Unikernel

From Wikipedia, the free encyclopedia

Unikernels are specialised, single address space machine images constructed by using library operating systems.[1][2] A developer selects, from a modular stack, the minimal set of libraries which correspond to the OS constructs required for their application to run. These libraries are then compiled with the application and configuration code to build sealed, fixed-purpose images (unikernels) which run directly on a hypervisor or hardware without an intervening OS such as Linux or Windows.

Contents

Library operating systems

In a library operating system, protection boundaries are pushed to the lowest hardware layers, resulting in:

1. a set of libraries that implement mechanisms such as those needed to drive hardware or talk network protocols;
2. a set of policies that enforce access control and isolation in the application layer.

The first such systems were Exokernel and Nemesis in the late 1990s.

The library OS architecture has several advantages and disadvantages compared with conventional OS designs. One of the advantages is that since there is only a single address space, there is no need for repeated privilege transitions to move data between user space and kernel space. Therefore, a library OS can provide improved performance by allowing direct access to hardware without context switches. A disadvantage is that because there is no separation, trying to run multiple applications side by side in a library OS, but with strong resource isolation, can become complex.[3] In addition, device drivers are required for the specific hardware the library OS runs on. Since hardware is rapidly changing this creates the burden of regularly rewriting drivers to remain up to date.

OS virtualization can overcome these drawbacks on commodity hardware. A modern hypervisor provides virtual machines with CPU time and strongly isolated virtual devices. A library OS running as a virtual machine only needs to implement drivers for these stable virtual hardware devices and can depend on the hypervisor to drive the real physical hardware. However, protocol libraries are still needed to replace the services of a traditional operating system. Creating these protocol libraries is where the bulk of the work lies when implementing a modern library OS.[1]

Benefits and drawbacks

Unikernels have a number of benefits and drawbacks when compared to traditional operating systems.

- Improved security — By reducing the amount of code deployed, unikernels necessarily reduce the likely attack surface and therefore have improved security properties.[4][5]
- Small footprint — Unikernels have been shown to be around 4% the size of the equivalent code bases using a traditional OS.[6]
- Whole-system optimisation — Due to the nature of their construction, it is possible to perform whole-system optimisation across device drivers and application logic, thus improving on the specialisation.[7][8]
- Low boot times — Unikernels have been regularly shown to boot extremely quickly, in time to respond to incoming requests before the requests time-out.[9][10][11]

These benefits lend themselves to creating systems that follow the service-oriented or microservices software architectures.

However, the high degree of specialisation means that unikernels are unsuitable for the kind of general purpose, multi-user computing that traditional operating systems are used for. Adding additional functionality or altering a compiled unikernel is generally not possible and instead the approach is to compile and deploy a new unikernel with the desired changes.

Modern implementations

There are a number of new approaches to constructing unikernels, which are at varying degrees of maturity.

ClickOS

ClickOS[8][12] is a high-performance, virtualized software middle box platform based on open source virtualization. Early performance analysis shows that ClickOS VMs are small (5MB), boot quickly (as little as 20 milliseconds), add little delay (45 microseconds) and more than 100 can be concurrently run while saturating a 10Gb pipe on an inexpensive commodity server.

Clive

Clive[13] is an operating system designed to work in distributed and cloud computing environments, written in the Go programming language.

Drawbridge

Drawbridge is a research prototype of a new form of virtualization for application sandboxing. Drawbridge combines two core technologies: a picoprocess, which is a process-based isolation container with a minimal kernel API surface, and a library OS, which is a version of Windows enlightened to run efficiently within a picoprocess.[14]

HaLVM

The Haskell Lightweight Virtual Machine (HaLVM (https://galois.com/project/halvm/)) is a port of the Glasgow Haskell Compiler tool suite that enables developers to write high-level, lightweight VMs that can run directly on the Xen hypervisor.

IncludeOS

IncludeOS (http://www.includeos.org) is a minimal, service oriented, open source, includeable library operating system for cloud services. Currently a research project for running C++ code on virtual hardware.

LING

LING[15] is a unikernel based on the Erlang/OTP and understands .beam files. Developers can create code in Erlang and deploy it as LING unikernels. LING removes the majority of vector files, uses only three external libraries and no OpenSSL.

MirageOS

MirageOS[16] is a clean-slate library operating system that constructs unikernels for secure, high-performance network applications across a variety of cloud computing and mobile platforms. There are now more than 100 MirageOS libraries[17] and a growing number of compatible libraries within the wider OCaml ecosystem.

OSv

OSv is a new OS designed specifically for cloud VMs from Cloudius Systems.[18] Able to boot in less than a second, OSv is designed from the ground up to execute a single application on top of any hypervisor, resulting in superior performance, speed and effortless management. Support for C, JVM, Ruby and Node.js application stacks is available.

Rumprun

Rumprun (http://repo.rumpkernel.org/rumprun) is a software stack which enables running existing unmodified POSIX software as a unikernel. Rumprun supports multiple platforms, including bare hardware and hypervisors such as Xen and KVM. It is based on rump kernels which provide free, portable, componentized, kernel quality drivers such as file systems, POSIX system call handlers, PCI device drivers, a SCSI protocol stack, virtio and a TCP/IP stack.[19]

Runtime.js

Runtime.js (http://runtimejs.org/) is an open-source library operating system for the clouds that runs on JavaScript VM, could be bundled up with an application and deployed as a lightweight and immutable VM image. Runtime.js built on the V8 Javascript engine and currently supports QEMU/KVM hypervisor.

References

1. "Unikernels: Rise of the Virtual Library Operating System". Retrieved 31 August 2015.
2. "Unikernel.org". *Unikernel.org*. Retrieved 1 December 2015.
3. Chia-Che, Tsai; Arora, Kumar-Saurabh; Bandi, Nehal; Jain, Bhushan; Jannen, William; John, Jitin; Kalodner, Harry; Kulkarni, Vrushali; Oliviera, Daniela; Porter, Donald E. (2014). "Cooperation and Security Isolation of Library OSes for Multi-process Applications" (PDF). *Proceedings of the Ninth European Conference on Computer Systems (EuroSys)*. doi:10.1145/2592798.2592812.
4. "Why Unikernels Can Improve Internet Security". Retrieved 31 August 2015.
5. Madhavapeddy, Anil; Mortier, Richard; Charalampos, Rotsos; Scott, David; Singh, Balraj; Gazagnaire, Thomas; Smith, Steven; Hand, Steven; Crowcroft, Jon (March 2013). "Unikernels: Library Operating Systems for the Cloud" (PDF). *SIGPLAN Notices (ASPLOS 13)* 48 (4). doi:10.1145/2499368.2451167.
6. Kaloper-Meršinjak, David; Mehnert, Hannes; Madhavapeddy, Anil; Sewell, Peter (2015). "Not-Quite-So-Broken TLS: Lessons in Re-Engineering a Security Protocol Specification and Implementation" (PDF). *Proceedings of the 24th USENIX Security Symposium (USENIX Security 15)*.
7. Madhavapeddy, Anil; Mortier, Richard; Sohan, Ripduman; Gazagnaire, Thomas; Hand, Steven; Deegan, Tim; McAuley, Derek; Crowcroft, Jon (2010). "Turning Down the LAMP: Software Specialisation for the Cloud" (PDF). *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*.
8. Martins, Joao; Mohamed, Ahmed; Raiciu, Costin; Huici, Felipe (2013). "Enabling Fast, Dynamic Networking Processing with ClickOS" (PDF). *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. doi:10.1145/2491185.2491195.
9. "Just-in-Time Summoning of Unikernels (v0.2)". *Magnus Skjegstad*. Retrieved 30 August 2015.
10. "Zerg". *Zerg — an instance per request demo*. Retrieved 30 August 2015.
11. Madhavapeddy, Anil; Leonard, Thomas; Skjegstad, Magnus; Gazagnaire, Thomas; Sheets, David; Scott, David; Mortier, Richard; Chaudhry, Amir; Singh, Balraj; Ludlam, Jon; Crowcroft, Jon; Leslie, Ian (2015). "Jitsu: Just-In-Time Summoning of Unikernels" (PDF). *the 12th USENIX Conference on Networked Systems Design and Implementation (NSDI)*. ISBN 978-1-931971-218.
12. "ClickOS and the Art of Network Function Virtualization" (PDF). Retrieved 31 August 2015.
13. "The Clive Operating System" (PDF). Retrieved 31 August 2015.
14. "Drawbridge". *Microsoft Research*. Retrieved 30 August 2015.
15. "Erlang on Xen: at the heart of super-elastic clouds". Retrieved 31 August 2015.
16. "MirageOS: A programming framework for building type-safe, modular systems". Retrieved 31 August 2015.
17. "MirageOS TROVE". Retrieved 31 August 2015.
18. Kivity, Avi; Costa, Glauber; Enberg, Pekka; Har'El, Nadav; Marti, Don; Zolotarov, Vlad (June 2014). "OSv: Optimizing the Operating System for Virtual Machines" (PDF). *2014 USENIX*

Annual Technical Conference.
19. "Rump Kernels". *rumpkernel.org*. Retrieved 31 August 2015.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Unikernel&oldid=693989903"

Categories: Operating system kernels

- This page was last modified on 6 December 2015, at 09:56.