

Unikernels and another way of secure cloud computing

Motiejus Jakštys
motiejus@amazon.com



2015-11-19

Disclaimer

This presentation is intended to give a high level overview of the subject matter and is intended for discussion purposes. This presentation is not intended to provide an exhaustive analysis of the subject matter and may differ depending on individual use cases.

Table of Contents

Introduction

Roles of an IT organization

How to deploy?

Meat of Unikernels

Look ma, no OS

What's in a unikernel?

Performance benefits

Security benefits

Unikernel examples

What's next?

Unmodified applications first

Local demo

In Amazon EC2

Summary

Typical IT organization

| Role | What system? |
|-----------------------------|--------------|
| Developer | |
| Operator Fred Flintstone | |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app. |
| Operator Fred Flintstone | |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|--|
| Developer | <ul style="list-style-type: none"><li data-bbox="458 215 916 251">▶ Cares only about the app.<li data-bbox="458 282 1071 319">▶ Familiar environment (POSIX API). |
| Operator Fred Flintstone | |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work. |
| Operator Fred Flintstone | |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work.▶ Hopefully prod matches dev. |
| Operator Fred Flintstone | |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work.▶ Hopefully prod matches dev. |
| Operator Fred Flintstone | <ul style="list-style-type: none">▶ Simple systems. |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work.▶ Hopefully prod matches dev. |
| Operator Fred Flintstone | <ul style="list-style-type: none">▶ Simple systems.▶ Predictable systems. |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work.▶ Hopefully prod matches dev. |
| Operator Fred Flintstone | <ul style="list-style-type: none">▶ Simple systems.▶ Predictable systems.▶ Speedy deployments and rollbacks. |
| Security certifier Mr Burns | |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work.▶ Hopefully prod matches dev. |
| Operator Fred Flintstone | <ul style="list-style-type: none">▶ Simple systems.▶ Predictable systems.▶ Speedy deployments and rollbacks. |
| Security certifier Mr Burns | <ul style="list-style-type: none">▶ Auditable components. |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work.▶ Hopefully prod matches dev. |
| Operator Fred Flintstone | <ul style="list-style-type: none">▶ Simple systems.▶ Predictable systems.▶ Speedy deployments and rollbacks. |
| Security certifier Mr Burns | <ul style="list-style-type: none">▶ Auditable components.▶ Least possible dependencies. |

Typical IT organization

| Role | What system? |
|-----------------------------|---|
| Developer | <ul style="list-style-type: none">▶ Cares only about the app.▶ Familiar environment (POSIX API).▶ Here's a JAR, make this thing work.▶ Hopefully prod matches dev. |
| Operator Fred Flintstone | <ul style="list-style-type: none">▶ Simple systems.▶ Predictable systems.▶ Speedy deployments and rollbacks. |
| Security certifier Mr Burns | <ul style="list-style-type: none">▶ Auditable components.▶ Least possible dependencies.▶ Trades convenience for security. |

How to deploy?

Deployment

Developer

Operations

Security

How to deploy?

| Deployment | Developer | Operations | Security |
|--|-----------|------------|----------|
| Mutable, full OS. <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | | | |

How to deploy?

| Deployment | Developer | Operations | Security |
|--|-----------|------------|----------|
| Mutable, full OS. <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | | |

How to deploy?

| Deployment | Developer | Operations | Security |
|--|-----------|------------|----------|
| Mutable, full OS. <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | |

How to deploy?

| Deployment | Developer | Operations | Security |
|--|-----------|------------|--------------|
| Mutable, full OS. <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | | | |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | Happy | | |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | Happy | Unhappy | |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | Happy | Unhappy | Doesn't care |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | Happy | Unhappy | Doesn't care |
| <p>Immutable, no OS.</p> <ul style="list-style-type: none">▶ Unikernels | | | |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | Happy | Unhappy | Doesn't care |
| <p>Immutable, no OS.</p> <ul style="list-style-type: none">▶ Unikernels | Happy | | |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | Happy | Unhappy | Doesn't care |
| <p>Immutable, no OS.</p> <ul style="list-style-type: none">▶ Unikernels | Happy | Happy | |

How to deploy?

| Deployment | Developer | Operations | Security |
|---|-----------|------------|--------------|
| <p>Mutable, full OS.</p> <ul style="list-style-type: none">▶ Ansible/Chef/Puppet▶ Adhoc shell scripts▶ AWS Opsworks | Unhappy | Unhappy | Doesn't care |
| <p>Immutable, full OS.</p> <ul style="list-style-type: none">▶ Docker▶ AMI per release | Happy | Unhappy | Doesn't care |
| <p>Immutable, no OS.</p> <ul style="list-style-type: none">▶ Unikernels | Happy | Happy | Happier |

Table of Contents

Introduction

Roles of an IT organization

How to deploy?

Meat of Unikernels

Look ma, no OS

What's in a unikernel?

Performance benefits

Security benefits

Unikernel examples

What's next?

Unmodified applications first

Local demo

In Amazon EC2

Summary

Unikernel - Library OS

Traditional VM

Unikernel

Unikernel - Library OS

Traditional VM

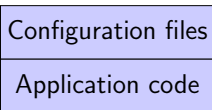
Unikernel

Configuration files

Unikernel - Library OS

Traditional VM

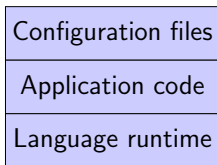
Unikernel



Unikernel - Library OS

Traditional VM

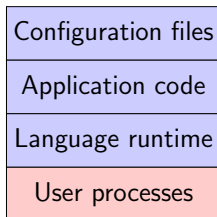
Unikernel



Unikernel - Library OS

Traditional VM

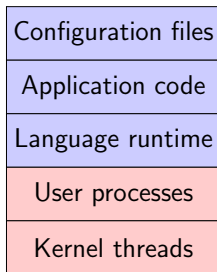
Unikernel



Unikernel - Library OS

Traditional VM

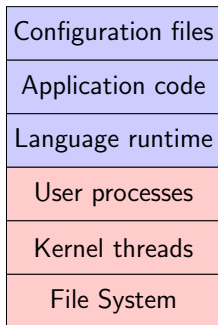
Unikernel



Unikernel - Library OS

Traditional VM

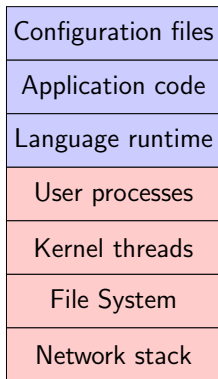
Unikernel



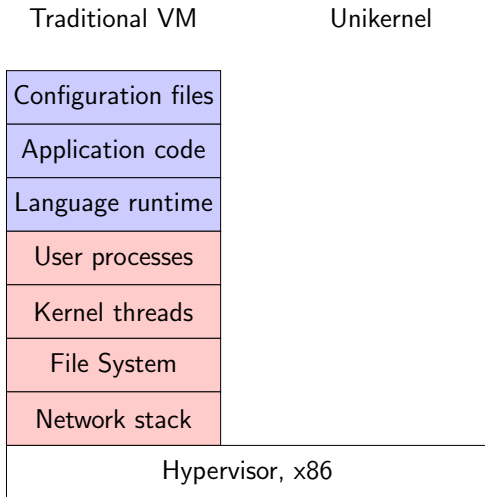
Unikernel - Library OS

Traditional VM

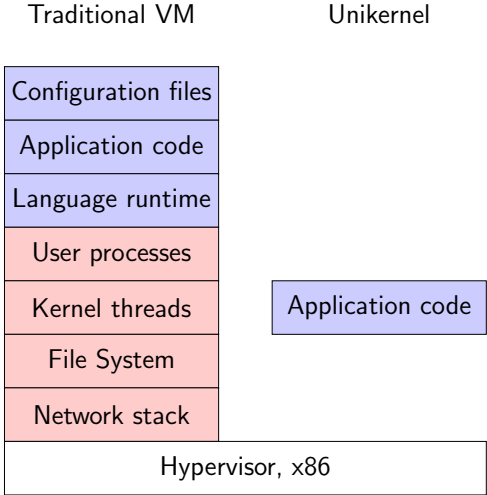
Unikernel



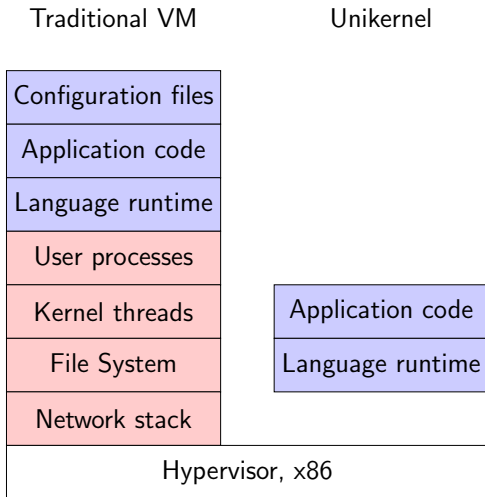
Unikernel - Library OS



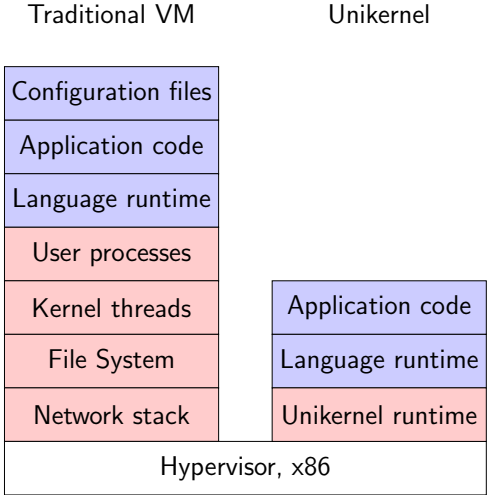
Unikernel - Library OS



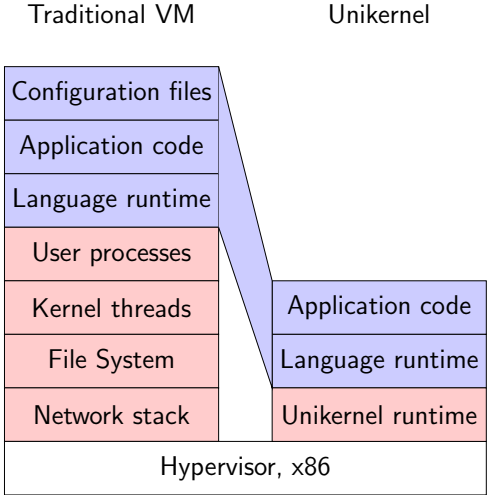
Unikernel - Library OS



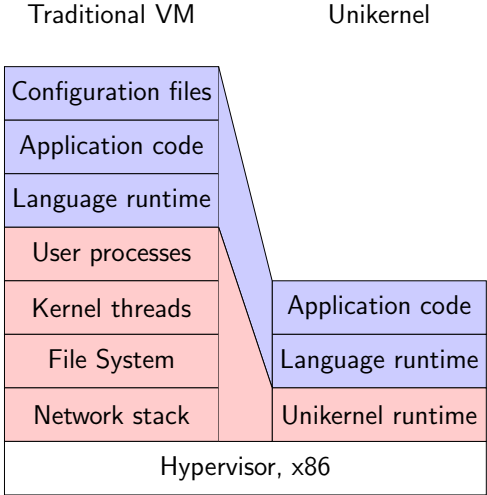
Unikernel - Library OS



Unikernel - Library OS



Unikernel - Library OS



Introduction

Roles of an IT organization

How to deploy?

Meat of Unikernels

Look ma, no OS

What's in a unikernel?

Performance benefits

Security benefits

Unikernel examples

What's next?

Unmodified applications first

Local demo

In Amazon EC2

Summary

Thanks

Unikernel has:

- ▶ One process, N threads.

Unikernel has:

- ▶ One process, N threads.
 - ▶ Haha! no context switches.

Unikernel has:

- ▶ One process, N threads.
 - ▶ Haha! no context switches.
- ▶ API for doing network and IO.

Unikernel has:

- ▶ One process, N threads.
 - ▶ Haha! no context switches.
- ▶ API for doing network and IO.
- ▶ Possibly application runtime:

Unikernel has:

- ▶ One process, N threads.
 - ▶ Haha! no context switches.
- ▶ API for doing network and IO.
- ▶ Possibly application runtime:



Where does performance come from?

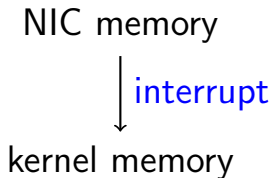
Where does performance come from?

Traditional style

NIC memory

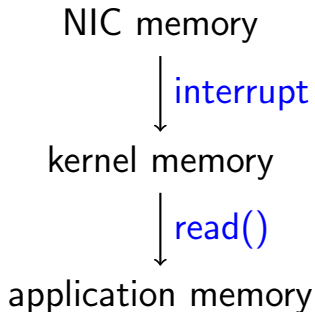
Where does performance come from?

Traditional style



Where does performance come from?

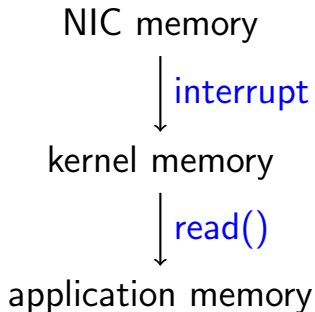
Traditional style



Where does performance come from?

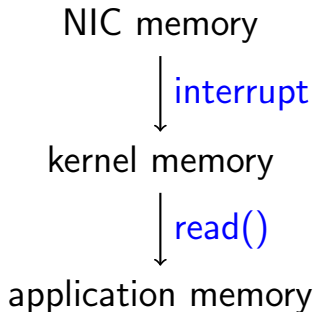
Traditional style

Unikernel style



Where does performance come from?

Traditional style

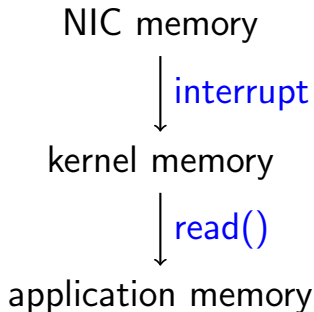


Unikernel style

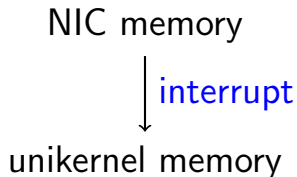
NIC memory

Where does performance come from?

Traditional style

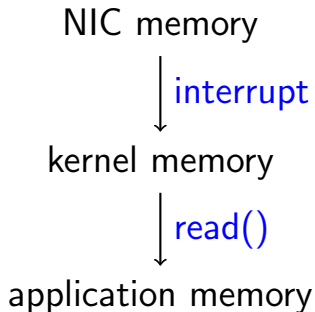


Unikernel style

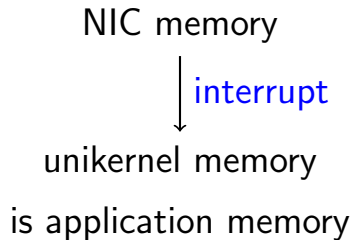


Where does performance come from?

Traditional style

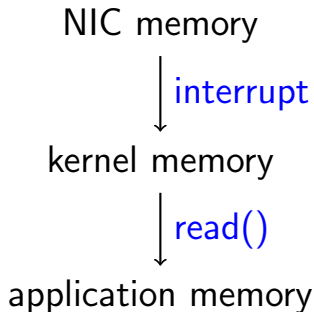


Unikernel style

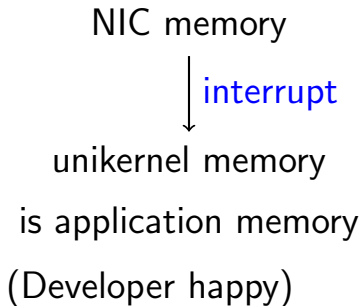


Where does performance come from?

Traditional style



Unikernel style



Introduction

Roles of an IT organization

How to deploy?

Meat of Unikernels

Look ma, no OS

What's in a unikernel?

Performance benefits

Security benefits

Unikernel examples

What's next?

Unmodified applications first

Local demo

In Amazon EC2

Summary

Thanks

Security

- ▶ Reduced attack surface area.

Security

- ▶ Reduced attack surface area.
 - ▶ No unneeded kernel modules.

Security

- ▶ Reduced attack surface area.
 - ▶ No unneeded kernel modules.
 - ▶ No Perl, shell...

Security

- ▶ Reduced attack surface area.
 - ▶ No unneeded kernel modules.
 - ▶ No Perl, shell...
 - ▶ Exploited environment is barely usable to useless.

Security

- ▶ Reduced attack surface area.
 - ▶ No unneeded kernel modules.
 - ▶ No Perl, shell...
 - ▶ Exploited environment is barely usable to useless.
- ▶ Some: language is type-safe.

Security

- ▶ Reduced attack surface area.
 - ▶ No unneeded kernel modules.
 - ▶ No Perl, shell...
 - ▶ Exploited environment is barely usable to useless.
- ▶ Some: language is type-safe.
- ▶ Some: can do 1 VM per request.

Security

- ▶ Reduced attack surface area.
 - ▶ No unneeded kernel modules.
 - ▶ No Perl, shell...
 - ▶ Exploited environment is barely usable to useless.
- ▶ Some: language is type-safe.
- ▶ Some: can do 1 VM per request.

Happy Mr Burns

Why only in 2013?

Virtualization provides uniform APIs for network and I/O.

Why only in 2013?

Virtualization provides uniform APIs for network and I/O.

E.g. *virtio* for KVM, *netfront/netback* for Xen.

- ▶ Small set of drivers to implement.
- ▶ Makes it economic to create unikernels.

Table of Contents

Introduction

Roles of an IT organization

How to deploy?

Meat of Unikernels

Look ma, no OS

What's in a unikernel?

Performance benefits

Security benefits

Unikernel examples

What's next?

Unmodified applications first

Local demo

In Amazon EC2

Summary

Classification

Classification

Niche : highly optimized applications/frameworks.

Classification

Niche : highly optimized applications/frameworks.

Generic : general-purpose applications:

Classification

Niche : highly optimized applications/frameworks.

Generic : general-purpose applications:

Fat : yes POSIX. Compatible.

Classification

Niche : highly optimized applications/frameworks.

Generic : general-purpose applications:

Fat : yes POSIX. Compatible.

Lean : no POSIX. Needs rewrite.

Mirage OS

Generic lean.

Mirage OS

Generic lean.

- ▶ OCaml-only.

Mirage OS

Generic lean.

- ▶ OCaml-only.
- ▶ \Rightarrow type-safe throughout the stack.

Mirage OS

Generic lean.

- ▶ OCaml-only.
- ▶ \Rightarrow type-safe throughout the stack.
- ▶ VM boot time $< 10ms$

Mirage OS

Generic lean.

- ▶ OCaml-only.
- ▶ \Rightarrow type-safe throughout the stack.
- ▶ VM boot time $< 10ms$
- ▶ \Rightarrow VM per request (inetd-like).

Mirage OS

Generic lean.

- ▶ OCaml-only.
- ▶ \Rightarrow type-safe throughout the stack.
- ▶ VM boot time $< 10ms$
- ▶ \Rightarrow VM per request (inetd-like).
- ▶ \Rightarrow security and economy.

Mirage OS

Generic lean.

- ▶ OCaml-only.
- ▶ \Rightarrow type-safe throughout the stack.
- ▶ VM boot time $< 10ms$
- ▶ \Rightarrow VM per request (inetd-like).
- ▶ \Rightarrow security and economy.
- ▶ Typical image sizes: 100s of KB.

Mirage OS

Generic lean.

- ▶ OCaml-only.
- ▶ \Rightarrow type-safe throughout the stack.
- ▶ VM boot time $< 10ms$
- ▶ \Rightarrow VM per request (inetd-like).
- ▶ \Rightarrow security and economy.
- ▶ Typical image sizes: 100s of KB.

Typical application: web services, data processing.

Generic Fat.

Generic Fat.

- ▶ Linux ABIs.

Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.

Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.
 - ▶ reported significant throughput increases on memcached.

Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.
 - ▶ reported significant throughput increases on memcached.
- ▶ Memory allocation tuning:

Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.
 - ▶ reported significant throughput increases on memcached.
- ▶ Memory allocation tuning:
 - ▶ JVM.

Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.
 - ▶ reported significant throughput increases on memcached.
- ▶ Memory allocation tuning:
 - ▶ JVM.
 - ▶ Memcached.

Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.
 - ▶ reported significant throughput increases on memcached.
- ▶ Memory allocation tuning:
 - ▶ JVM.
 - ▶ Memcached.
 - ▶ OOM Killer Manager!

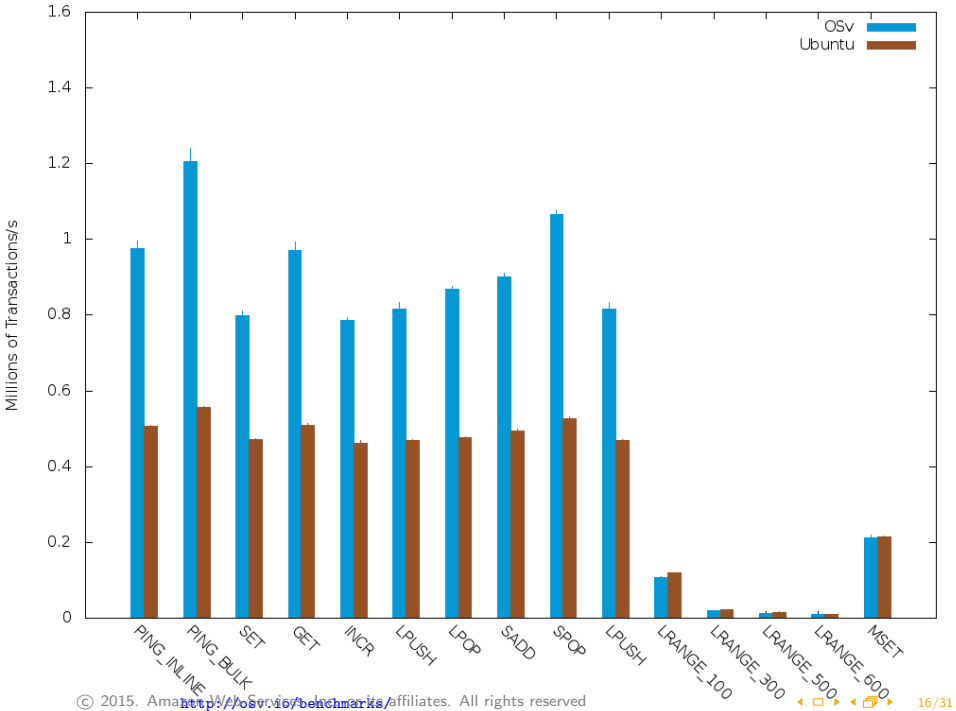
Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.
 - ▶ reported significant throughput increases on memcached.
- ▶ Memory allocation tuning:
 - ▶ JVM.
 - ▶ Memcached.
 - ▶ OOM Killer Manager!
- ▶ Subsecond boot times.

Generic Fat.

- ▶ Linux ABIs.
- ▶ Bonus: fast APIs.
 - ▶ reported significant throughput increases on memcached.
- ▶ Memory allocation tuning:
 - ▶ JVM.
 - ▶ Memcached.
 - ▶ OOM Killer Manager!
- ▶ Subsecond boot times.

Typical application: web services, data processing.



ClickOS

ClickOS

Niche.

ClickOS

Niche.

Optimized for network middleboxes:

ClickOS

Niche.

Optimized for network middleboxes:

- ▶ Firewalls.

ClickOS

Niche.

Optimized for network middleboxes:

- ▶ Firewalls.
- ▶ Intrusion Detection Systems.

Niche.

Optimized for network middleboxes:

- ▶ Firewalls.
- ▶ Intrusion Detection Systems.
- ▶ Load Balancers.

Niche.

Optimized for network middleboxes:

- ▶ Firewalls.
- ▶ Intrusion Detection Systems.
- ▶ Load Balancers.

Fun fact: Xen network optimization to the extreme.

Table of Contents

Introduction

Roles of an IT organization

How to deploy?

Meat of Unikernels

Look ma, no OS

What's in a unikernel?

Performance benefits

Security benefits

Unikernel examples

What's next?

Unmodified applications first

Local demo

In Amazon EC2

Summary

What do we do next?

Nobody likes rewrites.

What do we do next?

Nobody likes rewrites.

Take platforms (JVM, Erlang) and run unmodified apps.

- ▶ Fully immutable.

What do we do next?

Nobody likes rewrites.

Take platforms (JVM, Erlang) and run unmodified apps.

- ▶ Fully immutable.
- ▶ Faster deployment and rollbacks.

What do we do next?

Nobody likes rewrites.

Take platforms (JVM, Erlang) and run unmodified apps.

- ▶ Fully immutable.
- ▶ Faster deployment and rollbacks.
- ▶ Smaller attack surface.

Local demo

Contents:

- ▶ Take a JVM/Spring application.

Local demo

Contents:

- ▶ Take a JVM/Spring application.
- ▶ Demo on standard Linux.

Local demo

Contents:

- ▶ Take a JVM/Spring application.
- ▶ Demo on standard Linux.
- ▶ Generate the unikernel.

Local demo

Contents:

- ▶ Take a JVM/Spring application.
- ▶ Demo on standard Linux.
- ▶ Generate the unikernel.
- ▶ Run that unikernel locally.

Local demo

Contents:

- ▶ Take a JVM/Spring application.
- ▶ Demo on standard Linux.
- ▶ Generate the unikernel.
- ▶ Run that unikernel locally.

Takeaways:

Local demo

Contents:

- ▶ Take a JVM/Spring application.
- ▶ Demo on standard Linux.
- ▶ Generate the unikernel.
- ▶ Run that unikernel locally.

Takeaways:

- ▶ Small image size.

Local demo

Contents:

- ▶ Take a JVM/Spring application.
- ▶ Demo on standard Linux.
- ▶ Generate the unikernel.
- ▶ Run that unikernel locally.

Takeaways:

- ▶ Small image size.
- ▶ Trivial to implement.

Running in Amazon EC2

To create a VM image in AWS, do:

```
% qemu-img convert -f qcow2 -O raw 3.qemu 3.raw
```

```
% ./release-ec2.sh \  
  --override-image 3.raw \  
  --override-version 3 \  
  --region us-east-1
```


AMI in EC2

The screenshot shows the AWS Management Console interface. The left-hand navigation pane is expanded to the 'AMIs' section under 'IMAGES'. The main content area displays a table of AMIs owned by the user, with one entry selected: 'OSv-3' with AMI ID 'ami-05061469' and status 'available'. Below the table, the 'Details' tab is active, showing the following information:

| | | | |
|----------------------------|---------------------------------------|-------------------------|--|
| AMI ID | ami-05061469 | AMI Name | OSv-3 |
| Owner | 155766404048 | Source | 155766404048/OSv-3 |
| Status | available | State Reason | - |
| Creation date | November 17, 2015 at 4:43:13 PM UTC+1 | Platform | Windows |
| Architecture | x86_64 | Image Type | machine |
| Virtualization type | hvm | Description | [Copied ami-6612570c from us-east-1] OSv-3 |
| Root Device Name | /dev/sda1 | Root Device Type | ebs |
| RAM disk ID | - | Kernel ID | - |
| Product Codes | - | Block Devices | /dev/sda1=snap-d4ec2950:10:false:standard, |

AMI in EC2

The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there are browser tabs for 'Amazon WorkMail', 'EC2 Management Co...', and 'http://52.28.202.224/'. The address bar shows the URL 'https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#Inst...'. The navigation bar includes 'AWS', 'Services', 'Edit', and user information 'PowerUser/motiejus-Isengard ...' in 'Frankfurt' with a 'Support' link.

The main content area has a 'Launch Instance' button, 'Connect', and 'Actions' dropdown. Below this is a search bar with 'search : osv3' and 'Instance State : Running'. A table lists instances, with one instance selected:

| Name | Instance ID | Instance | Availability Z | Instance S | Status Checks | Public IP | Launch Time | Security Groups |
|------|-------------|----------|----------------|------------|---------------|---------------|-------------------------------|-----------------|
| osv3 | i-6b8f20d7 | c4.large | eu-central-1b | running | 2/2 check... | 52.28.202.224 | November 17, 2015 at 5:02:... | allow-http-ssh |

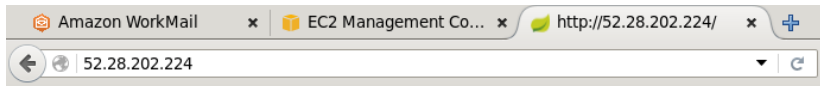
The instance details for 'osv3' are expanded, showing:

- Instance:** i-6b8f20d7 (osv3)
- Public DNS:** ec2-52-28-202-224.eu-central-1.compute.amazonaws.com

The details are organized into four tabs: Description, Status Checks, Monitoring, and Tags. The 'Description' tab is active, showing the following details:

| | | | |
|------------------------------|--|--------------------------|--|
| Instance ID | i-6b8f20d7 | Public DNS | ec2-52-28-202-224.eu-central-1.compute.amazonaws.com |
| Instance state | running | Public IP | 52.28.202.224 |
| Instance type | c4.large | Elastic IP | - |
| Private DNS | ip-172-31-4-84.eu-central-1.compute.internal | Availability zone | eu-central-1b |
| Private IPs | 172.31.4.84 | Security groups | allow-http-ssh . view rules |
| Secondary private IPs | | Scheduled events | No scheduled events |
| VPC ID | vpc-2a23c043 | AMI ID | OSv-3 (ami-05061469) |
| Subnet ID | subnet-e8a9a590 | Platform | windows |

Running OSv in EC2



Greetings from Spring Boot!

Try it out

- ▶ Run your unikernel in Free Tier right now.

Try it out

- ▶ Run your unikernel in Free Tier right now.
- ▶ t2.micro - \$0/month for 1 year.

Table of Contents

Introduction

Roles of an IT organization

How to deploy?

Meat of Unikernels

Look ma, no OS

What's in a unikernel?

Performance benefits

Security benefits

Unikernel examples

What's next?

Unmodified applications first

Local demo

In Amazon EC2

Summary

Summary

Library OS: OS embedded to your application.

Summary

Library OS: OS embedded to your application.

- ▶ Small \Rightarrow scale quickly.

Summary

Library OS: OS embedded to your application.

- ▶ Small \Rightarrow scale quickly.
- ▶ Very efficient \Rightarrow economic.

Summary

Library OS: OS embedded to your application.

- ▶ Small \Rightarrow scale quickly.
- ▶ Very efficient \Rightarrow economic.
- ▶ Reduced attack surface.

Summary

Library OS: OS embedded to your application.

- ▶ Small \Rightarrow scale quickly.
- ▶ Very efficient \Rightarrow economic.
- ▶ Reduced attack surface.
- ▶ Runs on public clouds: \Rightarrow try on EC2, for free.

Table of Contents

Introduction

- Roles of an IT organization

- How to deploy?

Meat of Unikernels

- Look ma, no OS

- What's in a unikernel?

- Performance benefits

- Security benefits

Unikernel examples

What's next?

- Unmodified applications first

- Local demo

- In Amazon EC2

Summary

Thanks

- ▶ Niels Brouwers (Amazon) for the right tools.
- ▶ Russel Pavlicek (Citrix) for spreading the word.

We're hiring!

- ▶ Check out amazon.jobs
- ▶ Also, contact me at motiejus@amazon.com

QA