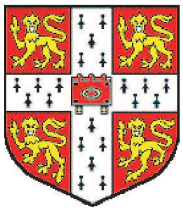
Virtual Devices for Virtual Machines

Andrew Kent Warfield

Clare Hall



A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
UK

Email: andrew.warfield@cl.cam.ac.uk

May 5, 2006

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.
This dissertation is not substantially the same as any I have submitted for a degree or diploma or any other qualification at any other university.
No part of this dissertation has already been, or is currently being submitted for any such degree, diploma or other qualification.
This dissertation does not exceed sixty thousand words.

Virtual Devices for Virtual Machines
Summary

Andrew Kent Warfield
Clare Hall College

May 5, 2006

Computer systems research has recently seen a huge resurgence of interest in hardware virtualization, a software technique originally developed to manage mainframe computers in the 1960s. Using virtual machines (VMs), a commodity PC may be divided into isolated "slices", each perceiving that it is executing on separate physical hardware. This thesis considers the effective virtualization of I/O devices on commodity hardware and presents an approach that allows developers to add new functionality to a piece of hardware as a software extension, running in an isolated VM. The new virtual device is presented to the OS using the existing virtualized hardware interface, allowing extensions to be easily applied across a wide range of operating systems.

Isolating extensions in their own virtual machines is effectively a "sledgehammer" version of the system decomposition that was attempted by microkernels through the 1980s and 1990s. The VM-based approach has the benefit of demonstrably working with a broad range of existing systems, and allowing developers to build extensions in their OS and language of choice. It concurrently maintains the benefits of isolation: extension crashes are protected from disrupting the rest of the system, and extension software has a clean and simple interface to devices. This thesis develops this work by demonstrating the construction of a set of device extensions for various pieces of hardware. Additionally, this thesis demonstrates that device extensions may be aggregated within cluster environments to implement *device services*, allowing specific device types to be treated as a service throughout a cluster of virtual machines.

Several examples are presented to validate the flexibility of device extensions: A packet symmetry-based rate limiter demonstrates a single-host network extension that prevents VMs from issuing common forms of denial of service attacks. Parallax, a distributed storage system for VMs, demonstrates the implementation of a device service for the management of storage within a cluster. Finally, device extensions are combined with other virtualization projects to develop deployable system-wide extensions to virtual hardware.

Acknowledgements

I am indebted to my advisor, Steven Hand, who has acted as both friend and mentor throughout my doctoral work. Steve is a gifted teacher and frequently provided the right guidance at the right time. I hope to be able to match his standard in supervising my own graduate students in the future.

The packet symmetry work in Chapter 4 stems from collaboration with Christian Kreibich, Jon Crowcroft, Steve Hand, and Ian Pratt. James Bulpin developed two excellent implementations of the distributed block store used by Parallax in Chapter 5, and Christian Limpach provided the file system measurements presented in Figure 5.6. The work in Chapter 6 is a result of collaborations in combining the soft device framework with other research projects. The extensions for debugging are in collaboration with Alex Ho, while taint tracking is the result of ongoing collaboration with Alex, Michael Fetterman, Christopher Clark, and Steve Hand.

I have been very lucky to be a member of the Systems Research Group during a period of time where so much great work has been in progress, and am delighted to have been able to participate in such a broad range of projects during my time at Cambridge. In particular, James Bulpin, Julian Chesterfield, Christopher Clark, Jon Crowcroft, Michael Dales, Tim Deegan, Michael Fetterman, Keir Fraser, Alex Ho, Eva Kalyvianaki, Christian Kreibich, Christian Limpach, Anil Madhavapeddy, Rolf Neugebauer, Ian Pratt, Russ Ross, and Steven Smith have all provided insightful discussion and generally been really good fun to work with.

Thanks to Steve, Jon, Christian K., Julian, Rolf, Tim D., Alex, and Richard Mortier for providing feedback on drafts of this thesis.

Thanks finally to the staff of the Castle Pub, who have provided the impetus for more interesting research than they can possibly realize.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.