# Performance Optimization for Latency-Sensitive Workloads in Virtual Systems

by

Matthew Welch

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved November 2015 by the
Graduate Supervisory Committee:

Violet Syrotiuk, Chair
Gil Speyer
Carole-Jean Wu

ARIZONA STATE UNIVERSITY

December 2015

# ABSTRACT

Network workloads have been increasingly migrating to virtual environments. Most of the time, there is *NO TUNING* utilized in these systems.

This is a bad idea and I'm gonna tell you why.

*To my mother who could never answer my questions of "What if...?"*

*To my wife who's unwavering support,*
*boundless love,*
*and endless patience*
*have buoyed my spirits and kept me going.*

*To my children Alyssa, Sarah, Connor, and Sean who give me purpose.*

# ACKNOWLEDGEMENTS

Insert acknwledgements here.

Be nice to those that have helped you along the way.

# TABLE OF CONTENTS

Page

# LIST OF TABLES

Table Page

# LIST OF FIGURES

Figure Page

# Chapter 1

# INTRODUCTION

Introduction goes here. Make it good.

# Chapter 2

# RELATED WORK

Virtualization has been an important part of computing as soon as more than one user wanted to share the expensive mainframe computer. This section will describe some of the work in the field of virtualization that has brought us to the current state of the art. We will also discuss some of the subsystems in Linux and processor hardware that enable virtualization.

## 2.1 History of Virtualization

Virtualization has been an important component in computing since the invention of virtual memory to support multiprogramming on mainframes in the 60's. Without going into too much detail:

1. Virtualization in early mainframes (IBM 360)
2. Early commercial virtualization
3. Entrance of Virtualization in x86
4. Virtualization as growth driver for datacenters and enterprise networks
5. Containers
6. Desktop virtualization
7. NFV

"The Linux Foundation Announces Project to Advance Real-Time Linux" [linux_foundation_2015] the Linux Foundation is the body that guides and sponsors the direction of Linux.

Realtime Linux has been struggling to find funding until recently [lwn.net reference here] This project is important because realtime Linux will also improve virtualization determinism. this will ensure that the realtime branch of Linux is supported going forward.

## 2.2 Architecture

Virtualization has been around in many forms. Virtual machines and containers, specifically Docker, are currently very popular so we will discuss the architecture of these types of virtualization here.

### *2.2.1 Virtual Machines*

The architectures available in virtual environments are as widely varied as the types of hardware they seek to emulate. Many of the concepts we use regarding virtual machines, however, originated in a seminal 1974 paper by Popek and Goldberg [**Popek:1974:FRV:361011.361073**]. Although virtual machines had already been implemented on ”third-generation computer systems” such as the IBM 360/67, Popek and Goldberg sought to establish formal requirements for and prove whether other systems of that era were capable of supporting a ”virtual machine monitor” [**Popek:1974:FRV:361011.361073**]. At the time of that writing, their analysis was on the possibility of a Virtual Machine Monitor (VMM), but the term hypervisor has largely come to replace VMM as the name of a software system that allocates, monitors, and controls virtual machine resources as an intermediary between the hardware and the virtual machine's operating system. What follows in this section is an illustration of some important concepts in hypervisor and virtual machine architecture and the work that led to them.

### *2.2.2 Type 1 vs. Type 2 Hypervisors*

# Chapter 3

# EXPERIMENTAL DESIGN

Paragraph here.

## 3.1 Architecture

Paragraph here

### *3.1.1 Preempt-RT*

Subection

### *3.1.2 Kernel Tuning*

Subsection

### *3.1.3 Benchmark Analysis*

subsection

# Chapter 4

# RESULTS

# Chapter 5

# CONCLUSIONS & FUTURE WORK

Conclusions text here

## 5.1 Future Work

### *5.1.1 DPDK*

DPDK is, like, super awesome and should be used by everyone!

### *5.1.2 Virtual Functions*

Virtual functions would be really good for scaling

# REFERENCES

[1] G. J. Popek and R. P. Goldberg, “Formal requirements for virtualizable third generation architectures”, *Commun. ACM*, vol. 17, no. 7, pp. 412–421, Jul. 1974, ISSN: 0001-0782. DOI: 10.1145/361011.361073. [Online]. Available: http://doi.acm.org/10.1145/361011.361073.