Unikernels

No OS? No Problem!

Kevin Sapper
WAMOS 2015
Outline

● What is a unikernel?
● Rump Kernels
● Demo
What is a unikernel?

A unikernel is a minimalist operating system kernel that is specifically designed to run a single application or a limited set of applications. It combines the kernel with the application and provides a layer of abstraction between the hardware and the application, allowing for a more efficient and secure execution environment. This approach eliminates the need for a traditional operating system and its associated layers, which can lead to significant performance improvements and reduced security risks.
What is a unikernel?
What is a unikernel?
Rump Kernel

Aims to run anywhere

- POSIX (and javascript)
- Linux
- Xen
- QEMU/KVM
- bare-metal

Anykernel

- Unmodified NetBSD code (~10^6 lines)
- Platform-independent glue code (~10^4 lines)
- Platform-specific code (~10^3 lines)
  e.g., Genode OS, Xen, userspace, bare-metal, ...
Rump Kernel

Aims to run anywhere

POSIX (and javascript)

Portable Operating System Interface defines API for Processes, Threads, Signals, Timer, Pipes, I/O, ....

POSIX-compliant OS are NetBSD, Contiki, PikeOS, ...

Anykernel

unmodified NetBSD code (~10^6 lines)

platform-independent glue code (~10^4 lines)

platform-specific code (~10^3 lines)
e.g., Genode OS, Xen, userspace, bare-metal, ...
Rump Kernel

Aims to run anywhere

POSIX (and javascript)
  Linux
  Xen
  QEMU/KVM
  bare-metal

Anykernel

unmodified NetBSD code (~10^6 lines)

platform-independent glue code (~10^4 lines)

platform-specific code (~10^3 lines)
e.g., Genode OS, Xen, userspace, bare-metal, ...
Rump Kernel

- Application(s)
- Userspace libraries
  - libc
    - Rump kernel calls
    - Syscall traps
  - File systems
  - TCP/IP
  - Device drivers
  - Syscalls
  - ... (unmodified POSIX userspace code, 10^9 lines)

- Glue code
  - Unmodified NetBSD code (~10^6 lines)
  - Platform-independent glue code (~10^4 lines)
  - Platform-specific code (~10^3 lines)
    - e.g., Genode OS, Xen, userspace, bare-metal,...
Rump Kernel

Can run **unmodified** POSIX applications.

No interrupts -> run to completion

Rump Kernels **CANNOT**:
- execute binaries
- schedule threads
- deal with privileged instructions
- use virtual memory
- handle page fault

Uses host threads and scheduling policy
Rump Clients

local

host kernel

local client
rump kernel
proc1

remote

host kernel

remote client
rump call stubs
proc1

host kernel

local client
rump kernel
proc2
Rump Clients

**local**
- host kernel
- local client
- rump kernel
  - proc1

**remote**
- host kernel
- remote client
- rump call stubs
  - proc1
- local client
  - rump kernel
  - proc2

**microkernel**
- host kernel
- µkernel client
  - proc1
- local client
  - rump kernel
  - proc2
Demo

local client

● Hello World
● Filesystem access
● Network access

remote client

● buildrump.sh (kernel + server)
● rumpctrl (userland utilities)