NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

DIVISION OF COMPUTER SCIENCE

Efficient file sharing between host and unikernel

DIPLOMA THESIS

Fotios Zafeiris M. Xenakis

Supervisor: Nectarios Koziris
Professor, NTUA

Athens, October 2020

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Efficient file sharing between host and unikernel

DIPLOMA THESIS

Fotios Zafeiris M. Xenakis

Supervisor: Nectarios Koziris
Professor, NTUA

Approved by the three-member examining committee on October 30, 2020.

.....	.....	.....
Nectarios Koziris	Georgios Goumas	Dionisios Pnevmatikatos
Professor, NTUA	Assistant professor, NTUA	Professor, NTUA

Athens, October 2020

.....
Fotios Zafeiris M. Xenakis
Dipl. in Electrical and Computer Engineering, NTUA

Abstract

Cloud computing is the dominant approach to compute infrastructure, established on the technology of virtualization. As the cloud expands, efficient utilization of its compute resources by software becomes imperative. One solution towards that are *unikernels*, operating system kernels specialized to run a single application, sparing resources compared to a general-purpose kernel. Efficient access from virtualized guests to the underlying host's resources is a substantial challenge in virtualization. In this aspect, *virtio* has been a significant contribution, as a specification of paravirtual devices enabling efficient usage of the host's resources. For host-guest file sharing, *virtio-fs* has been proposed, as a virtio device offering guest access to a file system directory on the host, providing high performance and local file system semantics.

This thesis is concerned with the implementation and evaluation of virtio-fs in the context of the OSvunikernel. We demonstrate that combining the two offers great benefits, both with regard to performance achieved, which is comparable to local file systems, and the operational aspect in a cloud context. Moreover, the above are carried out fully within the open-source project behind the unikernel we based our work on. This way, the resulting product gains practical value, being a useful contribution to the project, thus achieving a pivotal, non-technical goal. Furthermore, we explore how open-source software projects and the communities around them work, as we become active members of one.

Keywords

virtualization, cloud, file system, unikernel, virtio, OSv, virtio-fs, QEMU

Acknowledgements

For this work, signalling the completion of a long course, I would like to thank the members of the computing systems laboratory, under whose auspices it was carried out. Most of all, I want to thank them, as well as other members of the ECE school, for their teaching, their genuine interest and for cultivating the spirit of an engineer in me.

Moreover, I owe a big thank you to the people in the OSv and virtio-fs communities for their support, their guidance and their time, but more importantly for their openness, their spirit and work which sparked this contribution.

Finally, those I am most grateful for are my family and friends, who always stand on my side, bear with and support me and without whom nothing could be accomplished.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 Why unikernels

This past decade, we have witnessed the domination of the cloud computing model. The latter has become a cornerstone in many application fields, enabling previously impractical architectures, for a growing number of users. Its broad adoption has also led to a need for managing compute infrastructure of unprecedented scale. The technology behind cloud in its current form is that of virtualization: "curving" multiple, virtual, machines out of a single host system, at will.

Despite the changes brought about by the advent of cloud computing, the conventional slices of the software stack (the operating system being a stark example) have been minimally affected by it. Thus, this part of infrastructure is dominated by general-purpose OS's, like Linux, carrying decades-old design decisions and legacy. This is undoubtedly an indication of the difficulty involved in their implementation, as well as the value contained in current systems, accumulated over a long series of improvements.

A combination of factors makes it obvious that general-purpose operating systems are not the optimal solution for modern application requirements. These factors include the transition from physical to virtual machines, invalidating the assumption of exclusive ownership over the underlying hardware resources. Another contributing element is the rapid bridging of the gap between I/O and processing performance, rendering prohibitive the involvement of the OS in a high-performance application's data path, as backed by the growing popularity of frameworks like [2, 5]. Finally, the large scale of the infrastructure at hand, as mentioned above, amplifies the need for optimized efficiency, since sub-optimum operation bears huge costs.

Unikernels are a notable proposal for substituting conventional operating systems in guests, when those are used to run only a single application. Those result from

merging an application with all supporting elements it requires (typically offered by an OS), in the form of libraries, in a common address space. The executable images produced are run as virtual machines, achieving higher efficiency due to their reduced size (thus faster transport and startup [11], as well as lower storage space requirements), lower memory requirements and more performant system operations, e.g. due to the elimination of mode switches and a simplified security model.

1.1.2 Why shared file system and virtio-fs

Typically, file systems are local, implemented over block devices, inside an operating system kernel. There are cases though which benefit from sharing a common file system across several machines. One such case is that of virtualization, where host and guest share a file system (usually one pre-existing on the host). One example of how useful this can be are virtual machines that get reconfigured by the host, while another one is found in short-lived virtual machines which read input or configuration data and write output data to a common file system.

Virtio-fs is a recent effort in the field of shared file systems, the first one built to target virtualization environments exclusively. This specialization allows it to have no dependency on network protocols or (virtual) network infrastructure, resulting in lower requirements for a guest utilizing it, in addition to improved performance and local file system semantics [19]. What plays an important role in achieving these is the use of a shared memory area between the host and the guest, where file contents are mapped, the so called DAX (Direct Access) window.

1.2 Goals

This work aims to explore the possibilities offered by virtio-fs in a unikernel context. Specifically, we choose OSv and extend its existing, elementary virtio-fs implementation by adding read-only support for the DAX window as well as support for booting off of a virtio-fs file system. These render its use practical across a multitude of cases. Moreover, we evaluate our implementation's performance compared to that of an array of other file systems, in various representative scenarios.

Our goals though extend to non-technical ones, having to do with the free / open source model of software development. Since both projects involved use this model, we consider it an opportunity and a moral obligation for all of our work on OSv to be contributed back to the project. This way, other users may benefit from it, extending its value beyond the academic plain, while the respective project communities gain a new, active member.

1.3 Related work

In the years since the introduction of MirageOS [12], a plethora of unikernel frameworks have made their appearance, forming a diverse ecosystem. Some of them, beyond OSv are RumpRun [16], IncludeOS [3], HermitCore [10], and HermiTux [14], ukl [15], Toro [6] and Nanos [4].

The main pre-existing alternative to virtio-fs is VirtFS [9] which is based on the 9P network protocol [1]. Virtio-fs is still a young project, under active development, so there are relatively few implementations available. On the host side, apart from its main implementation in QEMU [18], there exists a complete implementation of it in cloud-hypervisor [8], an alternative virtiofsd implementation named memfsd from the nabla containers community [13] and an implementation for firecracker [7] which has not been accepted by the project for now. On the guest side, apart from the reference implementation in Linux [17], there is an implementation in Toro (a Pascal unikernel) [6], as well as another for Windows [20].

1.4 Thesis structure

A thorough description of the technical background of this thesis, from cloud use cases to the supporting technologies of virtio-fs is included in the second chapter. The third chapter concerns the specifics of the implementation of our extensions to OSv, followed by its evaluation, describing the methodology and examining the results in the fourth chapter. Finally, the fifth chapter contains a brief review in addition to directions for future work.

Bibliography

[1] *9P website*. `http://9p.cat-v.org/`, accessed 30/10/2020.

[2] *DPDK website*. `https://www.dpdk.org/`, accessed 30/10/2020.

[3] *IncludeOS website*. `https://www.includeos.org/`, accessed 30/10/2020.

[4] *Nanos website*. `https://nanos.org/`, accessed 30/10/2020.

[5] *SPDK website*. `https://spdk.io/`, accessed 30/10/2020.

[6] *Toro Kernel website*. `https://torokernel.io/`, accessed 30/10/2020.

[7] Agache, Alexandru, Marc Brooker, Alexandra Iordache, Anthony Liguori, Rolf Neugebauer, Phil Piwonka, and Diana Maria Popa: *Firecracker: Lightweight virtualization for serverless applications*. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 419–434, Santa Clara, CA, February 2020. USENIX Association, ISBN 978-1-939133-13-7. `https://www.usenix.org/conference/nsdi20/presentation/agache`.

[8] Cloud Hypervisor contributors: *Cloud Hypervisor repository*. `https://github.com/cloud-hypervisor/cloud-hypervisor`.

[9] Jujjuri, Venkateswararao, Eric Van Hensbergen, Anthony Liguori, and Badari Pulavarty: *Virtfs - a virtualization aware file system pass-through*. In *Proceedings of the Linux Symposium*, pages 109–120, 2010.

[10] Lankes, Stefan, Simon Pickartz, and Jens Breitbart: *Hermitcore: A unikernel for extreme scale computing*. In *Proceedings of the 6th International Workshop on Runtime and Operating Systems for Supercomputers*, ROSS '16, New York, NY, USA, 2016. Association for Computing Machinery, ISBN 9781450343879. `https://doi.org/10.1145/2931088.2931093`.

[11] Madhavapeddy, Anil, Thomas Leonard, Magnus Skjegstad, Thomas Gazagnaire, David Sheets, Dave Scott, Richard Mortier, Amir Chaudhry, Balraj Singh, Jon Ludlam, Jon Crowcroft, and Ian Leslie: *Jitsu: Just-in-time summoning of unikernels*. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 559–573, Oakland, CA, May 2015.

USENIX Association, ISBN 978-1-931971-218. https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/madhavapeddy.

[12] Madhavapeddy, Anil, Richard Mortier, Charalampos Rotsos, David Scott, Balraj Singh, Thomas Gazagnaire, Steven Smith, Steven Hand, and Jon Crowcroft: *Unikernels: Library operating systems for the cloud*. In *Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '13, page 461–472, New York, NY, USA, 2013. Association for Computing Machinery, ISBN 9781450318709. https://doi.org/10.1145/2451116.2451167.

[13] Nabla Containers contributors: *memfsd repository*. https://github.com/nabla-containers/qemu-virtiofs.

[14] Olivier, Pierre, Daniel Chiba, Stefan Lankes, Changwoo Min, and Binoy Ravindran: *A binary-compatible unikernel*. In *Proceedings of the 15th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, VEE 2019, page 59–73, New York, NY, USA, 2019. Association for Computing Machinery, ISBN 9781450360203. https://doi.org/10.1145/3313808.3313817.

[15] Raza, Ali, Parul Sohal, James Cadden, Jonathan Appavoo, Ulrich Drepper, Richard Jones, Orran Krieger, Renato Mancuso, and Larry Woodman: *Unikernels: The next stage of linux's dominance*. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, pages 7–13. ACM, 2019.

[16] RumpRun contributors: *RumpRun repository*. https://github.com/rumpkernel/rumprun.

[17] virtio-fs contributors: *virtio-fs linux repository*. https://gitlab.com/virtio-fs/linux.

[18] virtio-fs contributors: *virtio-fs qemu repository*. https://gitlab.com/virtio-fs/qemu.

[19] virtio-fs contributors: *virtio-fs website*. https://virtio-fs.gitlab.io/, accessed 30/10/2020.

[20] virtio-win contributors: *virtio-win repository*. https://github.com/virtio-win/kvm-guest-drivers-windows.