Unikernel talk in 6 minutes or your next talk is Free

by Aaron Grothe

Introduction

What is a Unikernel?

A Unikernel is a single address space program. It combines what we normally think of as a kernel and the program into one running executable. So it can usually be run on a hypervisor or bare metal.

OPS.city

Let's build a unikernel to show how it goes

We'll use ops.city's product for this since it is pretty easy to use.

```
% curl https://ops.city/get.sh -sSfL | sh
```

This will install the basic components for the nanovms software

For time purposes I've already installed ops.city

OPS.city - (cont'd)

Now we need an application - We'll go ahead and use the nodejs example from ops.city - simple hello world web application

```
var http = require('http');
http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Hello World\n');
}).listen(8083, "0.0.0.0");
console.log('Server running at http://127.0.0.1:8083/');
```

OPS.city - (cont’d)

Now lets go ahead and create and fire up a unikernel

```
% ops pkg load eyberg/node:v14.2.0 -p 8083 -a hi.js
```

What does this do. It creates an image in the .ops/image folder named hi

It fires up qemu and starts that image with hi.js program that has been compiled into an image named node

OPS.city - (cont'd)

Let's try it out.

Open a web browser and go to the url

https://localhost:8083

Success???

OPS.city - (cont'd)

Let's take a look at the image. The image by default is in the ~/.ops/images folder - should be named node

```
% file ~/.ops/images/node
```

It is a raw image.

```
% du -h ~/.ops/image/node
```

It is 77mb for the complete program/os environment

OPS.city - (cont'd)

Now we'll run it outside of the ops environment. We have a raw image so we should be able to boot it up.

```
% qemu-system-x86_64 -drive
file=/home/grothe/.ops/images/node,format=raw,if=none,id
=hd0 -device virtio-blk,drive=hd0 -device
virtio-net,netdev=n0 -netdev
user,id=n0,hostfwd=tcp::8083-:8083 -nodefaults -no-reboot
-device isa-debug-exit -m 2G -display none -serial stdio
```

Yes it is fugly. Could simplify a bit but it gives you an idea.

OPS.city - (cont'd)

Let's try it out.

Open a web browser and go to the url

https://localhost:8083

Success???

OPS.city - (cont'd)

So what have we done here.

- We've created a simple hello world nodejs file and created an ops image
- We've run the ops image both inside and outside of the ops environment

OPS.city - (cont’d)

OPS.city has a lot of additional capabilities

We took an executable and built it into an image. We’ll take a look at some of the packages included with it.

```
% ops pkg list
```

Includes things like ruby, php, nodejs, java, python and so on

Types of Unikernels

- Generic Unikernels - these are able to run general programs. Can be in many different locations, other examples of this include RumpRUN
- Language Specific Unikernels - these are designed to support one specific language/runtime. E.g. Clive for Go programming language. Kind of a glorified Read-Evaluation-Print-Loop (REPL)
- Reduced O/S. An example of this is Hermitux, that is able to run Linux executables with a reduced O/S size
- Other - there are a lot of other types included as well

Benefits of Unikernels

- Less code
- Smaller environment
- Works well in a devops type of environment
- Reduced attack surface
- Better Security?

Drawbacks of Unikernels

- “Unikernels are unfit for production” - article by Bryan Cantrill that is pretty tough
  - Debugging can be tough, hello printf
- Limitations of program (no memory swapping, processes)
- Can be tough to do complicated programs
- Once a unikernel is compromised the attacker has full privs to the environment
- No concept of UserIDs, User permissions, memory checks and so on

Future of Unikernels

- There will be some consolidation in this area and more startups and companies closing
- Internet of Things (IoT) offers some potential new places for the deployment of Unikernels
- When a large company announces a project using a Unikernel will be interesting
- Kubernetes plus Unikernels might be very interesting :-)

Links

OPS.city

https://ops.city

Other Unikernels

- Clive - http://lsub.org/ls/clive.html
- HalVM - http://galois.com/project/halvm/
- IncludeOS - http://www.includeos.org/
- UniK - https://github.com/emc-advanced-dev/unik
- Hermitux - https://ssrg-vt.github.io/hermitux/