

# Part 1: How We Got Here - Virtualization Solves Enterprise Challenges

**Guide to Virtualization  
on Portworx + Pure Storage**

**Part 1: How We Got Here**  
Virtualization Solves  
Enterprise Challenges



Welcome to the first post in our four-part blog series looking at what makes the combination of [Portworx®](#) and [Pure Storage](#) so awesome. If you're a virtualization or enterprise storage ninja and getting bombarded with requests or questions from a Kubernetes platform or application architecture team, this series is for you.

## Part 1: How Virtualization Solves Enterprise Challenges

First, we'll walk through all of the goodness that virtualization and enterprise storage has given us over the years and all of the operational excellence that you've been able to provide to your consumers in your organization. Then, in future posts in the series, we'll cover in detail:

- How containerization and Kubernetes came about and how that changed the way your application and developer teams consume infrastructure
- How Portworx provides the operational excellence that your platform consumers are looking for
- How enterprise features you've been accustomed to providing can be deployed for Kubernetes

and modern architectures

- How Portworx+Pure can be an absolute game-changer for you and your organization

Let's get started!

## 'Table for One, Please': Single Server, Single App

Let me give you a bit of background about myself. I started in IT back in 1997 as a sysadmin supporting 3D modelers and video production staff at a civil engineering firm. Before that, I was a process technician in multiple wafer fabrication facilities for almost three years, creating integrated circuits and spending 12-hour night shifts in a cleanroom wearing a full bunny suit and "making the sausage" from silicon.

My first "office" housed an HP Vectra and a 14" CRT monitor. It was here I would take Vectra desktops shipped to us from HP and "mod" them with components I'd buy at CompUSA: extra RAM, Matrox Millennium II graphics cards, Wacom drawing tablets, and 3Com Ethernet ISA NICs. Then, I'd load them up with 3D Studio Max and Adobe software so that the graphic artists and 3D modelers could create pre/post visualizations for civil engineering construction projects.

**This is where I got my introduction to "Enterprise" storage and servers.** I felt so lucky to be able to work with an ALR Q-SMP quad-Pentium 133MHz server with 128MB of RAM and almost 500GB of SCSI storage consisting of multiple trays of 9GB Seagate Barracuda disk drives. I also felt "lucky" to get to learn a different processor architecture and maintain a DEC AlphaServer 2100A, which had memory "boards" about the size of a pizza box.

I can't count the number of times I had to remove those boards and blow out the slots in that server with compressed air as the server wouldn't recognize all of its memory after a reboot! This was where I got my introduction to the amazing (ahem) operating system of Windows NT 3.51 and client/server networking. It was so "high-tech" compared to Novell NetWare and token-ring networks I had grown up on.

Fast-forward a couple of years and we got our first Data General Clariion array that ran Fibre Channel Arbitrated Loop (FCAL) and gave us RAID 5 capabilities. It was twice as fast as our old Ultra2/3 SCSI trays! This was where I was first introduced to Microsoft Cluster Server on NT 4.0 since we could have multi-server access to the LUNs on the Clariion and make our applications highly available in the case of an individual server failure. This was groundbreaking.

However, we still ran individual applications on single servers due to separation for security, performance of our applications, and "DLL hell" (library dependencies). This resulted in racks and racks of underutilized servers in our colocation space—until VMware released ESX in the early 2000s and **changed absolutely everything**.

## VMware Arrives and Flips Our World Upside Down

At first, nobody in management wanted to take the risk of taking applications running on bare metal and using software like PlateSpin and VMware Converter to convert operating systems to virtual machines. After all, there's no way that sharing x86 server and storage resources between multiple operating systems would provide the same performance that we were used to, right? So, we started migrating low-risk applications and dev environments onto VMs and kept QA and critical production applications like SQL Server, Exchange, and SharePoint on bare metal.

This was great, but there was the constant struggle between management and virtualization proponents on

**what we should virtualize and what we shouldn't.** Software vendors wouldn't support virtualized instances of their software, so you'd have to reproduce bugs on bare metal, and licensing schemes were still tied to bare metal infrastructure. Add to the mix that the "old school" server, network, and storage administrators viewed VMware as a "play toy" and the virtualization mountain became even harder to climb.

VMware noticed that only Tier 2 and 3 applications were really being virtualized and developed their Virtualizing Business Critical Applications (VBCA) program for their partners. This allowed them to prove to customers that VMware and ESX/ESXi was ready for prime time and Tier 1 critical applications in the enterprise. From this point forward, the innovation and product development from VMware took on a life of its own and gave birth to many of the enterprise features that you as a virtualization or enterprise storage admin have been giving your consumers ever since.

## **How Virtualization Changed the Experience for Admins, Developers, and Users Everywhere**

### **Application Availability**

[VMware HA](#) was a true game-changer for application availability in the enterprise. The capability to quickly restart a VM on another server running ESXi gave us way better RPO than we'd ever had before. The ability to create startup sequencing plans and have multi-tier applications start up properly in the case of a server failure increased app availability to levels never seen before in bare metal environments.

### **Resource Utilization**

[VMware DRS](#) was *the* feature that allowed us to properly utilize server resources—in many cases going from single-digit CPU/memory utilization to 70%-80% per physical server. DRS gave us the capability to properly balance VM resource utilization across large, multi-server clusters and get the most value out of every single server purchase in the enterprise. This drove down costs associated with rackspace footprints, power consumption, network and SAN capacity, and server maintenance.

### **Cost Management and Staff Skills**

Again, reducing cost of infrastructure was a huge benefit due to resource utilization and preventing downtime costs by increased application availability. On the staffing side, VMware gave admins, who had historically been relegated to a single stack of the infrastructure (compute, network, storage), the opportunity to broaden their skill sets and turn into well-rounded infrastructure engineers.

In my view, this is really where [platform engineering](#) was born, giving life to the virtualization skill sets and toolboxes that we have today. While dedicated infrastructure staff was necessary for larger enterprises and environments, the creation of the VM admin persona gave many organizations the opportunity to reduce staff spend, reducing cost even further and providing VM admins the capability to deepen and broaden their skills as they grew their careers.

### **Data and Application Portability**

Abstraction from the bare metal layer using VMware gave us some phenomenal capabilities around moving applications and their data between local clusters and geographically disparate data centers or office

locations. [vMotion](#) and Storage vMotion changed the game for migration between two connected VMware clusters within our organizations.

In addition, instead of having to build new server instances and restore data from tape or a portable hard disk, we could now export a VM with all of its necessary data to an OVF and have it up and running at a new disconnected location—simply by importing the OVF to a new ESXi cluster. This was also a huge benefit for replicating environments between dev, QA, and staging environments for quick standup of multi-tier application environments, which reduced infrastructure spin-up time and maximized development workflows.

## Capacity Management

Besides the obvious benefits of sharing CPU and memory resources on physical servers, the concept of shared storage and datastores gave us the capability to maximize investments of our underlying enterprise storage infrastructure. Instead of adding to the capacity monsters in our SAN environments and allocating multiple LUNs that might only be 10% utilized, we could right-size our storage allocations and have them grow as needed.

Being able to dynamically expand and grow VMFS datastores with no impact to VMs running on them allowed us to ensure that we were handling the growth of data proactively instead of reactively, providing further benefits to application availability and cost management.

## Storage Infrastructure

I moved forward in my career, from sysadmin to a consultant, where I provided VCE Vblock solutions and converged solutions using EMC storage and Cisco UCS compute. Then, I jumped to the vendor space and spent almost 10 years at Hitachi. This is where I saw so much innovation personally.

With VMware providing [VASA and VAAI integrations](#) for things like path selection and failover, surfacing storage capabilities from array vendors via [Storage Policy Based Management](#) (SPBM), to [vVols](#) and the development of [vSAN](#) so customers could use commodity hardware, the game changed for historic storage admin personas. Instead of having to rely solely on storage admins to provide enterprise features that lived within our arrays, we finally had a method to surface these capabilities to VMware admin personas. It really reduced the friction between VMware and storage admins.

## Developer Agility

How many hours/days/weeks did it take us to deploy a single server for a developer in the bare metal days? Between resource planning for rackspace, power, network, and storage to service tickets and change advisory board approval, to acquisition cost and time to receive equipment, developers could be waiting on infrastructure for way longer than needed.

The ability to create VM templates and provide self-service deployment of VMs so they could be more agile in their development process was yet again a game-changer, thanks to the abstraction layer of VMware. However, this brought additional challenges of VM sprawl. It led to overallocation of resources at times so you had to be careful of how well you enabled your developers and end users! Was this the infancy of [DevOps](#) practices as we know them today?

## Disaster Recovery

Our capabilities to provide low RPOs and RTOs to our consumers in virtualized environments was limited and complex until VMware announced [Site Recovery Manager](#) (SRM). This gave enterprise array vendors the capability to create Storage Replication Adapters (SRA) for their specific arrays, tying replication functionality in the microcode of their arrays into workflows inside SRM. We could finally have asynchronous replication of not only the VMDKs our VMs were using but also all of the VM objects within vCenter and our ESXi servers. VMware also gave us the capability to execute failover blueprints for multi-VM applications to ensure that our applications came up on the remote site properly before a disaster struck.

Shortly after SRM, VMware introduced a synchronous solution called [VMware Metro Storage Cluster](#) (VMSC). This allowed enterprise array vendors who had synchronous replication abilities to create zero-RPO disaster recovery solutions for customers, further extending business continuity capabilities for virtualized environments. I still remember poring over Duncan Epping's Yellow Bricks and Eric Shank's TheITHollow blogs to understand these technologies and designing VMSC solutions at Hitachi to enable our customers with this groundbreaking technology to meet SLAs that were previously unthinkable!

## Security, Encryption, and RBAC

Obviously, a major enterprise feature that was needed in VMware environments was security. From integration with industry-leading security solutions from vendors such as Cisco and Palo Alto (remember the Nexus 1000V, anyone?) and VM and VMDK-level encryption, to comprehensive RBAC roles within vCenter, VMware again gave us what we needed to secure our VMware infrastructure. Since we were combining so many different tiers within our application architectures on single servers, abstraction of the security layers was crucial and especially important from a regulatory perspective in specific verticals.

## Data Protection

Anyone who was privy to the early days of virtualization remembers the challenges we had with data protection. We used to load up third-party backup clients onto each operating system that we needed to protect, which would interface with a central backup server that would stream our data to tape. Then came the days of VMware integration by the backup vendors, who provided the ability to query VMware infrastructure and actually back up the entire VM, then allow restoration to any other VMware cluster.

This brought its own set of challenges until the arrival of disk-based backup, snapshots, and Changed Block Tracking (CBT) with deduplication. We no longer had to worry about tape libraries, off-site tape storage in bank vaults or Iron Mountain, and network bottlenecks to the backup server. If you're anything like me, you remember being able to get NFR licenses as a VCP for software like Veeam, which you could use in your home labs to protect your data. And, at the same time, you learned how to properly protect your organization's data and VMs on your virtualized infrastructure!

## Performance

Resource reservations for CPU and memory combined with thick provisioned storage and non-abstracted storage such as vVols gave us the capability to provide near bare metal performance for our virtual machines. As processor core densities increased along with network and storage throughput, we were finally able to guarantee performance requirements for our applications and the virtual machines that hosted them. Now that enterprise storage solutions such as Pure Storage [FlashArray](#)<sup>™</sup> and [FlashBlade](#)<sup>®</sup>

are bringing groundbreaking performance capabilities, customers can be guaranteed that the highest performance levels possible can be provided to their applications.

## Consolidating Even Further: The aaS-ification of All the Things

Amazon began spooling up public cloud infrastructure as early as 2006 while many organizations were trying to take advantage of private cloud infrastructures they were building using VMware and their own hardware. Many organizations were wary of the cost and perceived risk associated with handing over the infrastructure keys to a public cloud and wanted to take advantage of the investments they had made into their own infrastructure.

Providing self-service capabilities to consumers within their own organizations to consolidate infrastructure services even more was the talk of the town, and VMware provided innovation yet again by releasing products such as [VMware Cloud Director](#) (VCD) alongside other solutions such as [OpenStack](#). With that, [infrastructure as a service \(IaaS\)](#) was born.

Once organizations began consuming IaaS, much of the industry followed with platform as a service (PaaS) and software as a service (SaaS) leading the charge. Soon thereafter, everything was being offered as a service: DR, data protection, desktops, storage, and network. Even monitoring was being offered as a service! This dramatically changed spend and resource allocation as organizations focused more on consuming services as opposed to the infrastructure- and expertise-heavy approach that virtualization brought us.

Thus began the tipping point to next-generation application development and consumption, toward a more developer- and app-centric approach. No longer was infrastructure the king, driving app development from the bottom up. Developers became the king and infrastructure had to follow!

## What's Next?

In the [second blog post](#) in our series, we'll cover how containerization and [Kubernetes](#) came about and how that changed the way application and developer teams consume infrastructure. I hope you've enjoyed our little history lesson so far—and please keep the concepts and details we covered here in mind as we explore Kubernetes, modern application development concepts, and how they affect your ability to provide enterprise services!

