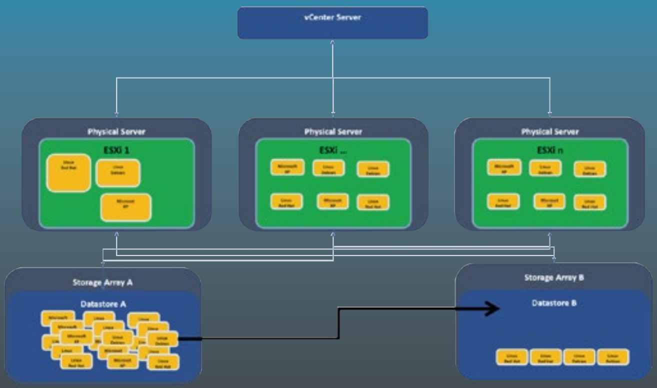
TECHNOLOGY IN ACTION™



Running ESXi on a Raspberry Pi

Installing VMware ESXi on Raspberry Pi 4 to run Linux virtual machines

—

Thomas Fenton
Patrick Kennedy

Apress®

Running ESXi on a Raspberry Pi

Installing VMware ESXi on Raspberry Pi 4 to run Linux virtual machines

Thomas Fenton
Patrick Kennedy

Apress®

Running ESXi on a Raspberry Pi: Installing VMware ESXi on Raspberry Pi 4 to run Linux virtual machines

Thomas Fenton
Ridgefield, WA, USA

Patrick Kennedy
Austin, TX, USA

ISBN-13 (pbk): 978-1-4842-7464-4
ISBN-13 (electronic): 978-1-4842-7465-1
https://doi.org/10.1007/978-1-4842-7465-1

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Aaron Black
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-7464-4. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

I dedicate this book to my mother, the noted author and historian Cleta Fenton. My father Darrell Fenton, who is the most honest man I have ever known. And, most importantly, my granddaughter Eleanor Jane Fenton, who brings so much joy to my life.

I would also like to publicly thank David Harriman and Josh Fenton for the work they did on the book. Their wordsmithing made the book much more readable.

– Tom Fenton

I dedicate this book to my parents and family for always supporting me and my nieces Gracen and Emerson and nephew Pierce for bringing joy while working on this. Also, a quick thanks to the STH (ServeTheHome.com) team, our audience, and those in the industry for supporting our efforts over the past 12 years.

– Patrick Kennedy

Table of Contents

About the Authors

Tom Fenton has worked with virtualization for over two decades. He's held staff and senior-level positions at VMware, IBM, ControlUp, and other high-tech companies. He wrote the first training course for Horizon, VMware's virtual desktop product. He is a frequent contributor to StorageReview.com, *Virtualization & Cloud Review* magazine, and other blogs and websites. He lives in the Pacific Northwest and enjoys skiing, snowboarding, hiking, and trail riding.

Patrick Kennedy after the better part of a decade working in a management consulting firm with many technology clients started ServeTheHome.com. Over the past 12+ years, STH has grown to be the place to learn about new server, storage, and networking technologies as Patrick and the STH team look at the latest solutions.

Patrick is a Silicon Valley, California, guy now roaming the hill country surrounding Austin, Texas.

About the Technical Reviewer

Darren Hirons is Lead Solutions Engineer at VMware. He has over 20 years of experience working with a wide variety of VMware products and other datacenter technologies. His focus more recently has been in the end user computing space, where he helps customers architect solutions to address business challenges.

Acknowledgments

We freely acknowledge that the work and information in this book has been greatly shaped by the VMware ESXi on Arm community whose legions are far too numerous to mention. Their work and dedication to this project can be found in the many forums that they contribute to. A special shout-out needs to go to William Lam, a Senior Staff Solution Architect at VMware, for documenting so much about VMware technologies and, in particular, ESXi on Arm on his website `https://williamlam.com/`. William's tireless work with the community is simply amazing!

Preamble

All of the author's proceeds from this book will go to Mount Hood Kiwanis Camp (`https://mhkc.org/`). This scenic camp sits on 22 acres of the US Forest Service land at the base of Mount Hood. It offers summer and winter camps that help empower children and adults with disabilities through overnight, outdoor, recreational programming. Both of us have had outstanding experiences with the Kiwanis organization over the years and gladly support this camp.

CHAPTER 1

Setting the Framework for ESXi on Arm on Pi

In this chapter, in order to set the stage for the rest of the book, we will look at the datacenter before virtualization was available. We will then discuss virtual machines (VMs) and the classifications of the types of hypervisors that are commonly in use. We will also give you a little background of VMware, the company that developed ESXi. As ESXi is an integral part of the vSphere family, we will also touch on some of the other vSphere products.

The Datacenter Before Virtualization

Before delving into topics like ESXi, Arm servers, and virtual machines (VMs), we need to go back and talk about what the datacenter looked like in 2001, right before virtualization became a mainstay. This was an exciting time; the Internet was just becoming widely used, and multitiered architecture was the *de facto* standard for delivering applications and services.

A typical three-tier architecture (see Figure 1-1) was composed of a presentation tier, a domain logic tier, and a data storage tier. By separating an application into these layers of abstraction, any single tier could, in theory, be modified without affecting any of the others. For availability

T. Fenton and P. Kennedy, *Running ESXi on a Raspberry Pi*,
https://doi.org/10.1007/978-1-4842-7465-1_1

and performance reasons, each layer within this architecture could be composed of multiple servers delivering the services needed for the respective layer.

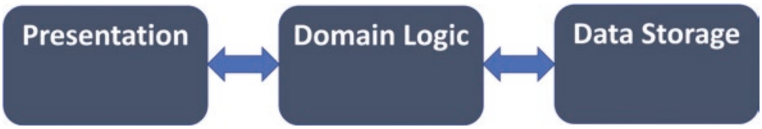


Figure 1-1. *Three-Tier Architecture*

During this time, a widely accepted best practice in datacenters was to have one physical server run a single service. The servers and operating systems (OSs) that ran on them didn't have quality of service (QoS) features that would prevent an application from overconsuming resources on a server, which in turn prevented other active applications from running effectively.

Furthermore, we didn't want any applications or OSs to bring down an entire multitiered application due to a software or hardware failure. Because it was impractical to "rightsize" the physical components for each physical server to match the load that was required of them by the services they were providing, this led to *server sprawl* (see Figure 1-2) – an overabundance of servers only using a small fraction of compute capacity that they were capable of.

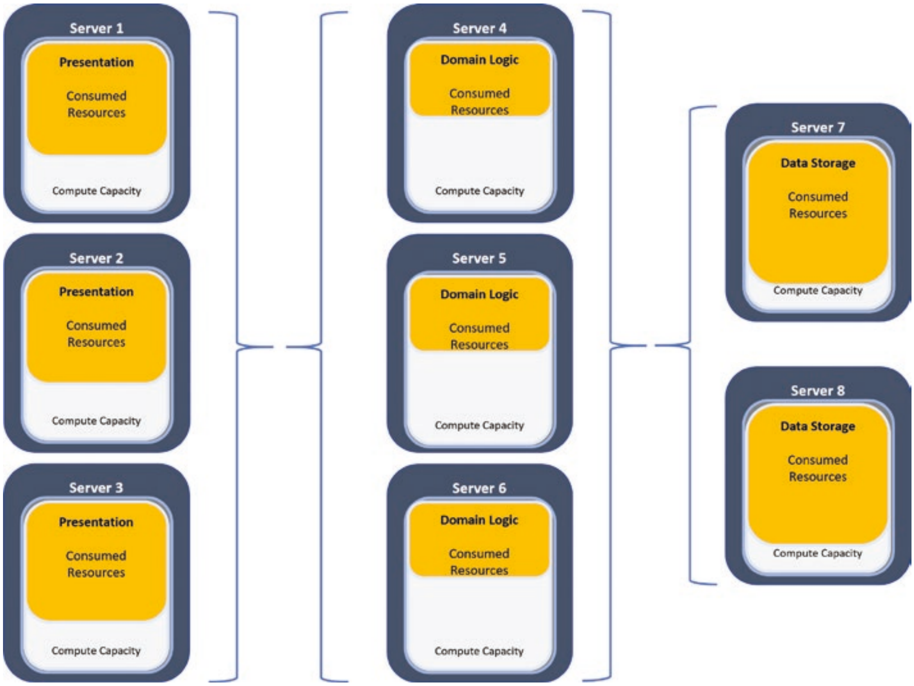


Figure 1-2. *Server Sprawl*

As the demand for computer servers exploded, it became apparent to even the most casual observer that the current datacenter was unsustainable; a new technology and/or datacenter methodology was needed, and fortunately, a new technology was coming on the scene at this exact time.

x86 operating system virtualization allowed multiple instances of an x86 operating system to run on a physical x86 server. These instances are now known as virtual machines, or VMs for short.

Introduction to Virtual Machines (VMs)

VM technology is not new; in fact, IBM supported the use of VMs back in the mid-1960s on their mainframe computers, and, later on, Unix systems supported something akin to VMs. However, these VM technologies were not widely used in the datacenter for a variety of reasons, including complexity, cost, and the fact that they were only viable on expensive proprietary hardware.

Type 2 Hypervisor

In the 2000s, a few companies devised hypervisor technology, a solution to run VMs on inexpensive commodity x86 servers. A hypervisor is the software or hardware that creates and runs VMs. In the early days of this revelation, we first had Type 2, or hosted, hypervisors that run on top of a base OS. Type 2 hypervisors can be basically thought of as just another application that is running VMs. In fact, you could run other applications, such as Microsoft Word, Excel, or even Solitaire, on the same base OS that is running the VMs (see Figure 1-3).

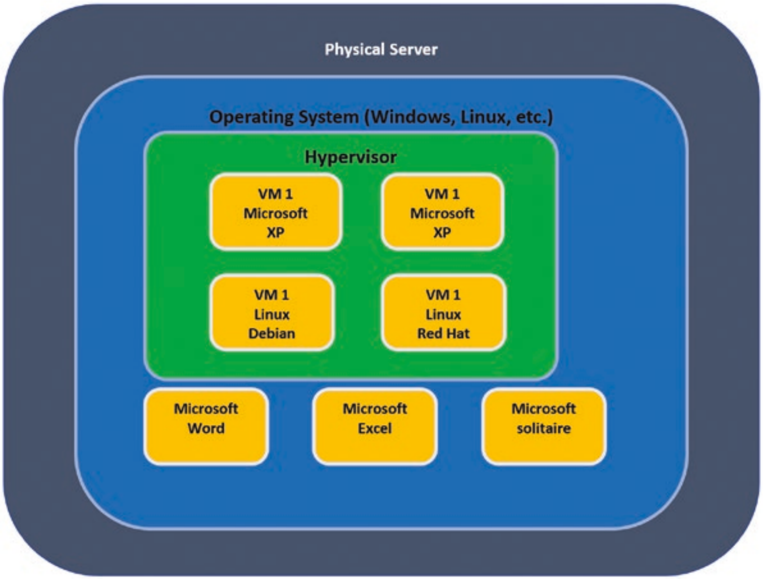


Figure 1-3. *Type 2 Hypervisor*

While Type 2 hypervisors were initially convenient to use, they posed various issues because of the fact that they run on top of an existing OS. These issues ranged from compatibility issues with the hypervisor when the base OS was changed or updated to the inability of the hypervisor provider to modify the base OS to optimize it for hypervisor use, rather than general application use.

Currently, the most popular Type 2 hypervisors, and the OSs that they run on top of, are VMware Workstation (MS Windows), VMware Fusion (macOS), KVM (Linux), Oracle VirtualBox (Linux, Windows, Solaris), and Parallels (Mac).

Type 1 Hypervisor

To overcome the limitations of the Type 2 hypervisors, companies created Type 1 hypervisors for x86 systems. Whereas a Type 2 hypervisor runs as an application on a base OS, a Type 1 hypervisor runs natively on the hardware, thereby supplanting the need for a base OS (see Figure 1-4). Type 1 hypervisors are also referred to as native, or bare metal, hypervisors as they run natively on the bare metal of a server.

The advent of Type 1 hypervisors radically changed the datacenter as it allowed multiple instances of OSs to run on a single physical server to safely, securely, and fully utilize the resources of the server. But server consolidation was just the beginning of the benefits that VMs and hypervisors brought to the datacenter. Later in this book, you will learn about some other benefits they offer.

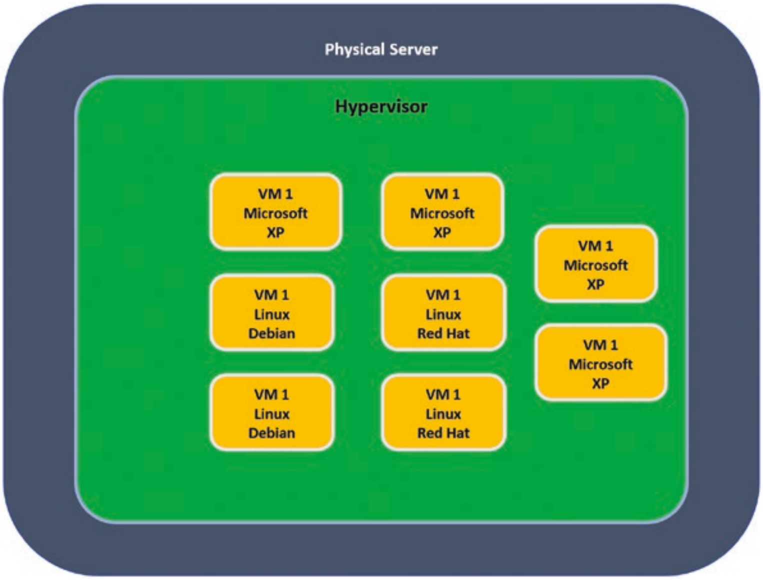


Figure 1-4. *Type 1 Hypervisor*

A hypervisor not only provides a place for VMs to run, it also provides the means to create and manage them. VMs can be created from scratch by installing an OS on them, or they can be cloned from an existing VM. The management functions that hypervisors provide vary greatly depending on the hypervisor, but they all provide compute resource management and security for the VMs that they manage.

A VM is composed of various configuration and data files that describe the VM and hold the data that makes up the VM. For example, a VMware VM is composed of a VM configuration file, BIOS or EFI configuration file, VM swap file, log file, data disk, as well as other files. Usually, each VM is kept in a separate folder.

The most popular Type 1 hypervisors are VMware ESXi, Microsoft Hyper-V, and XenServer.

VMware

VMware was founded in 1998 in Palo Alto, CA, by five individuals, including Mendel Rosenblum who was, at the time (and still is), a professor of computer science at Stanford University. Since their founding, VMware has become a leader of enterprise-level virtualization; and their Type 1 hypervisor, ESXi, is currently the most popular hypervisor in the enterprise.

VMware's first product was VMware Workstation, a Type 2 hypervisor launched in 1999 that ran on top of a Windows x86 system. While Workstation was designed to run on desktop systems, VMware also launched two products for the server market in 2001: GSX (a Type 2 hypervisor) and ESX (a Type 1 hypervisor). GSX quickly fell to the wayside as ESX became the *de facto* standard in enterprise datacenters. In 2004, ESX was renamed to ESXi, with the "i" standing for *integrated* (around this time VMware shrunk the code to ~32MB and moved many of the management functions of ESX to vCenter Server).

Throughout the years, VMware has continued to develop new technologies and acquire companies for their technologies. One of VMware's more notable acquisitions was Propero, a virtual desktop infrastructure (VDI) provider, in 2007; this acquisition later morphed into VMware Horizon, their very popular VDI solution. Also, in 2012, they acquired Nicira Inc., the creator of network virtualization products, who later became known as VMware NSX.

While VMware's acquisitions have primarily focused on companies within the virtualization space, they have also branched out and acquired technologies in other fields. As a prime example, in 2018, they acquired Heptio, a company deeply steeped into Kubernetes (K8s).

Of course, not all of VMware's current technologies and products were inherited from acquisitions; some of their great technologies have also been developed in-house. For instance, in 2014, they released vSAN, a software-defined storage (SDS) solution for VMware ESXi, which is currently one of the highest-selling SDS solutions in the datacenter.

VMware vSphere

When ESX was first released, it quickly became adopted by enterprise-class users. It also soon became apparent that its various instances running in the datacenter needed another program to oversee and manage them. To tackle this job, VMware created vSphere, a platform composed of not only ESXi but also other products and technologies to provide reliability, performance, business continuity (BC), and disaster recovery (DR) for a datacenter. Some of the products and features currently under the banner of vSphere include

- vCenter Server – A central management portal for controlling and managing ESXi hosts.
- VMware vSphere Client – An HTML5-based GUI for vCenter Server, replacing the vSphere Web Client that was Flash-based and before that the C# client.
- vSphere vMotion – A feature that allows a user to move a running VM from one ESXi host to another without incurring any downtime.
- vSphere Storage vMotion – A feature that allows a user to move a running VM from the storage on which it resides onto another storage location.
- VMware vSphere Distributed Switch (VDS) – A virtual network switch that can span multiple ESXi hosts.

- vSphere High Availability (HA) – A technology that monitors and restarts VMs if needed.
- Fault Tolerance (FT) – A technology that mirrors a VM across different physical servers. If one of the physical servers fails, the VM running on the other physical server will continue to operate unimpeded.
- VMware Distributed Resource Scheduler (DRS) – A feature that balances the compute resource usage of VMs between the ESXi hosts in a cluster.
- VMware Storage Distributed Resource Scheduler – A feature that balances the storage resource usage of VMs between the ESXi hosts in a cluster.

VMware is continuing to develop the above-listed features and technologies and to integrate new features into their products.

vCenter Server

While ESXi handles running VMs on a physical server, vCenter Server tells the ESXi hosts what to do and acts as a central administration point (see Figure 1-5). vCenter Server can be accessed in a few different ways: graphically via the vSphere Client; from a command line; or programmatically using APIs or PowerCLI, VMware's extension to PowerShell.

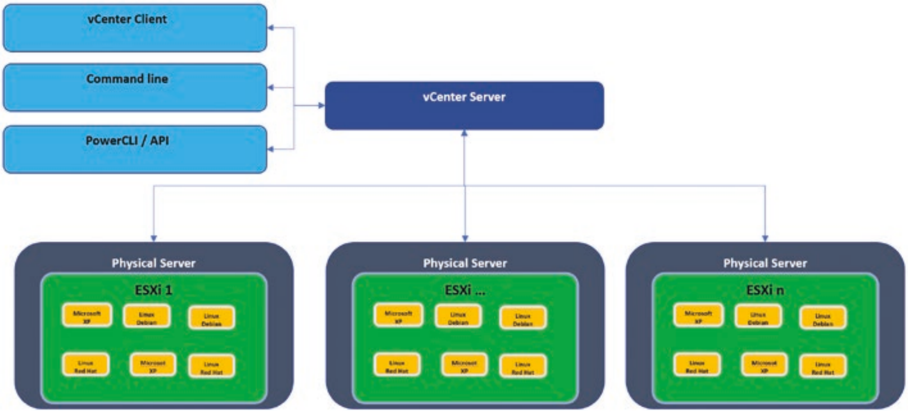


Figure 1-5. *vCenter Server*

As VMs are simply a collection of files, any ESXi server is capable of running vCenter Server as long as it has access to these files. In order to enable this functionality, VMware created a shared file system called VMware Virtual Machine File System (VMFS) and called the location on which a VMFS resides a *datastore*.

vCenter Server can be used to instantiate (start) a VM being stored on a datastore to any of the ESXi servers that are attached to it (Figure 1-6). Also, because the VMs are just software constructors, they can be configured or reconfigured as needed, for instance, if a VM needs more virtual CPU (vCPU) cores or more memory.

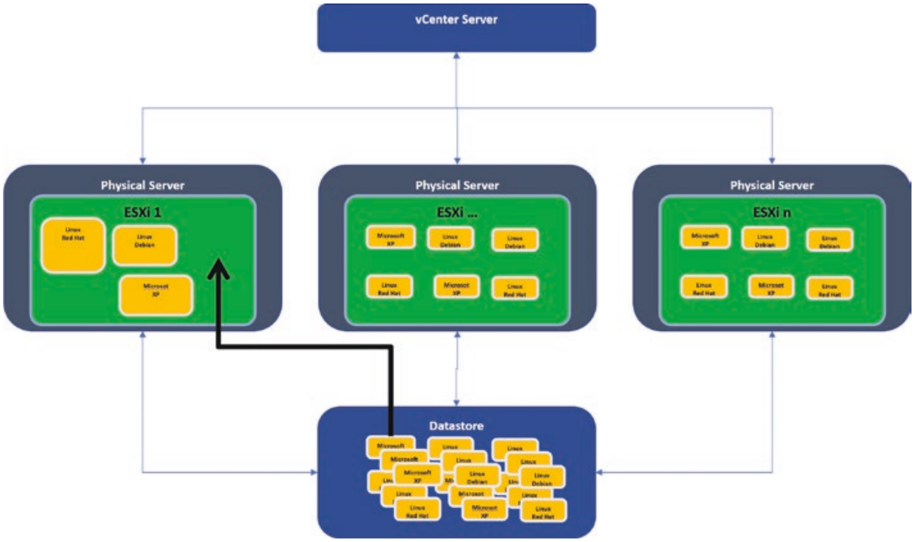


Figure 1-6. *Instantiating a VM*

vMotion

vCenter Server 1.0 was released in 2003 and enabled one of VMware's most defining features – vMotion, a tool that allows a running VM to be moved from one ESXi server to another (Figure 1-7). Although the vMotion action itself occurs between two ESXi servers, the integration of vCenter Server allowed for cross-server coordination. vMotion made it possible for the files that make up a VM to be stored on a datastore that multiple ESXi hosts could access, rather than on local storage that only a single ESXi host could access.

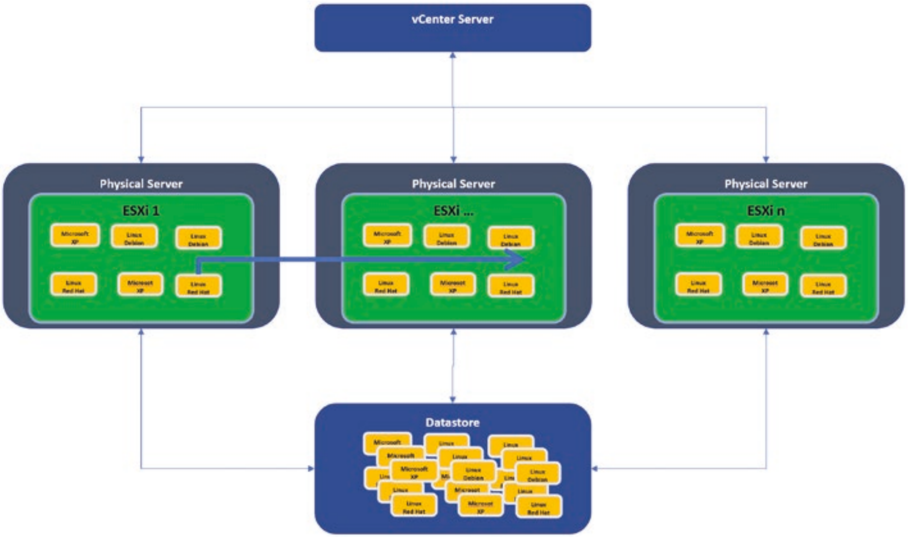


Figure 1-7. vMotion

The impact that vMotion had on VMware and the datacenter as a whole cannot be overstated; it allowed VMware to sell virtualization as a business continuity (BC) solution as well as for server consolidation.

From a BC standpoint, vMotion allows applications to be evacuated while still running from one host to another without disruption. This allows servers to be brought down for maintenance or replacement without affecting the day-to-day operation of a business. Datacenters can rebalance the VMs running on an ESXi host to make sure that their physical resources are being consumed uniformly. VMware even came up with a scheme, using vMotion, to consolidate VMs on a minimal number of servers and then power off servers during the night or at other times when they were unneeded.