customization of the base system.

It's just one step closer to Unikernels ...

... who knows what Docker will do next ?? ...

Unikernels: What they are not ... summary

They are not minified general purpose OS

- Not μ-kernels
- Not minified Linux kernels or Container OS

Unikernels: What they are not ... summary

They are not minified general purpose OS

- Not μ-kernels
- Not minified Linux kernels or Container OS

They are not real-time Oses

- But they are very fast

Unikernels: What they are not ... summary

They are not minified general purpose OS

- Not μ-kernels
- Not minified Linux kernels or Container OS

They are not real-time Oses

- But they are very fast

They are not

- Multi-kernels (though HermitCore is !)
- Multi-process (though Graphene is !)

Unikernels: What they are not ... summary

They are not minified general purpose OS

- Not μ-kernels
- Not minified Linux kernels or Container OS

They are not real-time Oses

- But they are very fast

They are not

- Multi-kernels (though HermitCore is !)
- Multi-process (though Graphene is !)

They are not all the same, but work on similar principles ...

Unikernels: What they are not ... summary

They are not minified general purpose OS

- Not μ-kernels
- Not minified Linux kernels or Container OS

They are not real-time Oses

- But they are very fast

They are not

- Multi-kernels (though HermitCore is !)
- Multi-process (though Graphene is !)

They are not all the same, but work on similar principles ...

Building a specialized application with only the "OS" components needed
==> a "bottom-up" approach

Unikernels: Are ...

Very small compared to an application + OS

- use few resources
- immutable, suitable for micro-services
- No legacy drivers
- No unneeded shell - did I mention this?

Unikernels: Are ...

Very small compared to an application + OS

- use few resources
- immutable, suitable for micro-services
- No legacy drivers
- No unneeded shell - did I mention this?

Have no separate kernel space

- No need to copy between kernel and user space

Unikernels: Are ...

Very small compared to an application + OS

- use few resources
- immutable, suitable for micro-services
- No legacy drivers
- No unneeded shell - did I mention this?

Have no separate kernel space

- No need to copy between kernel and user space

More secure

- small attack surface
- If compromised, the attacker can’t do much - no shell, users, processes ...

Unikernels: Are ...

Very small compared to an application + OS

- use few resources
- immutable, suitable for micro-services
- No legacy drivers
- No unneeded shell - did I mention this?

Have no separate kernel space

- No need to copy between kernel and user space

More secure

- small attack surface
- If compromised, the attacker can't do much - no shell, users, processes ...

Fast to boot

- Possibility of on demand services

Unikernels: Are ...

Very small compared to an application + OS

- use few resources
- immutable, suitable for micro-services
- No legacy drivers
- No unneeded shell - did I mention this?

Have no separate kernel space

- No need to copy between kernel and user space

More secure

- small attack surface
- If compromised, the attacker can't do much - no shell, users, processes ...

Fast to boot

- Possibility of on demand services

More difficult to develop

- libraries, languages, debugging limitations

Unikernels: Application Domains

Cloud Computing and NFV

- Fast to boot: On demand services
- Secure immutable images

Unikernels: Application Domains

Cloud Computing and NFV

- Fast to boot: On demand services
- Secure immutable images

IoT / Embedded

- Small images for OTA updates
- Secure immutable images

Unikernels: Application Domains

Cloud Computing and NFV

- Fast to boot: On demand services
- Secure immutable images

IoT / Embedded

- Small images for OTA updates
- Secure immutable images

HPC

- Secure in the cloud
- Very efficient (no context switches, just 1 process)

Unikernel implementations

Unikernel Implementations: 2 families

Clean-Slate	Legacy
- A minimalist approach	- POSIX compatibility
- Re-implement all OS functions	- Re-use existing libraries
- Typically uses type safe language	- Possible binary compatibility
- Very small code size, resources	- Small to large code size/resources
- Harder to develop apps	- Easier to develop apps

Unikernel Implementations: 2 families

Clean-Slate	Legacy
- A minimalist approach	- POSIX compatibility
- Re-implement all OS functions	- Re-use existing libraries
- Typically uses type safe language	- Possible binary compatibility
- Very small code size, resources	- Small to large code size/resources
- Harder to develop apps	- Easier to develop apps

This means that clean-slate Unikernels tend to be implemented solely in one high-level language (and possibly derived languages)

Unikernel Implementations: 2 families

Clean-Slate	Legacy
- A minimalist approach	- POSIX compatibility
- Re-implement all OS functions	- Re-use existing libraries
- Typically uses type safe language	- Possible binary compatibility
- Very small code size, resources	- Small to large code size/resources
- Harder to develop apps	- Easier to develop apps

This means that clean-slate Unikernels tend to be implemented solely in one high-level language (and possibly derived languages)

We can see that Legacy Unikernels trade off some principles for ease of use ...

Unikernel Implementations:

Clean-Slate	Legacy
MirageOS (Ocaml)	OSv
HalVM (Haskell)	Rumprun (+LKL)
LING (Erlang)	Runtime.js
IncludeOS (C/C++)	HermitCore
	Graphene
	ClickOS
	Vorteil
	Clive
	Magnios
	Ultibo
	Drawbridge
	… others ? …

There's some collaboration going on across projects especially to use some common underlying layers such as Minio, Solo5/ukvm.

Unikernel Implementations: MirageOS - Xen project

MIRAGE OS

mirage.io

Clean Slate

Open Source

Backing (Docker/Xen)

OCaml-Based

MirageOS "Library OS" components and apps are written in Ocaml , a type-safe functional (& OO) language with extensive libraries.

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

- Xen Hypervisor (PV)
- Unix (Linux or OS/X binaries)
- MirageOS 3 (/Solo5) supports kvm (/ukvm) and xhyve

Building applications for unix or xen

```
mirage configure -t [unix|xen|ukvm]
make depend
make
./mir-console
```

Use cases: BNC Pinata  , E/// Research NFV, PayGarden

@mjbright

Unikernel Implementations: HalVM

halvm.org

Clean Slate

Open Source

Backing (Galois)

A port of GHC (the Glasgow Haskell Compiler) to run as a Unikernel

Runs on Xen

Considering port to Solo5 for HalVM v3.

[2012] HalVM is a "nifty platform" for

- developing simple cloud services
- creating critical, isolated services

Aimed at highly secure network appliances such as CyberChaff

 @mjbright

Unikernel Implementations: IncludeOS

includeOS

includeos.org

Clean Slate

Open Source

Backing (IncludeOS)

C/C++

FAQ

Written in C++.

Create Unikernel from an application by including `#include <os>`

Runs on hypervisors (KVM, VMWare) maybe baremetal ...

Single-threaded, single-process, single-memory space

Delegates to route messages between TCP/IP stack components.

No blocking POSIX calls implemented yet, only async i/o.

Recent developments:

- Working with Mender (mender.io) for OTA updates
- 64-bit
- ARM?
- Solo5 (ukvm)

@mjbright

Unikernel Implementations: OSv

osv.io

Legacy

Open Source

Backing (Cloudius)

Written in C++ but with "POSIX" compatibility

- includes threads, tcp/ip, ZFS filesystem
- support for other languages and memory-managed platforms (JVM, Go, Lua)
- used in Mikelangelo EU Project (OpenStack+Unikernels)

Runs on KVM, Xen, VBox, VMWare

The OSv Manifesto

- Run existing Linux apps, run them faster
- Boot time ~ Exec time
- Leverage memory-managed platforms
- Stay open

Single process, address space

TCP/IP stack components (C++ classes) communicate via net channels

Possibility for MMU to handle garbage collection

Unikernel Implementations: Rumprun

rumpkernel.org

Legacy

Open Source

Backing (NetBSD)

A refactoring of the NetBSD kernel allowing to select OS modules as needed.

Unikernel base in C/C++, supports many languages

- C/C++, Lua, PHP, Python, Ruby, Node.js, Erlang, Go

Workflow is

- cross-compile against NetBSD libc (modified)
- bake in the hypervisor choice (not KVM ...)
- launch VM

Baremetal "Hypercall" implementation.

Many available packages: apache2, nginx, haproxy, redis, mysql, sqlite, leveldb, tor, mpg123

NOTE: LKL (Linux Kernel Libraries) an experimental Linux version since 2015

Unikernel Implementations: Runtime.js

runtimejs.org

Legacy

Open Source

Implementation of v8 Javascript engine as a Unikernel

Supports Node.js on KVM Hypervisor

Ongoing discussions about supporting WebAssembly ..

Unikernel Implementations: HermitCore

hermitcore.org

Legacy

Open Source

Experimental unikernel from University of Aachen, initial performance results are promising.

Supports SMP in multi-kernel mode.

Modes:

- "classical unikernel" - runs on a VM
- multikernel on VM: proxy "Linux" kernel on one core, separate applications on other cores
- multikernel on BM: proxy "Linux" kernel on one core, separate applications on other cores

Uses Intel OpenMP runtime.

Languages:

- C++, Fortran, Go (all via gcc)

Unikernel Tools

Open Source tools help to advance the various projects.

Unik: Unikernel Compiler

Cloud Foundry project (Dell-EMC) compiles several Unikernel Technologies

- Supports: RumpRun, OSv, IncludeOS, MirageOS

'VboxUnikInstanceListener' VM handles requests from the 'unik' cli.

Solo5/ukvm

A common Unikernel (Solo5) base and (ukvm) library hypervisor developed by IBM.

Integrated into MirageOS v3 to extend to KVM support. Other projects (HalVM, IncludeOS) are also considering this approach. Ongoing port to ARM64.

@mjbright 20

Deferpanic

Demo

- MirageOS
  - compilation for unix
  - compilation/run for Solo5/ukvm
- Runtime.js
- Deferpanic.net

What's coming?

Docker bought Unikernel Systems (main MirageOS developers) in Jan 2016

- Unikernel technology used in Docker for Mac, Docker for Windows

MirageOS v3 released in March 2017

- improves MirageOS implementation (less code, more func)
- New Solo5 backend: kvm via Solo5/ukvm

Unikernels are becoming easier to use

- Adoption of existing backends: Minios/Xen, Solo5/ukvm
- LinuxKit/MirageSDK synergies with MirageOS?
- Docker facilitates Build Ship and Run for Unikernel technologies
- Unik project facilitates use of different Unikernel technologies
- Cloud Foundry and Kubernetes look to deploy Unikernels
- Solo.io "Squash" project producing debugger for µ-services and Unikernels

Many Unikernel projects are advancing quickly … and specialized deployment trials ongoing

@mjbright

Unikernels: Usage? Baremetal?

Specific applications (network appliances - Hybrid solutions)

Well-suited for very specific applications such as target networking components

```
- DNS, DHCP, NAT, Firewall, TLS, Chaff
```

Can be used as standalone appliances or as secure network front-end.

But what about Baremetal ?

Some Unikernels target baremetal, but not appropriate for all use cases

- requires maintaining h/w specific device drivers
- may not support more than 1 core !

You won't want to dedicate your latest Proliant server to one Unikernel (flea on an elephant's back), but rather to a Hypervisor running Unikernels

May be appropriate for the smallest IoT devices (webcam, sensor)

Unikernels: Conclusions ...

A very active research area

- many active projects, several with commercial backers
- mostly Open Source
- healthy collaboration - common tooling possible

Unikernels: Conclusions ...

A very active research area

- many active projects, several with commercial backers
- mostly Open Source
- healthy collaboration - common tooling possible

Some projects adopt a "Clean-Slate" approach building up capabilities.

- impose a particular language
- smallest, most secure Unikernels
- potentially harder to develop

Other projects trade off some of the Unikernel advantages for "ease of use".

Unikernels: Conclusions ...

A very active research area

- many active projects, several with commercial backers
- mostly Open Source
- healthy collaboration - common tooling possible

Some projects adopt a "Clean-Slate" approach building up capabilities.

- impose a particular language
- smallest, most secure Unikernels
- potentially harder to develop

Other projects trade off some of the Unikernel advantages for "ease of use".

We will start to hear of deployments for specific use cases

Unlikely to become a mainstream approach

- competition from VMs, containers, serverless
- unless someone surprises us ...

Q&A

@mjbright

Resources

@mjbright

Resources - General

	URL
.	
Unikernel.org	site
Wikipedia	Wiki
.	
Scoop.It	Unikernels
Playlist	YouTube Unikernels

@mjbright

Resources - Unikernel Implementations

Technology	Backers	URL
.		
MirageOS	Xen	mirage.io
HalVM	Galois	galois.com/project/halvm
LING		erlangonxen.org
.		
IncludeOS	IncludeOS	includeos.org
Rumprun	NetBSD	rumpkernel.org
OSv	Cloudius	osv.io
HermitCore	Univ. Aachen	hermitcore.org
.		
Unik	CloudFoundry	github.com/cf-unik/unik
Solo5	IBM	github.com/Solo5/solo5
Ukvm	IBM	github.com/Solo5/solo5/tree/master/ukvm

Resources - Unikernel Implementations (2)

Technology	Backers	URL
.		
Ultibo (Raspi) Clive (Go) Magnios ClickOS	NEC	
.		
Drawbridge	Microsoft	project/drawbridge
.		
DeferPanic	DeferPanic	deferpanic.net

@mjbright