



# VMware App Volumes Reference Architecture

VMware App Volumes 2.10

TECHNICAL WHITE PAPER

**Table of Contents**

- Audience ..... 4
- Executive Summary ..... 4
- App Volumes Overview ..... 5
  - App Volumes Benefits .....5
  - How App Volumes Works .....6
  - AppStacks .....6
  - Writable Volumes .....7
  - Other Components .....7
- Test Results ..... 8**
  - Topics of Exploration .....8
  - Test Approach .....8
  - User-Experience Performance Results .....9
  - Host Performance Results ..... 14
  - Desktop Performance Results ..... 16
  - Infrastructure Server Performance Results ..... 17
  - Database Performance Results ..... 20
  - Storage Performance and Capacity Results ..... 21
- Testing Details ..... 23**
  - Methodology ..... 23
  - Key Metrics ..... 25
  - Test Tools ..... 26
- Conclusions ..... 31
- About the Authors ..... 32
- Acknowledgements ..... 32
- Appendix A: Environment Design ..... 33
  - vSphere Design ..... 33
  - View Design ..... 35
  - View Connection Servers ..... 35
  - App Volumes Design ..... 36
  - Storage Configuration ..... 38
  - Network Configuration ..... 42
  - Desktop Master Images ..... 42

- Appendix B: Test Environment Settings ..... 44
  - Infrastructure Server Virtual Machine Configurations..... 44
  - View in Horizon 6 Server Configurations and Settings..... 45
  - Desktop Pool Configurations..... 45
  - Desktop Image Configuration ..... 46
  - Login VSI Test Parameters ..... 47
- Appendix C: Bill of Materials ..... 48
  - Hardware BOM ..... 48
  - Software BOM..... 48
- Appendix D: Additional Resources ..... 49
  - VMware..... 49
  - Login VSI ..... 49
  - EMC XtremIO..... 49
  - Blogs..... 50
  - VMware Hands-On Labs ..... 50

# Audience

This document is intended for IT architects and administrators who want to understand the performance and scale attributes of VMware App Volumes™ in a virtualized desktop environment. The reader should have a solid understanding of desktop and application virtualization, familiarity with VMware Horizon®, especially the View feature, and VMware vSphere® products, in addition to an understanding of sizing and performance concepts.

# Executive Summary

We carried out extensive testing to evaluate the performance and capacity characteristics of VMware App Volumes in a View desktop environment. This paper describes a simple, validated architecture and details of the test results, which are summarized in the following infographic:

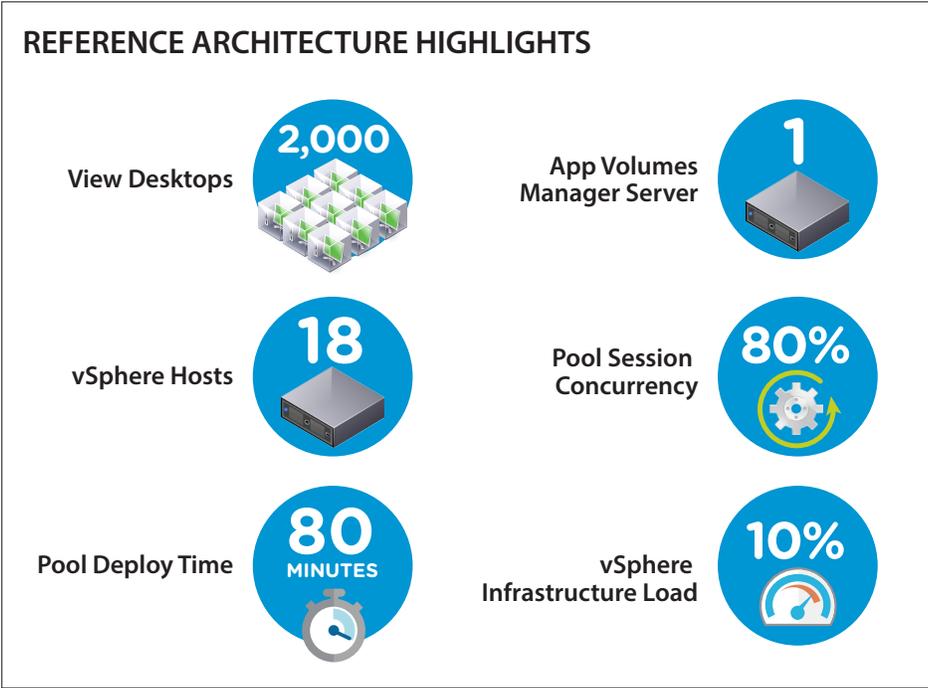


Figure 1: Reference Architecture Highlights

## App Volumes Overview

VMware App Volumes is a transformative solution that delivers applications to View virtual desktops. Applications installed on multi-user AppStacks or user-specific writable volumes attach instantly to a desktop at user login. The App Volumes user experience closely resembles that of applications natively installed on the desktop.

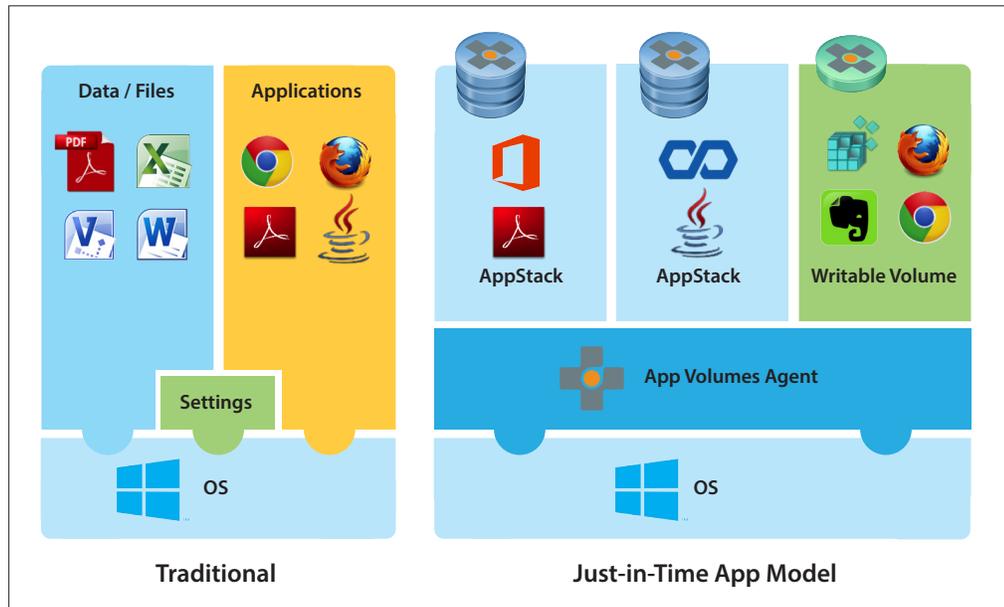


Figure 2: App Volumes Overview

App Volumes complements the VMware End-User Computing portfolio by integrating with other VMware application and desktop solutions.

For further details, see the latest [App Volumes documentation](#) and the [App Volumes Deployment Guide](#).

### App Volumes Benefits

With App Volumes, applications become VM-independent objects that can be moved easily across data centers or to the cloud and shared with thousands of virtual machines. In a virtual desktop environment, App Volumes provides the following benefits:

#### Real-Time Application Delivery

- Delivers and upgrades applications at scale and in seconds
- Dynamically delivers applications without interrupting users even if they are logged in

#### Cost-Optimized Infrastructure

- Optimizes application delivery to drive down compute, network, and storage costs
- Can reduce storage costs for VDI
- Works with existing infrastructure with flexible delivery to users, groups, or devices

**Seamless End-User Experience**

- Supports fully customizable desktops, with the freedom for end users to install their own applications
- Creates a persistent user experience with nonpersistent economics

For a detailed description of the technical and non-technical benefits that App Volumes brings to organizations, users, and administrators, see the [App Volumes Deployment Guide](#).

**How App Volumes Works**

App Volumes integrates a simple agent-server-database architecture into an existing View deployment. Centralized management servers are configured to connect to deployed virtual desktops that run an App Volumes Agent. An administrator can grant application access to shared storage volumes for users or virtual machines or both.

This figure shows the major components of a View environment where App Volumes is deployed.

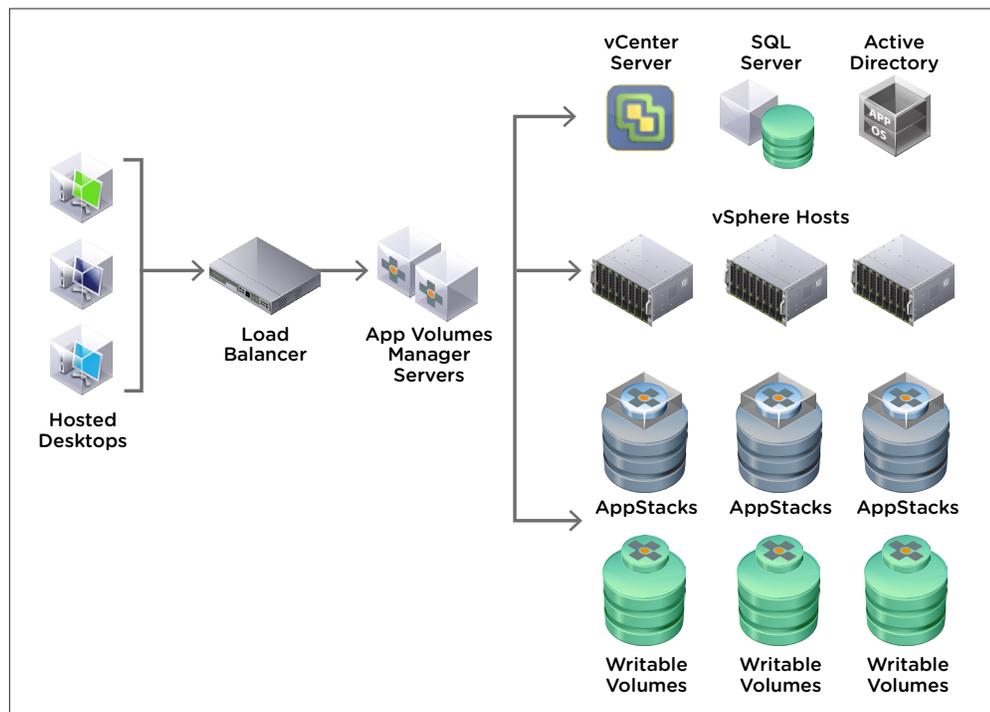


Figure 3: App Volumes High-Level Architecture

**AppStacks**

AppStacks are read-only volumes containing applications that can be assigned to Active Directory (AD) user accounts, groups or organizational units (OUs), or computer accounts to enable delivery of applications to end users.

Administrators can combine core applications into a single AppStack, making the AppStack easy to assign to users through AD object assignment. Administrators can make application updates available immediately or on next login or reboot.

Administrators can also manage line-of-business or departmental apps by combining them in an AppStack, which is managed and deployed separately from the core applications.

## Writable Volumes

The writable volumes feature is an optional component that enables a per-user volume where the following user-centric data can be installed and configured in different ways and can move with the user:

- Application settings
- Licensing information
- Configuration files
- User-installed apps

The writable volumes feature does not provide a user environment management (UEM) solution. Writable volumes complement UEM solutions, which can manage data within writable volumes at a more granular level and provide contextual rules to enforce policy based on different conditions or events.

It is important to remember the differences between AppStacks and writable volumes.

- AppStack virtual machine disks (VMDKs) are mounted as read-only and shared among all virtual machines (VMs) within the data center.
- Writable volumes are dedicated to individual users and are mounted as the user authenticates to the desktop. Writable volumes are user-centric and can be assigned to specific computers and then reassigned to other computers. Writable volumes cannot be shared or reassigned to multiple users.

## Other Components

The other components of App Volumes are briefly described in the following table.

COMPONENT	DESCRIPTION
App Volumes Manager Server	A Windows Server system used as the Web Console for administration and configuration of App Volumes and assignment of AppStacks and writable volumes. App Volumes Manager is also used as a broker for the App Volumes Agents, for automated assignment of applications and writable volumes during desktop startup or user login.
App Volumes Database	A Microsoft SQL database that contains configuration information for AppStacks, writable volumes, users, machines, entitlements, and transactions.
App Volumes Agent	Software installed on all Windows desktops where users receive AppStack volumes and writable volume assignment. The agent runs as a service and uses a filter driver to handle application calls and file system redirects to AppStack and writable volume VMDKs. Windows desktops do not have to be members of the domain where the App Volumes Manager server resides.
Provisioning Virtual Machine	The provisioning virtual machine is a clean desktop virtual machine that includes the OS, necessary updates and service packs, and has only core applications installed. The provisioning virtual machine must have the App Volumes Agent installed and configured to connect to the App Volumes Manager.
vCenter	VMware vCenter Server™ provides centralized management of vSphere virtual infrastructures. App Volumes leverages vCenter for inventory information and operational connectivity to host, virtual machine, and storage resources within a deployed vSphere environment.

**Table 1:** App Volumes Components

## Test Results

While establishing the framework for this project, we developed the following set of test hypotheses, based on conversations with the product team, customers, partners, and integration experts, to address most of the important production considerations for App Volumes:

- Applications delivered by App Volumes provide a user experience similar to native applications, with only modest increases in vSphere and infrastructure host load.
- App Volumes introduces new storage performance characteristics and new capacity consumption patterns.
- Adjusting the number of apps per AppStack can affect performance.
- App Volumes shows linear performance scaling from 0 to 2,000 active users.

### Topics of Exploration

During the project, as we tested and built up a large library of test data, we also explored the following questions as they apply to performance and scalability in production environments:

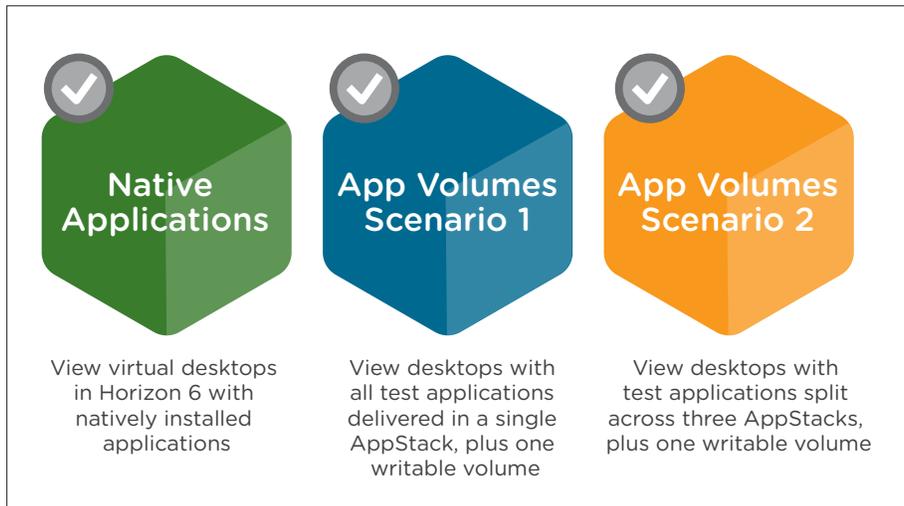
- How do native applications compare to applications delivered by AppStacks with regard to user-experience metrics?
- Do vSphere hosts show the same resource loading with natively installed applications as they do with applications in App Volumes containers?
- How do storage capacity consumption and I/O compare between native application installations and applications delivered by App Volumes?
- What vSphere, App Volumes Manager, vCenter, and SQL resources are required for 2,000 active users?

### Test Approach

Using the Login VSI Knowledge Worker workload with a 64-bit Windows 7 image, 2 vCPU, and 2 GB of RAM, we loaded as many active user sessions as we could on a host until the host CPUs reached an average of 80 percent usage. We then discovered the ideal number of desktops per host with this workload is about 88—a little less than 9 vCPUs per host processor core.

Our test scenarios were all run with a 2,000-desktop linked-clone pool running at 80 percent session concurrency. This means that during all Login VSI tests, 1,600 desktops (80 percent of the pool) had a user logged in doing work. During a full Login VSI test run, we logged in a new user every 4 seconds until we reached 80 percent concurrent session load, and then let the workload run for about 4 hours. During the test run, we used VMware vRealize® Operations for Horizon, VMware vRealize Log Insight™, vCenter, EMC XtremIO Storage Reporting, and Windows Performance Monitor to record detailed performance statistics. Login VSI also records an impressive number of user-experience metrics that are used to produce a score called VSImax. This score represents the largest number of user sessions you can run in a particular environment with good user experience.

We ran each of the following test scenarios at least four times, with desktops pools refreshed after each test.



**Figure 4:** Test Scenarios

**Important:** We exercised every test configuration with full-session concurrency, to ensure that View, vCenter, and App Volumes could handle the simultaneous user load of 2,000 active users with a single View Connection Server, App Volumes Manager server, and desktop vCenter server. Throughout testing, we had no problems with 2,000 users, each with three AppStacks and a writable volume attached. Testing was conducted at 80 percent session concurrency because, when all 2,000 desktops are in use, the vSphere host CPUs are running at maximum load. *This is never a safe practice in a production View environment.*

For full details on the tests, see [Appendix A: Environment Design](#).

## User-Experience Performance Results

User login times are a key measure of desktop performance. Users who have to wait for long periods of time for their desktops to be ready are generally not satisfied with the VDI experience. During each test scenario, we measured the time for each of the 1,600 active users to log in to their desktop. This included the time to mount AppStacks, as applicable. For more information on the definition of login times and how we obtained this information, see [Testing Details](#).

During testing we observed an increase in login time with both App Volumes scenarios. The additional time it takes to mount AppStacks and writable volumes on the back end means longer login times for users on the front end, and the more AppStacks assigned to users, the longer the login times. This is a key consideration for making decisions about the number of applications in a single AppStack, or AppStack density.

**Note:** During each test scenario, we used View event logs, which were fed to vRealize Log Insight, to measure the time between the initial user authentication request on the View Connection Server and the time that the user's View session was considered to be in **Connected** status.

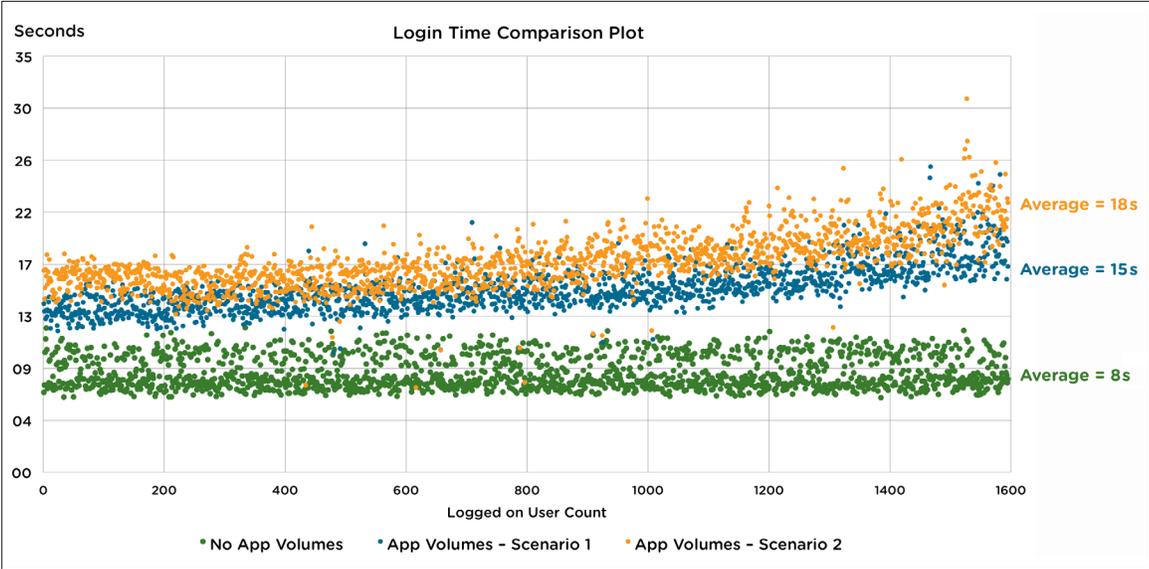


Figure 5: Average User Login Times

Login VSI is a valuable VDI user experience benchmark tool. It allowed us to adjust the active user session count and measure the corresponding impact on user experience. During our testing, we increased the session count until we reached the point where user experience degraded below acceptable levels. This VSImax score is the user count where the benefits of increasing the number of active sessions has a real impact on user experience. If you run more sessions than this value, users will not enjoy the desktop experience. For more information on the definition of VSImax, see [Testing Details](#).



Figure 6: VSI max Score Comparison

**Note:** Testing was performed with synthetic user workloads. Real-world users exercise applications and access data in a more chaotic, random manner. Also, most organizations deploy more than a single use case in a View environment. Real-world consumption patterns vary from organization to organization. Before deploying any desktop workspace technology it is important to understand the use-case resource requirements. Reference architecture workloads based on lab testing may not precisely match real-world user workloads.

During testing, we observed that the VSImax scores were lower after the introduction of App Volumes. This means that the maximum number of supported desktop sessions was also slightly lower.

Fewer active desktop sessions can be supported in a given View environment when users are assigned AppStacks and writable volumes. When we increased the number of AppStacks, we saw a marginal decrease in the VSImax score.

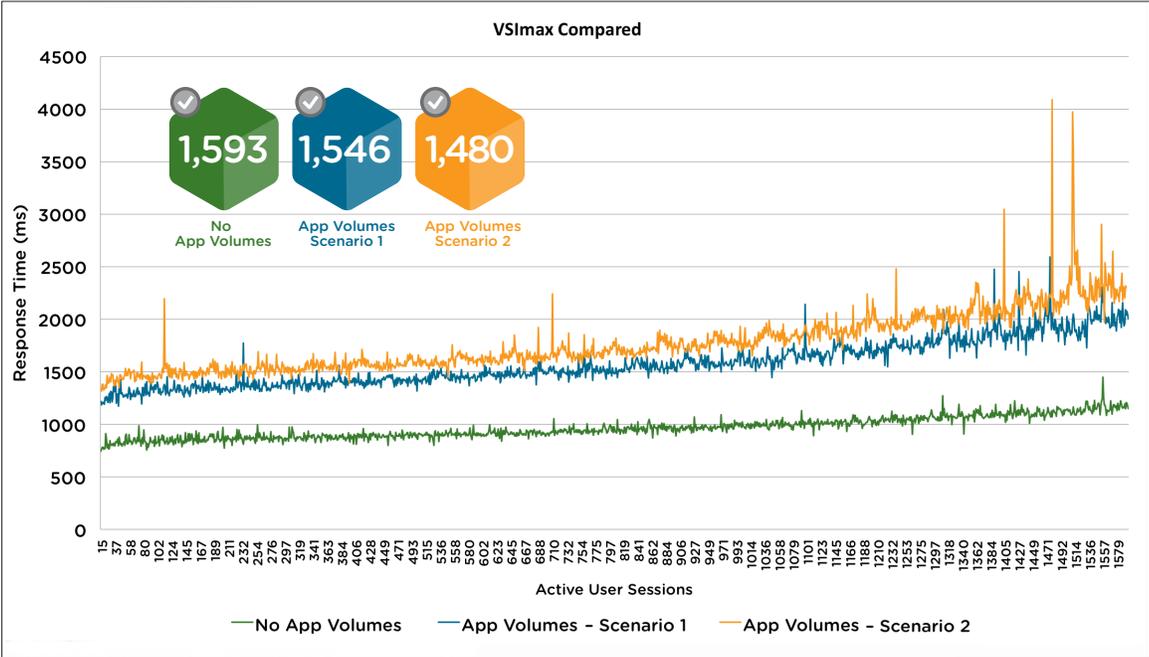


Figure 7: VSImax Based on Active Session Count

During Login VSI testing, we also measured the approximate launch time of the desktop applications that are part of the Login VSI test workload. The opinions of many users about their desktop experience are based on how long it takes for their applications to load. During each test scenario, we documented this important metric for each application. For more information on the Login VSI application launch times, see [Methodology](#).

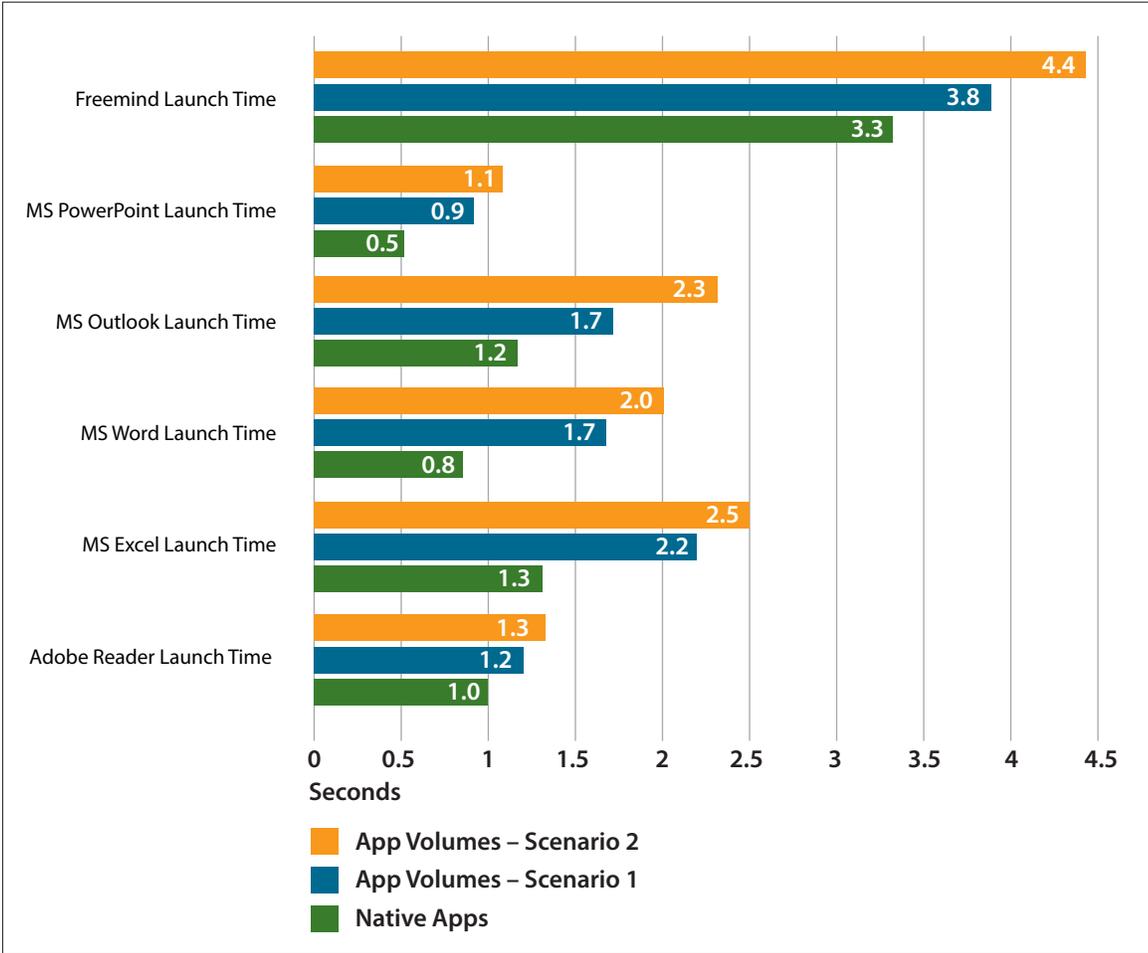


Figure 8: Application Launch Times

Applications delivered with AppStacks showed consistently longer launch times than natively installed applications. In some cases, however, the launch times were similar enough that users may not notice a difference.

Virtual disk latency has enormous impact on user experience. During testing we measured the aggregate virtual disk (vDisk) read and write latency for the desktops undergoing testing. These peak average values were measured with vRealize Operations.

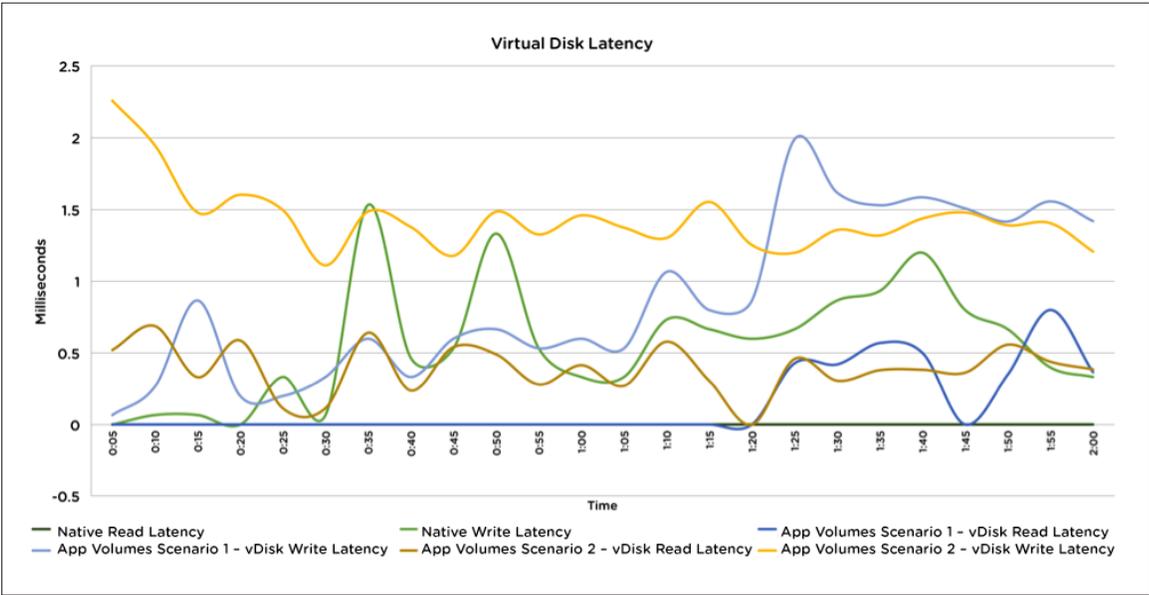


Figure 9: vDisk Latency Comparison

Desktop vDisk latency was very low during all test scenarios. Latency was slightly higher during App Volumes tests, but average read and write latency never exceeded 5 ms, which is considered the ideal maximum for vDisk latency in a View environment. It is important to note that during these tests, an all-flash shared storage array with a fast storage network was used. For more details on the storage environment, see [Storage Configuration](#). By using high-performance XtremIO storage, we eliminated most storage-related performance bottlenecks that could have negatively impacted user experience.

**Note:** Because we utilized EMC XtremIO all-flash storage and a very fast storage network, during testing vRealize Operations often did not record any significant virtual disk latency. All-flash storage often delivers read and write I/O performance that must be measured in microseconds. VMware vRealize Operations measurements are made in milliseconds.

### Host Performance Results

It is of vital importance for organizations to monitor vSphere host resource utilization to ensure that hosts are not loaded to the point where resource contention develops. Host resource contention can sometimes be tolerated in server environments, but in hosted desktop environments resource contention can result in bad user experience.

Throughout testing, hosts were carefully monitored, and comparative resource consumption metrics are presented in the following graphs.

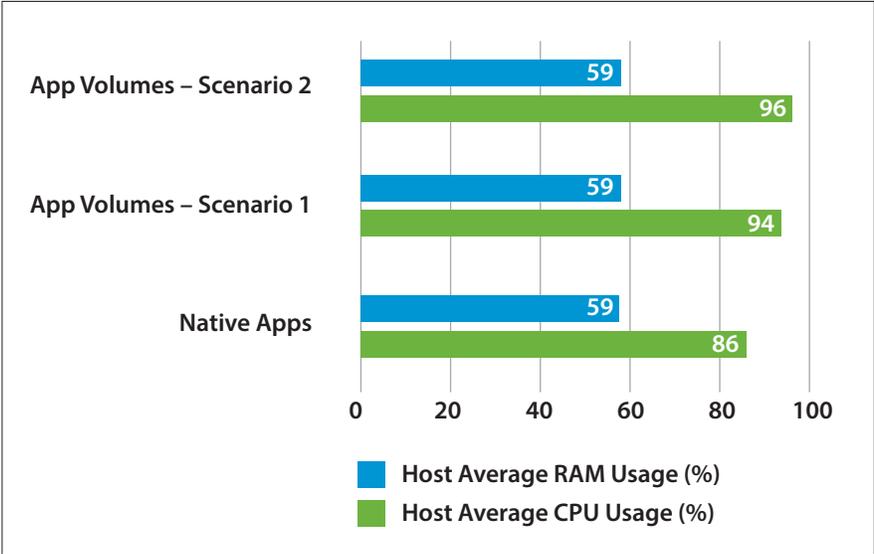


Figure 10: Host CPU and RAM Usage

Host average memory usage was the same for all three test scenarios. However, during the App Volumes tests, the average CPU usage averaged about 10 percent higher than with natively installed applications. This should be taken into consideration when App Volumes are added to an existing View environment. This extra host processor load should be accounted for when host and cluster maximum session capacity are determined.

**Note:** A common vSphere best practice is to never allow host CPU average usage to exceed 80 percent for production environments. During testing, we pushed our host CPU load beyond this “safe” production threshold. Exercise caution when running hosts with more than 80 percent resources consumed.

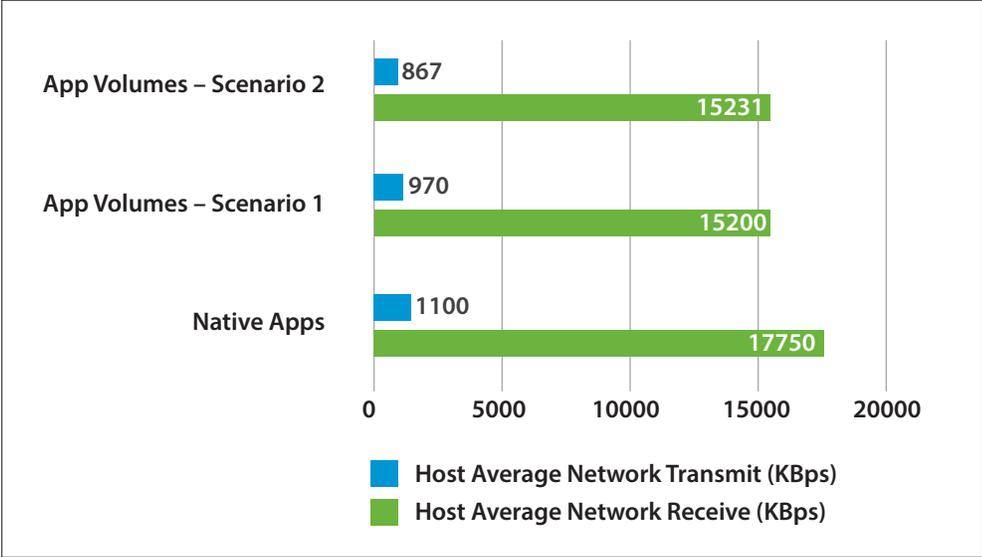


Figure 11: Host Network Usage

There was a very slight decline in network transmit and receive traffic in both App Volumes test scenarios. This was not considered to be statistically significant. Host network utilization is roughly the same in native and App Volumes deployments.

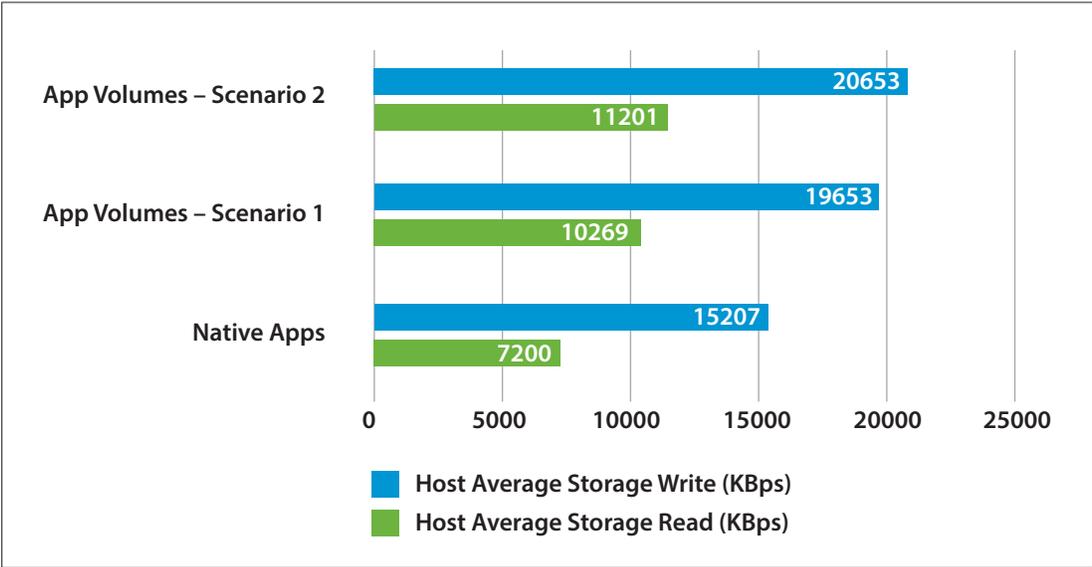


Figure 12: Host Average Storage Rates

Hosts recorded much higher shared storage utilization rates (both reads and writes) in the App Volumes scenarios, and this is to be expected with the App Volumes application delivery method. This increase can be seen clearly in Figure 12. And in the App Volumes test scenarios, more AppStacks (Scenario 2) meant higher overall shared storage utilization rates than in single AppStack tests (Scenario 1).

### Desktop Performance Results

Desktop performance was very good throughout all test phases and scenarios.

Desktop Performance Metric	Native Applications	App Volumes Scenario 1	App Volumes Scenario 2
Desktop CPU Usage %	12.5	14.1	15.2
Desktop RAM Usage %	60.2	60.3	61.7
Desktop Disk Read Rate (KBps)	98.6	340.2	203
Desktop Disk Write Rate (KBps)	189.2	258.2	260.5
Virtual CPU per Host Core (at VSImax)	8.9	7.6	7.4

Figure 13: Desktop Average Resource Usage

Desktop processor and memory usage were the same in all three test scenarios. Desktop network usage (not shown) was the same for all test scenarios. A much higher volume of disk activity, both reads and writes, was observed during both App Volumes test scenarios. Virtual CPU-to-host density at VSImax (when environment VSImax was reached) shows that in all three scenarios, seven virtual CPUs can operate on each host processor core.

App Volumes adds a noticeable amount of performance overhead on each virtual desktop. The slightly higher desktop CPU resource usage is reflected in the slightly higher CPU usage that was observed at the vSphere host measurement point. The major increase in storage I/O should also be taken into consideration when a storage platform for View desktops and App Volumes is chosen.

**Note:** For details on desktop CPU configuration and host processor information, see [Appendix A: Environment Design](#).

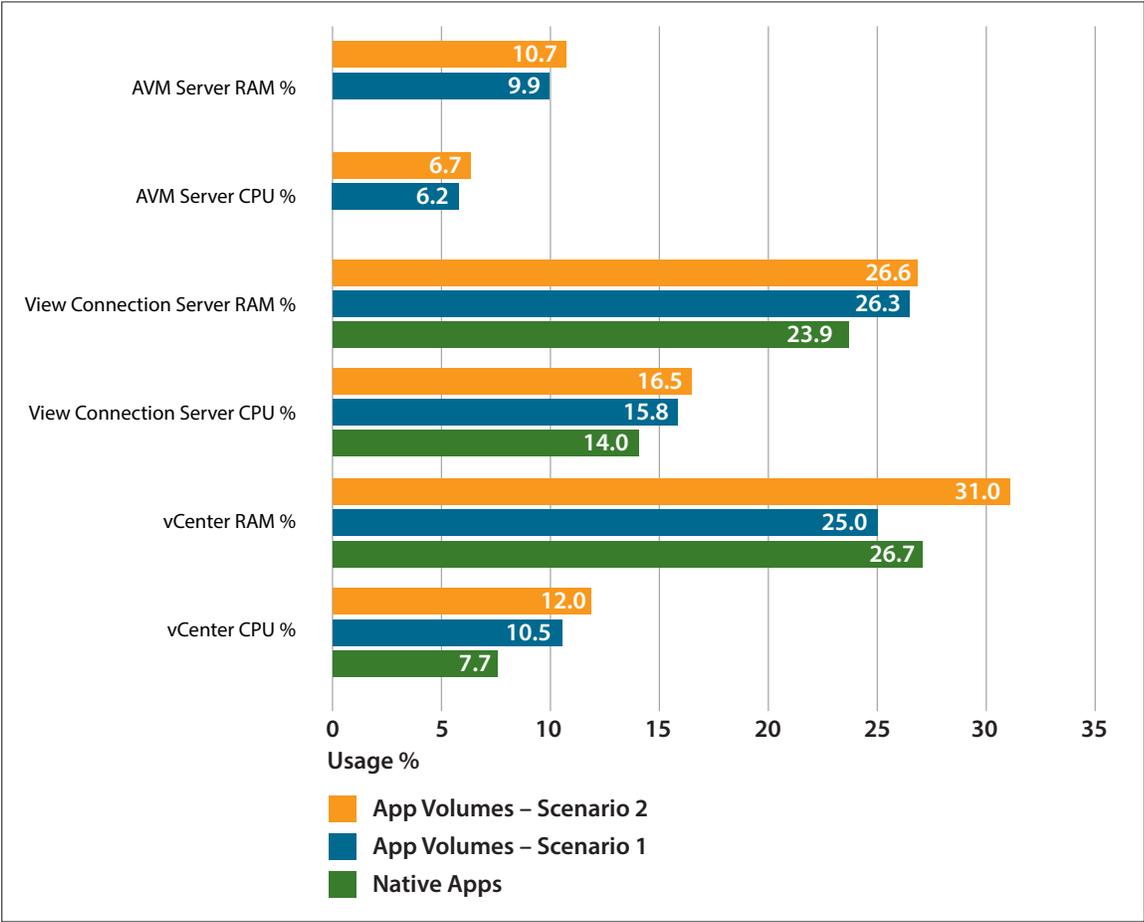
### Infrastructure Server Performance Results

All infrastructure servers were sized according to the requirements and specifications in the product documentation. Infrastructure server specifications are detailed in [Appendix B: Test Environment Settings](#).

SERVER CONFIGURATION	vCENTER SERVER	VIEW CONNECTION SERVER	APP VOLUMES MANAGER SERVER
Operating system	OVA Appliance SUSE Linux 11	Microsoft Windows Server 2012	Microsoft Windows Server 2012
Virtual machine CPU	8	4	4
Virtual machine RAM	24 GB	10 GB	8 GB
Storage provisioned	320 GB	82 GB	80 GB

**Table 2:** Infrastructure Server Sizing and Configuration

Throughout testing all vSphere, View, and App Volumes infrastructure servers were carefully monitored with vRealize Operations. Comparative resource usage metrics are highlighted in the following figures.



**Figure 14:** Infrastructure Server CPU and RAM Average Usage

During all testing, infrastructure resource utilization patterns were very similar, although it should be noted that vCenter CPU and memory resources were more heavily utilized during tests that included App Volumes. This is likely due to the number of virtual disk operations that are required on each desktop virtual machine. It must be stated here that App Volumes was configured to use **Mount on Host** for all storage operations. Storage commands were sent directly to vSphere hosts rather than being processed by vCenter.

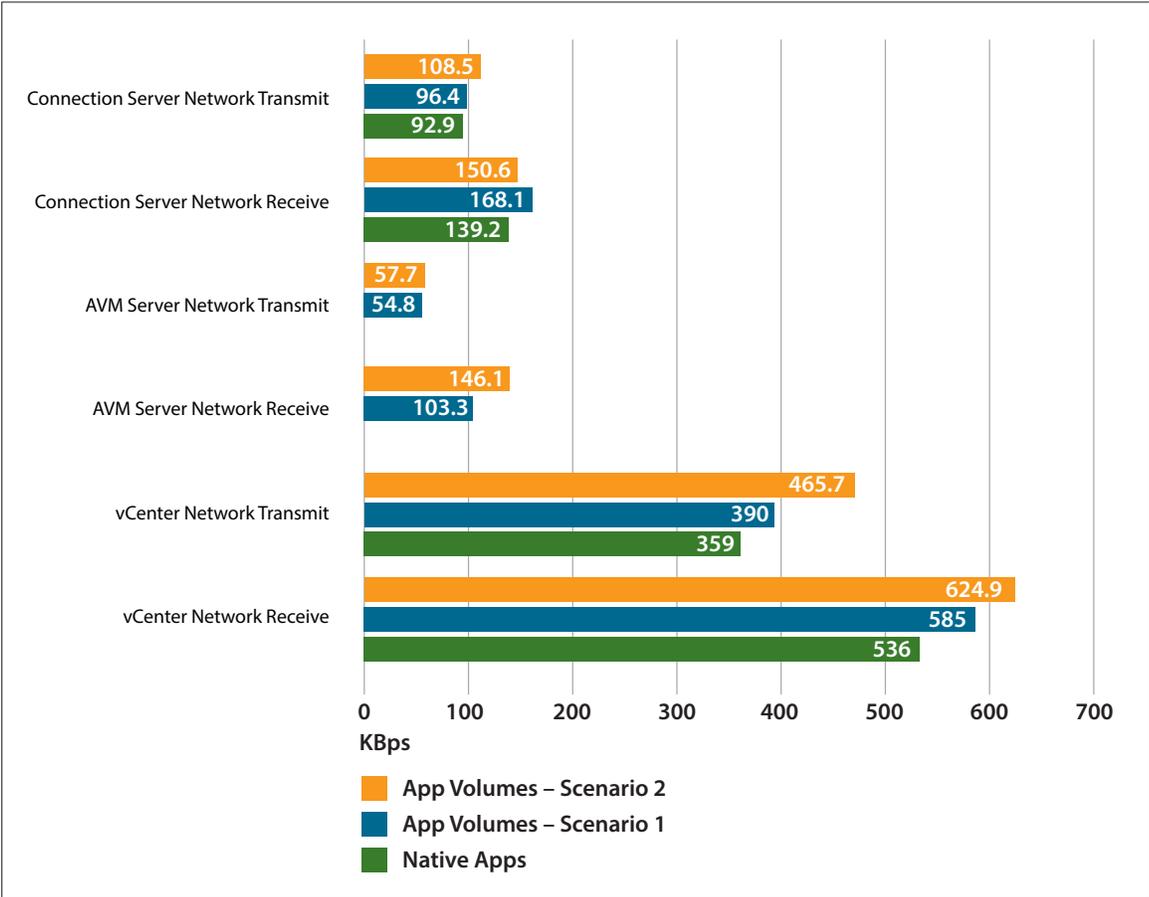


Figure 15: Infrastructure Server Average Network Activity

Throughout testing, infrastructure servers showed slightly increased network activity during App Volumes scenarios.

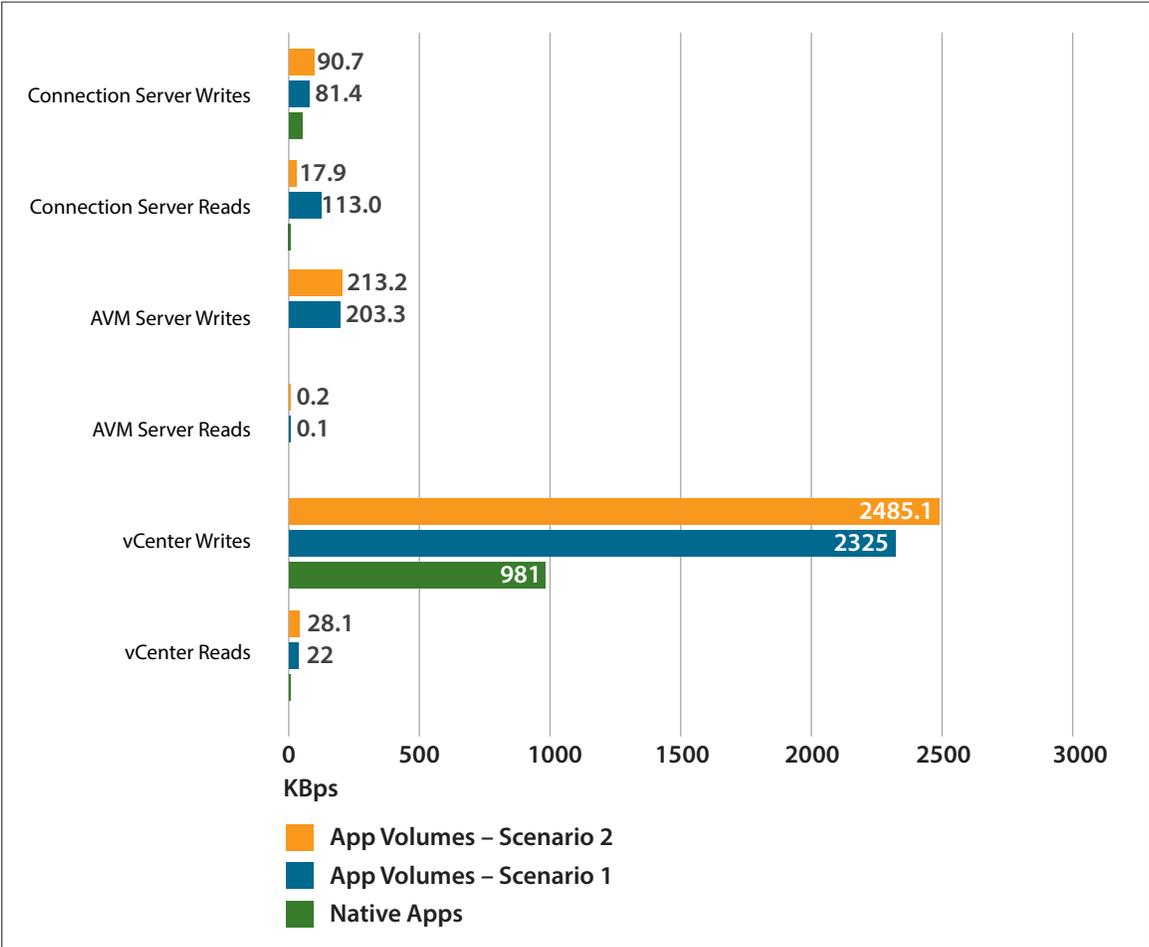


Figure 16: Infrastructure Server Average Storage Rates

VMware vCenter showed much higher storage activity during the App Volumes tests. This can be attributed to the increase in activity with AppStack and writable volumes VMDK mount and unmount operations.

### Database Performance Results

Basic capacity and performance information was collected for the Microsoft SQL Server instance that was deployed for App Volumes during this project.

SQL Metric	App Volumes Scenario 1	App Volumes Scenario 2
SQL Data File Size (MB)	73.8	151.6
SQL Transaction Log File Average Size (MB)	0.77	1.01
SQL Average Transaction per Second	8.32	10.89
SQL Peak Average Transactions per Second	11.8	260.5

Figure 17: App Volumes Database Key Metrics

The size of the App Volumes SQL database increased when we deployed three AppStacks (Scenario 2) instead of one (Scenario 1). The transaction rates increased in Scenario 2, too.

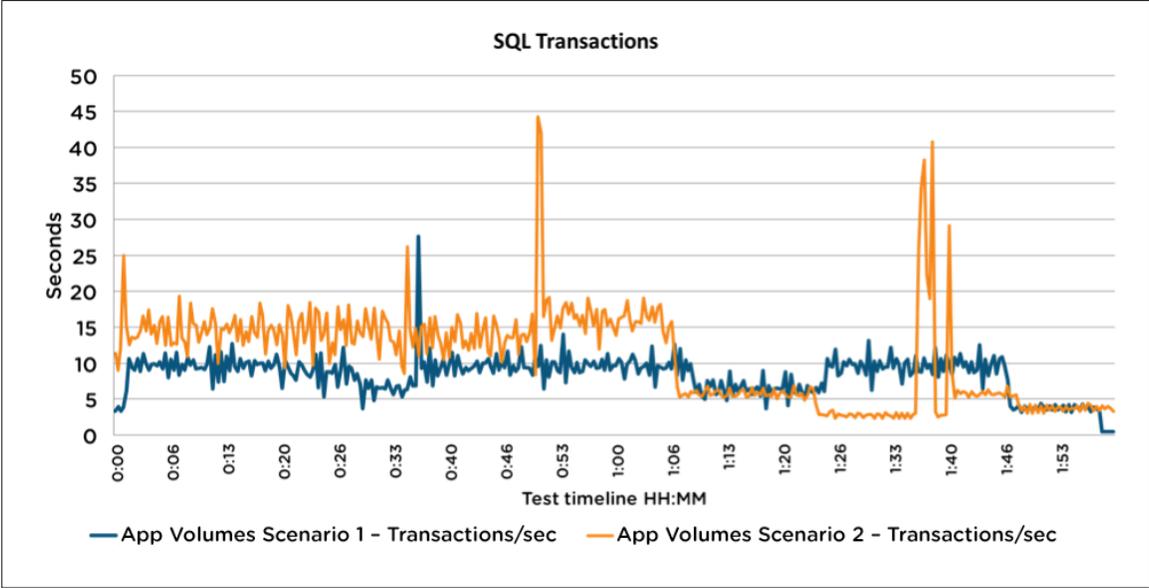


Figure 18: App Volumes SQL Transaction Rates

### Storage Performance and Capacity Results

Because App Volumes is a storage-focused technology, we collected many storage-related metrics during all phases of testing. Comparative shared storage performance and capacity metrics are highlighted in the following figures.

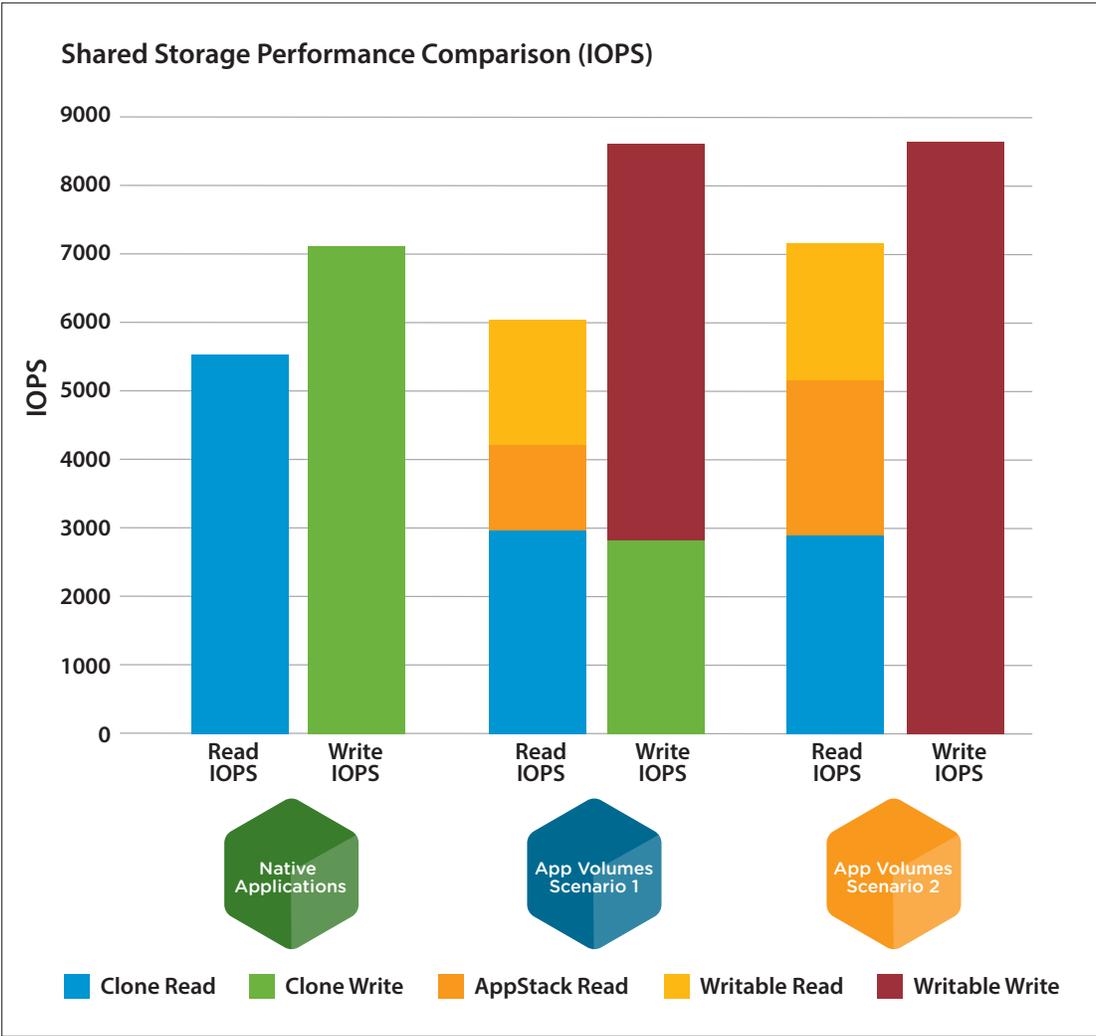


Figure 19: Storage Performance Comparison

With 1,600 active users, we observed that storage I/O consumption patterns (both reads and writes) change with the introduction of App Volumes into a View environment. The cumulative rates of both read and write I/O is higher in both App Volumes scenarios, and both reads and writes are significantly reduced for the linked clones themselves. For organizations that do not assign writable volumes (only AppStacks in use) to their users, there is a marked decrease in cumulative I/O rates. With the increased overall demand for read IOPS, and so much of that read I/O being application-centric, it is reasonable to assume that lower-latency storage will greatly impact an environment where App Volumes is deployed. Storage systems with faster read performance (VMware Virtual SAN™ and all-flash arrays) will benefit App Volumes performance.

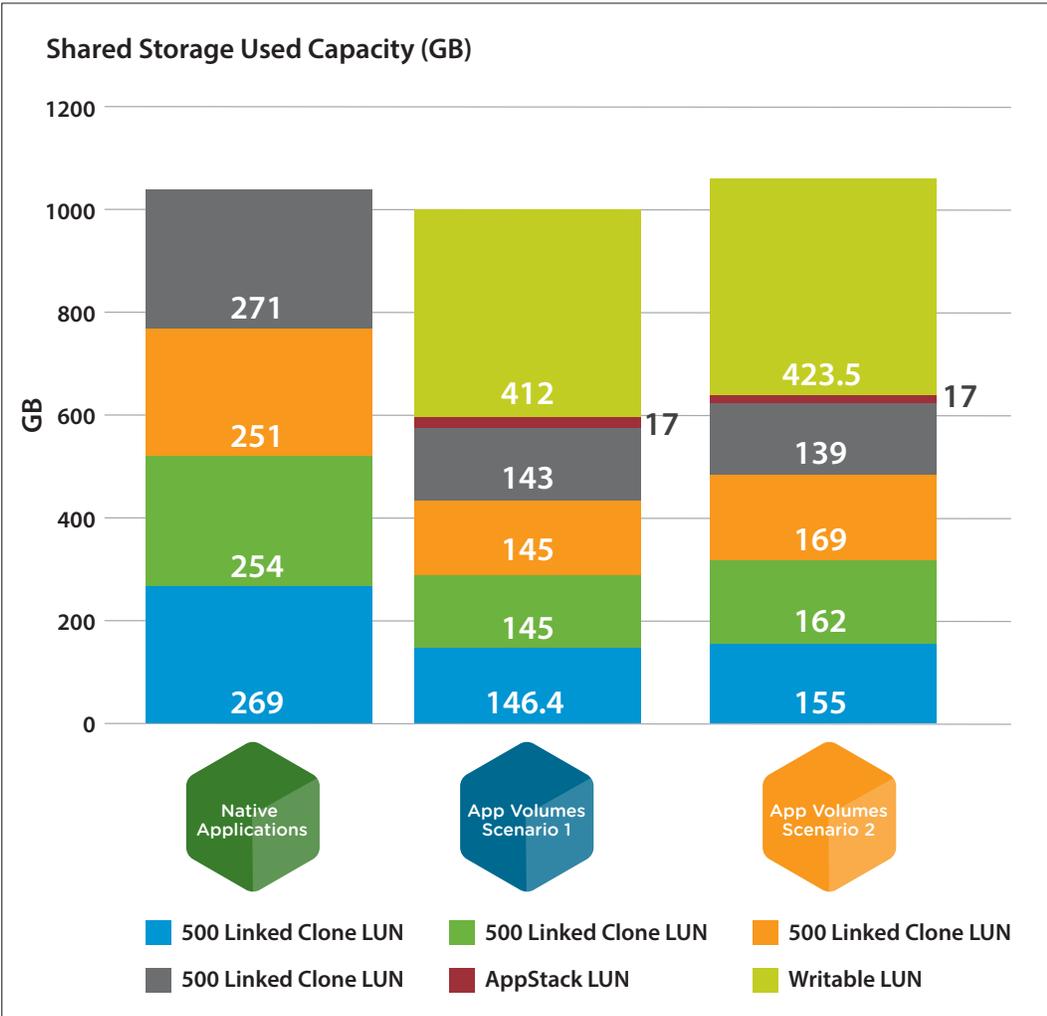


Figure 20: Storage Capacity Comparison

We observed storage capacity benefits from App Volumes as well. Linked-clone desktops consume far less storage when they all share common AppStacks. App Volumes can help minimize the hosted desktop storage footprint. If writable volumes are not deployed (users are assigned only AppStacks) there is a tremendous reduction in overall storage footprint as compared with natively installed applications. Depending on the writable volume configuration, storage capacity may be reduced as well.

## Testing Details

We took a methodical approach to all test operations. Tests were performed in an environment without background workload. All core infrastructure servers (Active Directory and SQL, for example) were not being used by any other applications.

After deploying all components in a production best-practice configuration and then conducting functional validation testing, we followed a very simple test methodology.

### Methodology

We tested natively installed applications against applications delivered as AppStacks and then tested AppStacks in a variety of user and application configurations.

#### Testing Phase - Native Applications Scenario

To begin, we ran a baseline Login VSI Knowledge Worker test with natively installed applications with 1,600 active user desktop sessions in a single 2,000-desktop View pool.

**Why:** This provided a simple baseline that showed performance and scale information for traditional, natively installed applications.

#### Testing Phase - App Volumes Scenario 1

We then ran a Login VSI Knowledge Worker test with 1,600 user desktop sessions in a single 2,000-desktop View pool, with all tested applications in a single AppStack. Each user had a single AppStack and a writable volume.

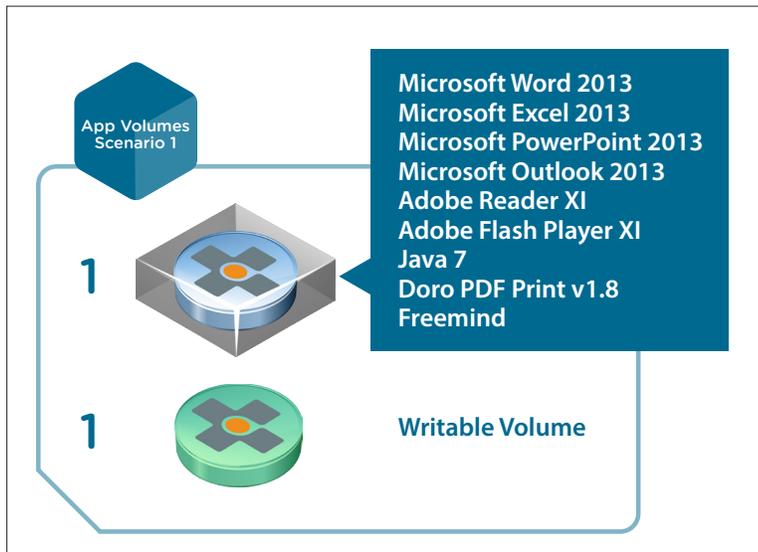


Figure 21: App Volumes Scenario 1

**Why:** We now see how performance and scalability are affected by AppStack delivery.

**Testing Phase – App Volumes Scenario 2**

In this phase, we repeated the same scenario as in Scenario 1, but splitting the tested applications in the Login VSI Knowledge Worker profile into three AppStacks. Each of the 1,600 test users was assigned three AppStacks and a writable volume.

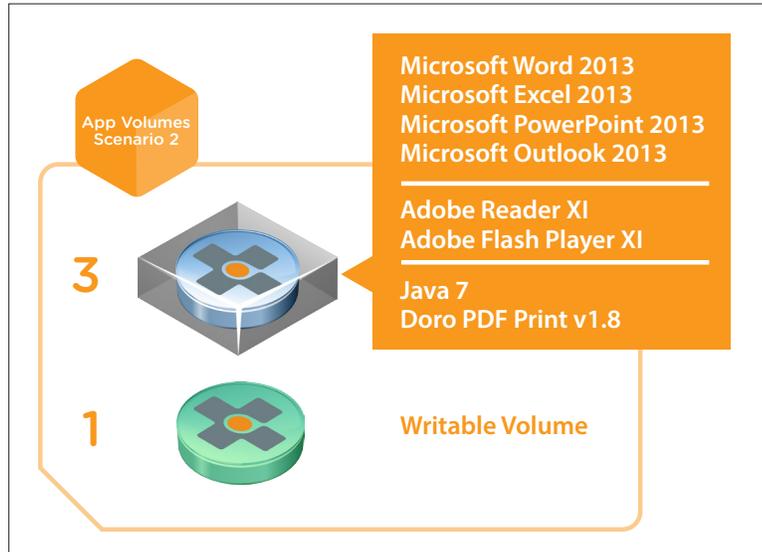


Figure 22: App Volumes Scenario 2

**Why:** By reducing the number of users per AppStack while running the same workload, we observed how lower user-to-AppStack density affects performance and scale.

**User Login Times**

During each test scenario, we used View event logs, which were fed to vRealize Log Insight, to measure the time between the initial user authentication request on the View Connection Server and the time that the user’s View session was considered to be in **Connected** status. In the App Volumes test scenarios, the AppStacks and writable volumes were mounted to the desktop virtual machine between user authentication and when the desktop is connected.

**Application Launch Times**

During all scenarios, we captured application launch times with Login VSI. Login VSI **App start** is generally measured with a window title, so that when an application command is given, a timer starts. This timer stops when the application window is visible. Typically the application window is detected by a window title that includes a test document name.

### Key Metrics

Hundreds of performance and capacity metrics were collected during testing. Some metrics are of key importance in evaluating performance, capacity, and scalability.

METRIC TYPE	SOURCE	METRIC LISTING
User-experience metrics	<ul style="list-style-type: none"> <li>• Login VSI</li> <li>• vRealize Operations Manager</li> <li>• vRealize Log Insight</li> </ul>	<ul style="list-style-type: none"> <li>• Login Times</li> <li>• Application Launch Times</li> <li>• VSI max scores</li> </ul>
Host performance and capacity metrics	vRealize Operations Manager	<ul style="list-style-type: none"> <li>• CPU Consumed</li> <li>• vRAM Consumed</li> <li>• Storage Usage (reads/writes)</li> <li>• Network Transmit/Receive</li> </ul>
vCenter performance and capacity metrics	vRealize Operations Manager	<ul style="list-style-type: none"> <li>• CPU Consumed</li> <li>• vRAM Consumed</li> <li>• Storage Usage (reads/writes)</li> <li>• Network Transmit/Receive</li> </ul>
SQL server performance and capacity metrics	vRealize Operations Manager	<ul style="list-style-type: none"> <li>• CPU Consumed</li> <li>• vRAM Consumed</li> <li>• Storage Usage (reads/writes)</li> <li>• Network Transmit/Receive</li> <li>• Database transactions per second</li> <li>• Database and log file sizes</li> </ul>
Desktop VM performance and capacity metrics	vRealize Operations Manager	<ul style="list-style-type: none"> <li>• CPU Consumed</li> <li>• vRAM Consumed</li> <li>• Storage Usage (reads/writes)</li> <li>• Network Transmit/Receive</li> </ul>
App Volumes Manager (AVM) server performance and capacity metrics	vRealize Operations Manager	<ul style="list-style-type: none"> <li>• CPU Consumed</li> <li>• vRAM Consumed</li> <li>• Storage Usage (reads/writes)</li> <li>• Network Transmit/Receive</li> </ul>
Desktop storage performance and capacity metrics	vRealize Operations Manager	<ul style="list-style-type: none"> <li>• Read Throughput</li> <li>• Write Throughput</li> <li>• Read I/O per second</li> <li>• Write I/O per second</li> <li>• Capacity Consumed</li> </ul>
App Volumes storage performance and capacity metrics	vRealize Operations Manager	<ul style="list-style-type: none"> <li>• Read Throughput</li> <li>• Write Throughput</li> <li>• Read I/O per second</li> <li>• Write I/O per second</li> <li>• Capacity Consumed</li> </ul>
User login times	vRealize Log Insight	<ul style="list-style-type: none"> <li>• Time delta between user authentication and connection View desktop</li> </ul>

METRIC TYPE	SOURCE	METRIC LISTING
XtremIO storage reports	XtremIO Management System	<ul style="list-style-type: none"> <li>• On-array latencies</li> <li>• Throughput per LUN</li> <li>• Capacity (logical and consumed)</li> <li>• Storage Controller(s) Utilization</li> </ul>

Table 3: Test Metric Types and Listings

**Test Tools**

We used a standard set of tools to orchestrate the workload, monitor the environment, and collect performance metrics.

**Login VSI**

Login Virtual Session Indexer (Login VSI) is the industry-standard benchmarking tool for measuring the performance and scalability of centralized desktop environments such as virtual desktop infrastructure (VDI) and server-based computing (SBC).

Active Directory users are systematically logged into client endpoints called launchers, which are standard PCs running the latest Horizon Client. During a test, domain users launch View desktop sessions from launchers to the target desktop pool that is under test. In this project, we used the PCoIP protocol for the desktop sessions.

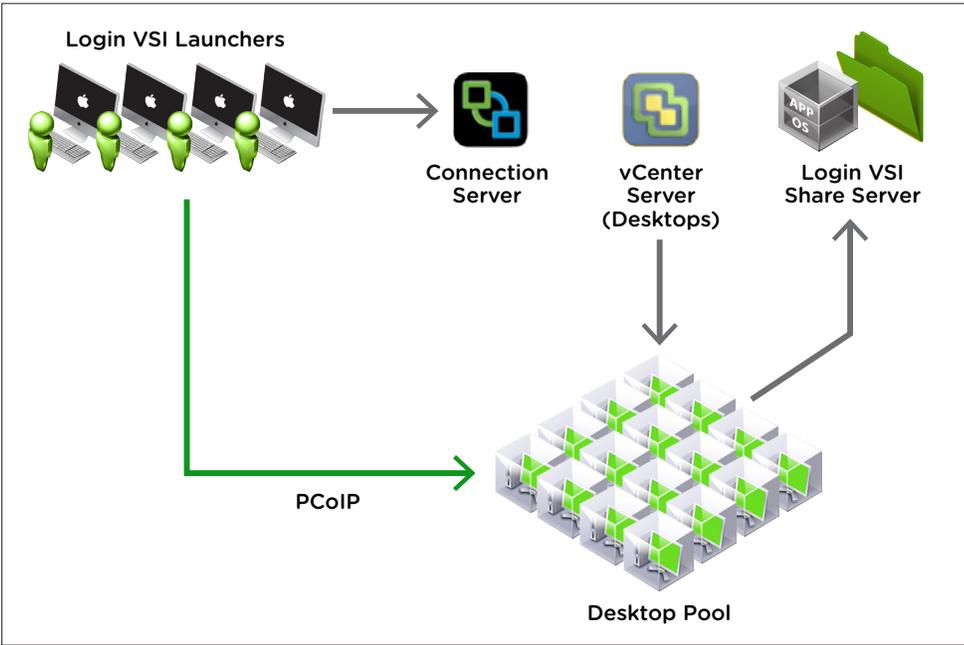


Figure 23: Login VSI Logical Diagram

We used Login VSI to generate a reproducible, real-world test case that simulated the execution of various applications, including Microsoft Internet Explorer, Adobe Flash video, and Microsoft Office. This benchmark workload was then run against various configurations of the test environment. Hardware and software remained the same, but we ran different user, desktop, and application configurations.

Various workload profiles can be run during a Login VSI test. The medium-level Knowledge Worker workload was selected for this test because it is the closest analog to the average desktop user that we see in our customer deployments.

Login VSI was configured to run a Knowledge Worker workload against a View in Horizon 6 pool of 2,000 virtual desktops, with the tests set up to log users in to virtual desktops incrementally every 4 seconds.

Once logged in, each session remained active for the duration of the test, and for at least 15 minutes after the final user has logged in, thereby ensuring full concurrency for the desired number of sessions. Not reaching VSI<sub>max</sub> is an indication of satisfactory user response at the predetermined user count.

Login VSI measured the total response time of all the applications from each session and calculated the VSI Index by taking the average response times and dropping the highest and lowest 2 percent.

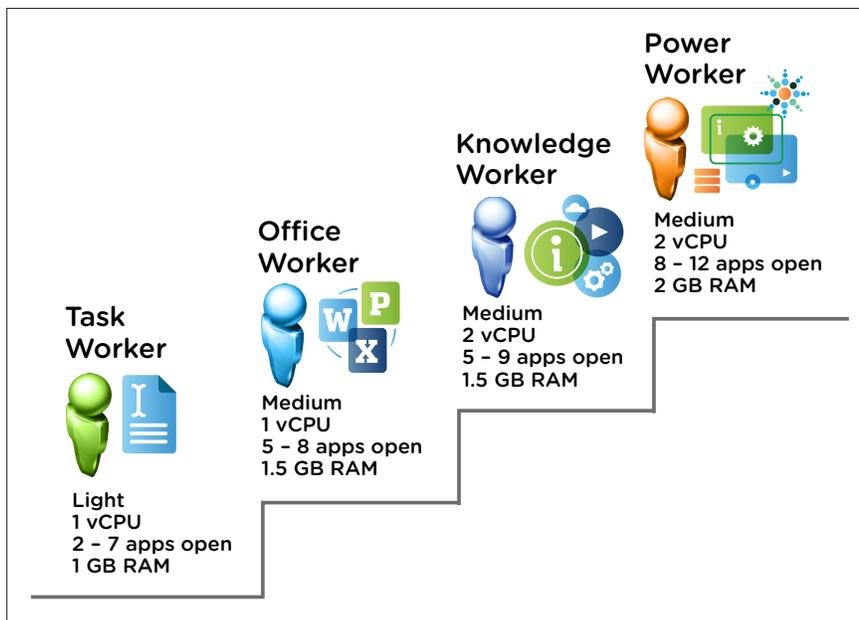


Figure 24: Login VSI Workloads

During testing, Login VSI sessions were initiated by launchers (simulated user endpoints) that ran on separate compute and storage infrastructure. A total of 200 launchers were used, each running a maximum of 10 sessions. Each launcher was configured with 2 vCPUs and 4 GB of vRAM, following Login VSI sizing guidelines. PCoIP was the display protocol in use during all tests.

**vRealize Operations Manager**

VMware vRealize Operations delivers intelligent operations management for every component in a VMware vSphere environment. It correlates data from applications to storage in a unified, easy-to-use management tool that provides control over performance, capacity, and configuration, with predictive analytics.

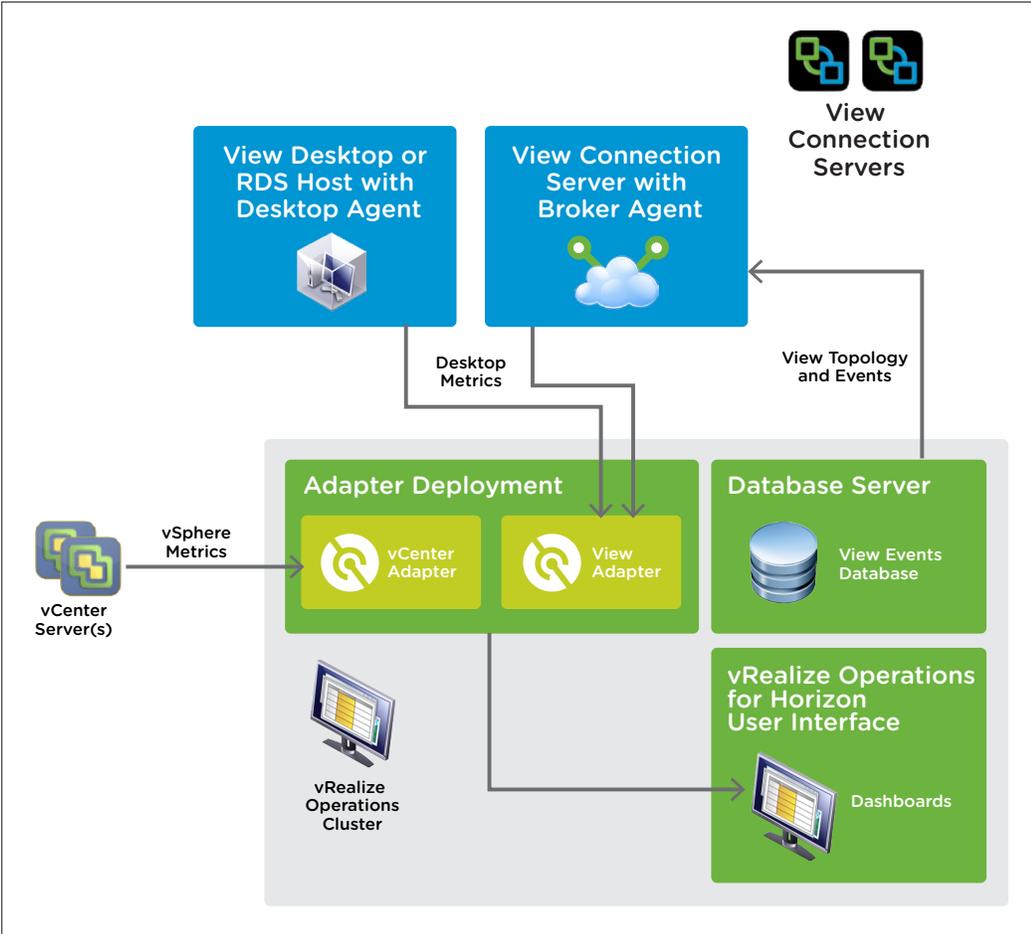


Figure 25: vRealize Operations for Horizon Logical Topology

VMware vRealize Operations was enormously useful throughout the project because we could feed data streams from vCenter servers, host servers, storage systems, and SQL databases into a single point of aggregation.

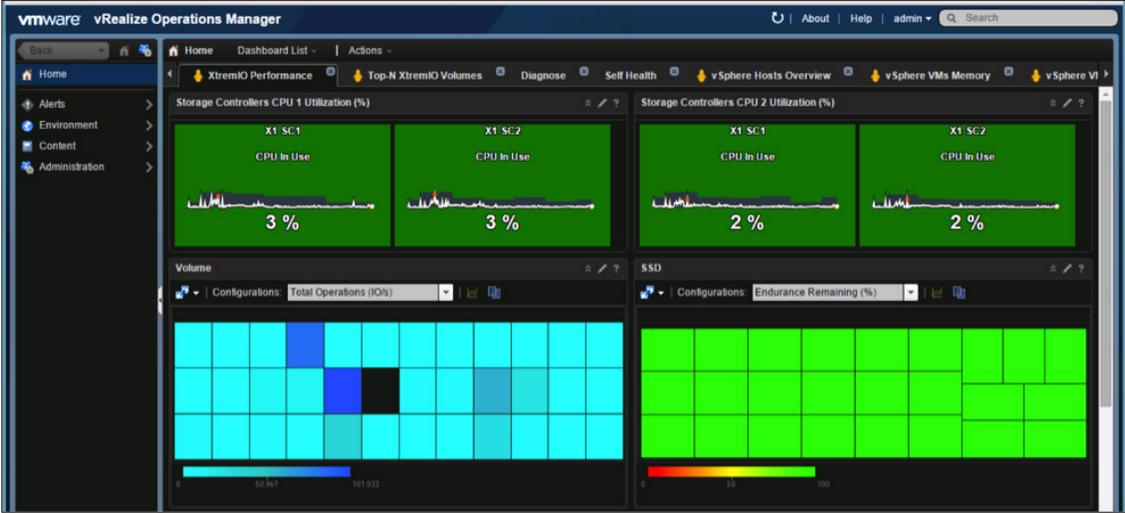


Figure 26: vRealize Operations Dashboard

The ability to create custom dashboards allowed us to profile the most meaningful metrics and focus on the time periods when each test scenario was running.

The [EMC Storage Analytics](#) (ESA) adapter was incorporated to provide a link for connecting the capabilities of vRealize Operations Manager with the available reporting metrics of the underlying EMC XtremIO storage platform. After adding ESA, storage-centric aggregated data and metrics can be presented through alerts, dashboards, or in predefined reports within the vRealize Operations Manager interface. Using ESA with vRealize Operations Manager allows for the monitoring of capacity usage, performance history of the array and individual volumes, data reduction ratios, array-side alerts, and potential configuration problems.

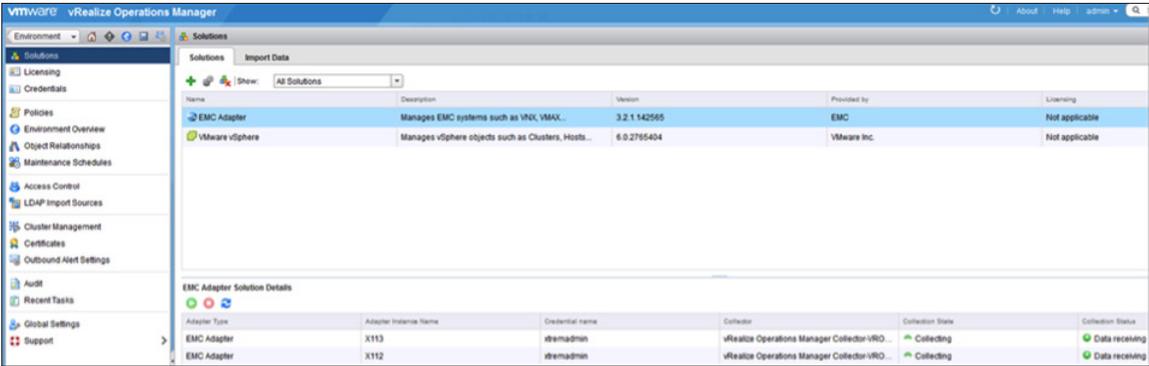


Figure 27: Integrating EMC Storage Analytics with vRealize Operations Manager

**vRealize Log Insight**

VMware vRealize Log Insight delivers real-time log management for VMware environments, with machine learning-based intelligent grouping, high-performance search, and better troubleshooting at scale across physical, virtual, and cloud environments. With an integrated cloud operations management approach, it provides the operational intelligence and enterprise-wide visibility needed to proactively enable service levels and operational efficiency in dynamic hybrid cloud environments.

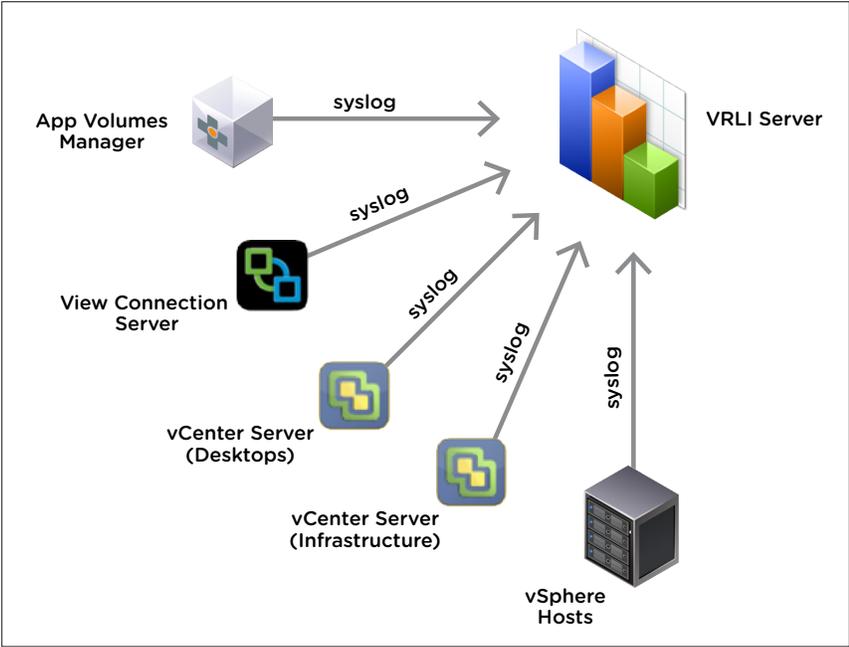


Figure 28: vRealize Log Insight Logical Topology

Where vRealize Operation Manager was used to aggregate structure data, vRealize Log Insight was used for unstructured log data from some key servers in the environment. VMware vRealize Log Insight was the syslog collector for event log feeds from the vSphere environment, the App Volumes Manager server, and the View Connection Server. By combing all logs from these various sources, it was very easy to obtain accurate timing of events such as user logins, AppStack mounts, and desktop pool recompose operations.

## Conclusions

Although App Volumes introduces some additional host, desktop, and storage resource load, it does not have an unacceptable impact on session concurrency, desktop performance, or infrastructure server workloads. User login times are slightly increased due to storage mount operations, and this is to be expected. The operational benefits of managing AppStacks centrally more than make up for slightly lower operational density.

As with all hosted desktop and application technologies, organizations should be careful about overloading vSphere hosts to the point where host resources reach a state of contention. During testing, it was observed that App Volumes introduces approximately 10 percent greater CPU load on the host as compared to the equivalent View deployment with native applications. It should also be noted that during App Volumes testing, we pushed our host CPU usage averages beyond 80 percent with only 80 percent of the desktops in use.

Because our hosts were operating at close to 100 percent utilization in some tests, some upward skew in terms of realized latencies can be expected between the results shared in this paper and those seen in non-overloaded environments. Organizations should identify the sweet spot of active desktop-to-host density, where hosts are running 80 percent CPU and memory usage at the desired active session concurrency.

The success of any virtual desktop delivery scheme can typically be measured relative to the end-user experience and ability to match that of a physical desktop or workstation. App Volumes is a storage-centric technology and, as with all comparable desktop and application-delivery technologies, lower-latency storage tends to provide the best user experience. During testing, we saw excellent user experience, primarily because we deployed EMC XtremIO, a low-latency, data-aware, all-flash storage solution that was coupled with an ultra-fast storage network. We encourage organizations to specifically consider all-flash Virtual SAN or enterprise-class all-flash SAN storage for their App Volumes environments, for a number of reasons. First, because AppStacks are essentially read-only volumes that are shared across many VMs, these datastores should be deployed on storage that can deliver consistently high IOPS and low latencies when subjected to intense random read-request I/O profiles of a highly localized nature. A distributed all-flash storage platform is well suited to such a workload. Second, App Volumes-based configurations have a much heavier per-desktop read and write I/O resource requirement, so there is a need to provide consistently low response times to ensure optimal end-user experience.

The Storage Groups feature in App Volumes v2.10 is a good way to balance load across multiple datastores. By spreading AppStacks across multiple volumes and using a round-robin policy for writable volumes, we ensured that our workload and capacity were properly balanced, giving us a better chance of eliminating storage performance hotspots and capacity bottlenecks.

In our testing with increased application-to-AppStack density, we observed little performance gain with either configuration. There were observed differences in user login times, storage workload, and vCenter performance, but the differences were not significant. It is likely that with a larger number of AppStack assignments per user, login times could become unreasonable.

During testing, we used a single View Connection Server and a single App Volumes Manager server to make sure that the View and App Volumes infrastructure systems could handle the maximum published workload. In production environments, this is not a good practice; there should always be redundancy in the environment at the server level. A better production design would include placing a load balancer in front of a team of App Volumes Manager servers as well as a load-balanced View Connection Server team. This design would allow a full population of users to be supported in the event that a server fails or is offline for maintenance. Additionally, load balancing across multiple systems helps to prevent situations where a server might become overloaded.

## About the Authors

Tristan Todd is an Architect in the VMware End-User-Computing Technical Enablement Group. He has extensive customer, field, and lab experience with VMware End-User-Computing and ecosystem products.

Tirtha Bhattacharjee is a Member of the Technical Staff, QE, in the End-User-Computing Enterprise Horizon Readiness Team at VMware, where he works in interoperability projects for VMware Horizon, App Volumes, and other end-user-computing products.

Girish Narkhede is leading the Scalability and Interoperability effort in the End-User-Computing Enterprise Readiness Team at VMware. In over seven years at VMware, Girish has worked in various product areas and played several engineering roles, including Staff Engineer, leading the vSphere/ESXi scalability validation effort.

Michael Cooney is a Principal Solutions Architect for XtremIO, a division of EMC Corporation. He is responsible for delivering virtualized enterprise solutions that highlight the value of the XtremIO all-flash array. Michael has been with EMC for eight years.

## Acknowledgements

VMware recognizes its partners' generosity in providing equipment, time, and expertise, without which this reference architecture project would not have been possible.

EMC is a global leader in enabling businesses and service providers to transform their operations and realize the Software-Defined Data Center. Fundamental to this transformation is cloud computing. Through innovative products and services, EMC accelerates the journey to cloud computing, helping IT departments to store, manage, protect, and analyze their most valuable asset, information, in a more agile, trusted, and cost-efficient way. For more information, visit <http://www.emc.com>.

Login VSI is an international software company focused on helping both end users and vendors of virtual desktop infrastructures to design, build, implement, and protect the best-performing hosted desktop infrastructures possible. For more information, visit <http://www.loginvsi.com>.

# Appendix A: Environment Design

Extensive planning went into every element of the design of the vSphere, View, and App Volumes environment that was used during their project. VMware best practices were referenced and followed at all times. Wherever possible, the configuration conforms to recommendations for production environments.

## vSphere Design

VMware vSphere was deployed in a standard configuration. Two virtual data centers were deployed, one for infrastructure and management systems, and one dedicated to desktops and applications. A total of 24 physical hosts were deployed, and all hosts shared an identical hardware configuration (see [Appendix C: Bill of Materials](#)). Shared storage was connected using Fibre Channel over Ethernet. Each host server was configured with multiple 10 GbE network adapters.

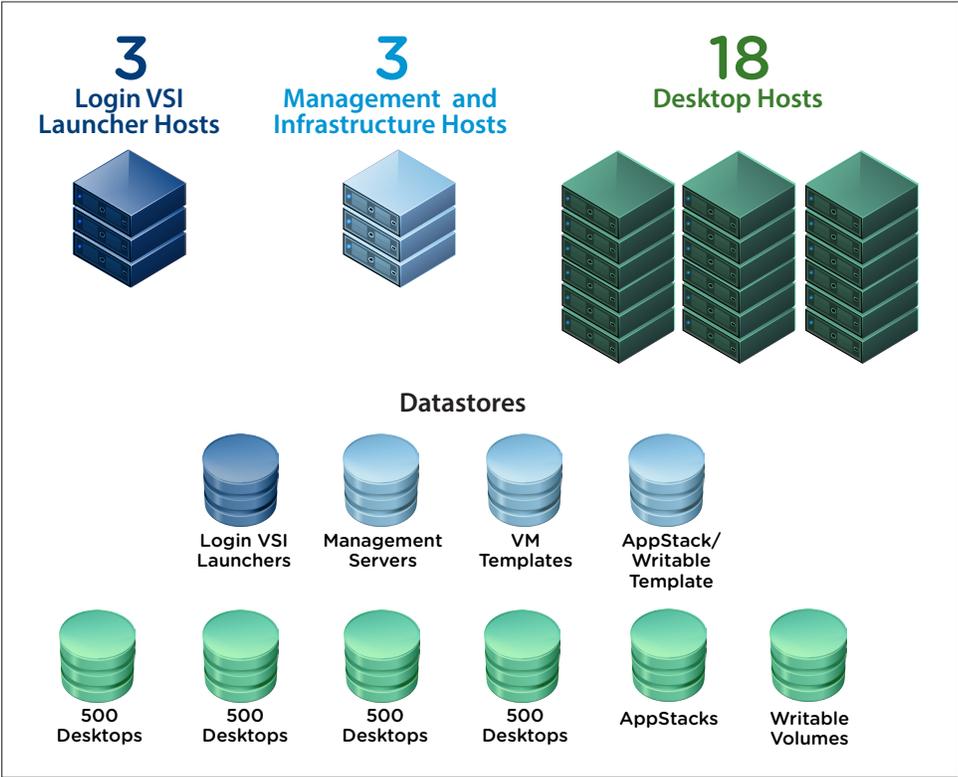


Figure 29: Host and Datastore Layout

Three vSphere clusters were deployed, each sized and configured to house virtual machines based on their role:

- Infrastructure and Management Cluster
- Login VSI Launcher Client Cluster
- Desktop Cluster

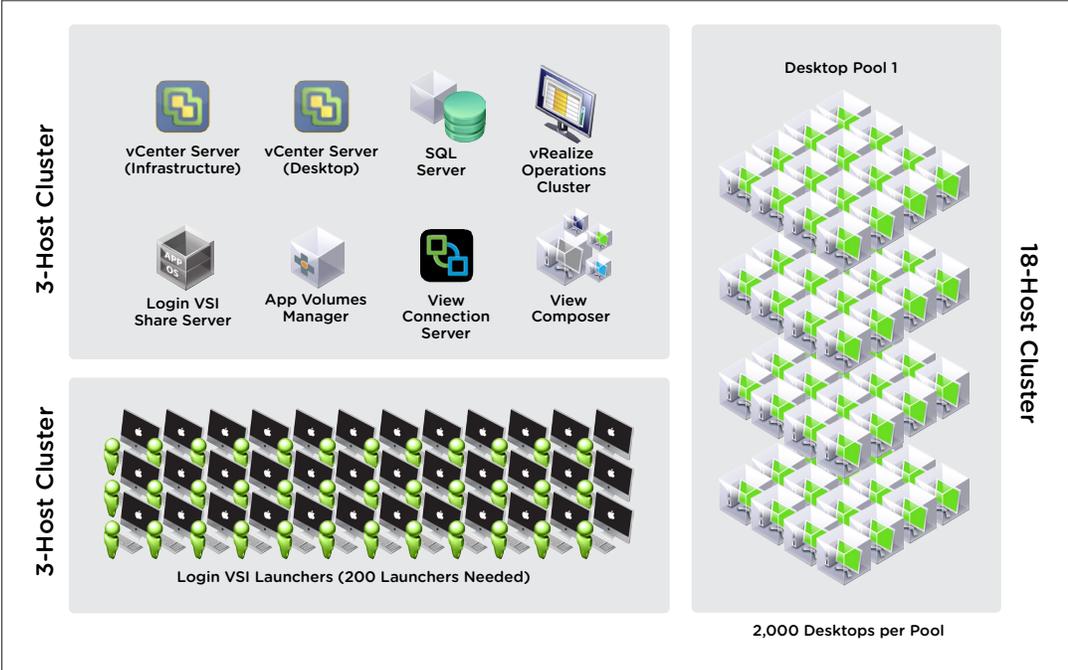


Figure 30: Three-vSphere-Cluster Layout

The vSphere environment was segregated with some virtual machines being deployed in the appropriate virtual data center, vSphere host, and datastore according to their role.

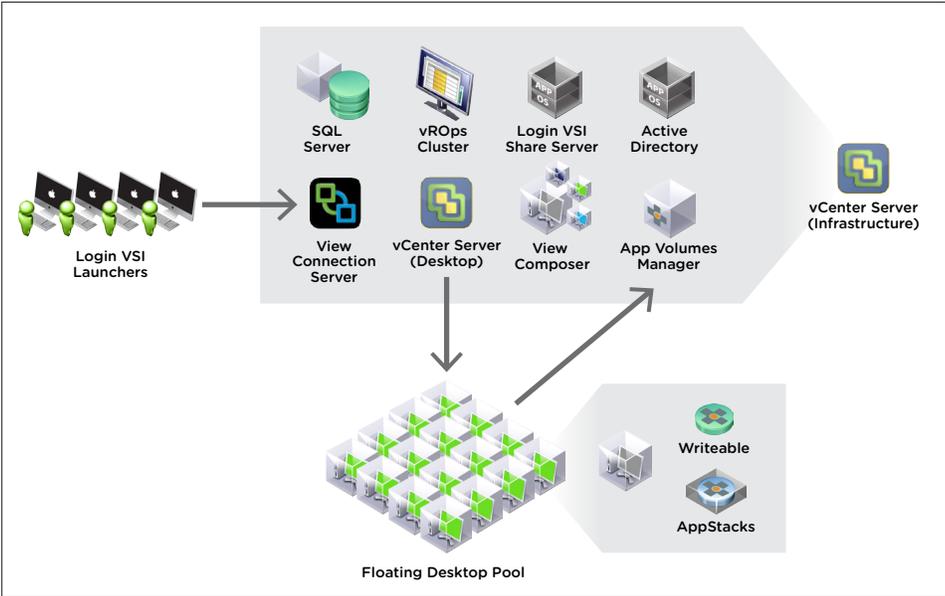


Figure 31: vCenter Separation

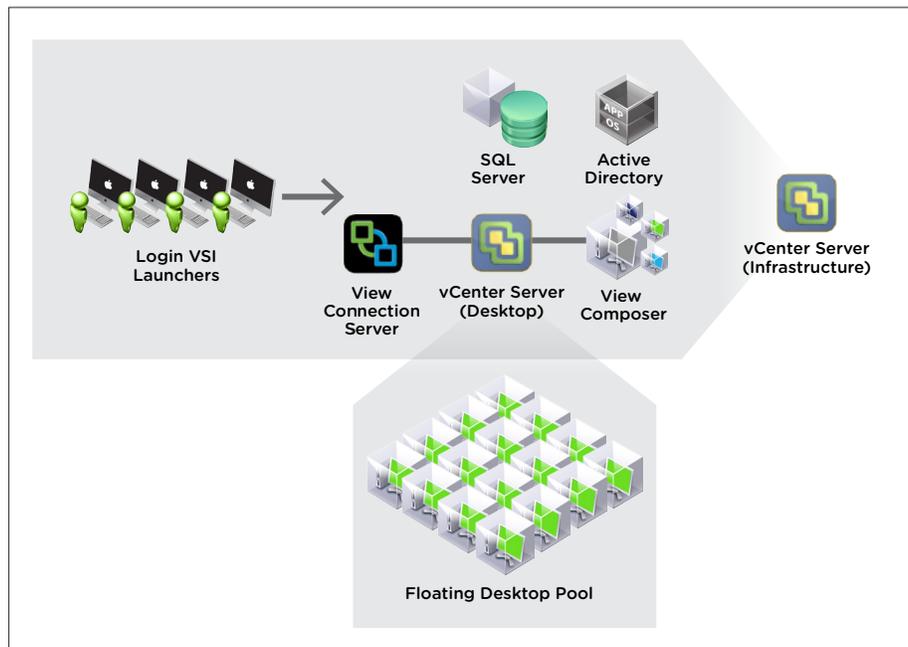
VMware vSphere host servers and vCenter servers were deployed in a best-practice configuration. VMware vCenter servers were deployed using the Linux-based appliance and were sized according to vSphere 6 sizing best practices.

VMware vSphere clusters were configured with HA and DRS enabled. DRS policy was set to default, moderate configuration. VMware vSphere clusters were set to place virtual machine swap file on a datastore that was deployed specifically to house vSwap files. At the time of testing, vSphere hosts were patched with all available VMware updates.

The basic vSphere configuration, including version and build information, is listed in [Appendix C: Bill of Materials](#).

## View Design

A very basic View environment was used. A single View Connection Server was used, although a second replica View Connection Server (not depicted) was deployed and used only for scripting, reporting, and integration with vRealize Log Insight. This second View Connection Server was never configured to handle any client connections.



**Figure 32:** High-Level View Layout

## View Connection Servers

The basic View configuration and all appropriate settings are listed in [Appendix B: Test Environment Settings](#).

### App Volumes Design

A basic App Volumes deployment was used during testing. A single App Volumes Manager was used. It was sized according to published best practices and in accordance with all requirements and prerequisites.

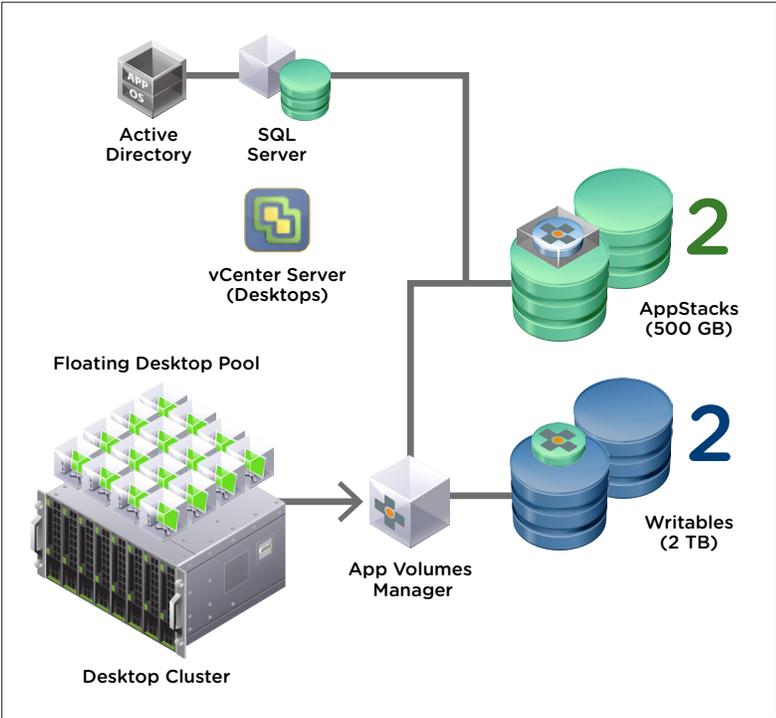


Figure 33: App Volumes Deployment Configuration

### AVM Key Settings

A Microsoft SQL 2012 database was housed on a production-grade SQL Server that was built according to Microsoft published best practices. Standard Active Directory integration was configured, with only domain administrators granted permissions to log into the App Volumes Manager server.

### Storage Group Configurations

Two storage groups were deployed to house AppStacks and writable volumes.

STORAGE GROUP CONFIGURATION	APPSTACKS	WRITABLE VOLUMES
Intended use	AppStacks only	Writables only
Individual datastore size	500 GB	2 TB
Storage distribution policy	Spread	Round-robin
Storage selection policy	Direct	Direct
Automatically import AppStacks	Yes	N/A
Automatically replicate AppStacks	Yes	N/A

Table 4: App Volumes Storage Groups

**AVM Configuration and Settings**

The AppStack storage group was configured to replicate AppStacks between both datastores, and the spread distribution policy ensured that users were connected to AppStacks in a round-robin fashion.

The Writable storage group was configured to distribute user writable volumes in a round-robin fashion, and it was confirmed that both datastores hosted an equal number of writable volumes throughout testing.

App Volumes was configured to use the **Mount on Host** option for all volume mount operations. VM reconfiguration commands to hot-add the disk to present the AppStacks or writable volumes were sent directly to vSphere hosts, rather than being processed by vCenter. We also successfully performed selective test runs with **Mount on Host** disabled.

**AppStack Layouts**

All of the tested (Login VSI) applications that were part of the App Volumes tests were delivered by AppStack, with the exception of Internet Explorer, which was installed natively. A total of four AppStacks were built and maintained during this project.



Figure 34: AppStacks

## Storage Configuration

A total of twelve datastores were provisioned on the shared EMC XtremIO storage system.

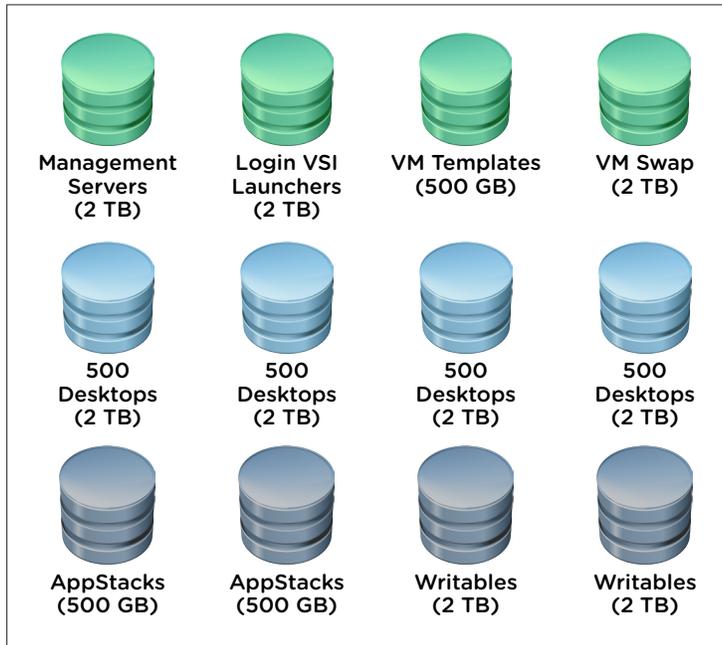


Figure 35: Basic LUN Layout

### vSphere Host Storage-Specific Configurations

To ensure optimal performance for the test environment described in this paper, we performed the following tasks on the vSphere hosts in accordance with [EMC published best practices](#) for use with XtremIO all-flash storage arrays:

- Confirmed that VAAI was enabled and set XCOPY maximum transfer size (DataMover.MaxHWTransferSize) to 256 KB.
- Deployed all VMs with Thick Provision Eager Zero format.
- Set the maximum number of active storage commands allowed at any given time at the VMkernel level (Disk.SchedNumReqOutstanding) to the maximum allowed value of 256.
- Set the maximum number of consecutive sequential I/Os allowed (Disk.SchedQuantum) from one VM before switching to another VM to the maximum permissible value of 64.
- Changed the Host Bus Adapter (HBA) queue depth settings maximum of 256 for each host's HBAs. This value differs depending on the HBA vendor.
- Set Disk Max I/O Size (Disk.DiskMaxIOSize) to 4 MB.
- Changed the native storage multipath policy in vSphere to Round Robin with IOPS=1.

### vCenter Concurrency Settings

Because low-latency storage was utilized, we adjusted the vCenter concurrency settings in View Administrator to provide faster pool provisioning operations.

SETTING	VALUE	APPLICABILITY
Maximum concurrent vCenter provisioning operations	30	Applies to full clones only
Maximum concurrent power operations	40	Applies to full and linked clones
Maximum concurrent View Composer maintenance	40	Applies to linked clones only
Maximum concurrent View Composer provisioning operations	40	Applies to linked clones only

**Table 5:** vCenter Concurrency

During testing, the fastest 2,000-desktop linked clone deployment took 80 minutes. The fastest 2,000-desktop linked clone recompose operation took 127 minutes. All timing was measured in the View Administrator interface.

Both [VMware](#) and [XtremIO](#) offer conservative guidance in relation to the advised number of maximum concurrent vCenter operations permissible. Because of previous experimentation and a deep technical knowledge of both the hypervisor layer and underlying storage, this solution was able to increase those values beyond typical recommendations without fear of exceeding the infrastructure's capabilities.

**Note:** It is always advisable to follow vendor-issued best practices and amending these settings without a deeper understanding of the underlying VMware vSphere mechanics is not advised.

### Storage Platform

This test environment was designed with the purpose of understanding the performance characteristics of scaling an App Volumes-integrated View infrastructure, so it was desirable to use a high-performance storage solution that delivered high IOPS and low latencies and to avoid any performance limitations that could affect the potential scalability and validity of performance metrics recorded during the test scenarios. That is why the test environment used EMC XtremIO best-in-class all-flash shared storage array.

[EMC XtremIO](#) is an all-flash block storage array that is purposely designed to maximize the performance and storage efficiencies available through the use of flash-based storage technology while dramatically simplifying storage administration, configuration, and optimization. XtremIO uses industry-standard hardware and proprietary data-aware intelligent software to take advantage of flash media to deliver extremely high levels of performance in addition to consistent and predictable low latency response times.

The XtremIO AFA consists of the following building blocks:

- The X-Brick is the physical building block of the XtremIO AFA. This is the basic hardware building block for these scale-out arrays. Each X-Brick has two active-active controller nodes and a disk array enclosure packaged together with no single point of failure.
- The XtremIO Management Server (XMS) is a stand-alone Linux-based server that is used to manage the XtremIO cluster. This can either be deployed as a physical 1U server or as a VMware virtual appliance.
- The XtremIO Operating System (XIOS) is the intelligence behind the data-aware storage features available with XtremIO. It runs on each of the storage controllers.
- An InfiniBand interconnect provides 40 Gb-per-second-capable RDMA paths between all storage controllers.



Figure 36: EMC X-Brick Storage Appliance

An X-Brick is a high-availability, high-performance SAN storage appliance available in 5 TB, 10 TB, 20 TB, or 40 TB capacity physical configurations that can provide close to ultra-high IOPS and can support thousands of virtual desktops within a 6U package. XtremIO uses a multi-controller active-active architecture. It can therefore scale out linearly in terms of performance, capacity, and connectivity through the addition of more X-Bricks.

The scale-out, global data reduction architecture of XtremIO is specifically optimized for flash to enable multiplying effects across many aspects of the array. This in turn leads to benefits that include extending the effective capacity of the array as well as minimizing the required writes to media to improve XtremIO hosted application performance and increase the usable lifespan of the purchased flash per host data written. The XtremIO data reduction architecture consists of the following components:

- **Content-addressable data engine** – Enhances data reduction, balances data, augments efficiency, and increases performance.
- **Global scale-out metadata engine** – Delivers fast, consistent performance experience across all array service for all applications.
- **Always-on inline data services** – Provides data services which never stop working and never have to be disabled—including thin provisioning, data deduplication, compression, and space-efficient writable snapshots.
- **XtremIO Data Protection (XDP)** – Provides flash-specific data protection with no legacy from disk-based RAID, which is faster than RAID 10, better than RAID 6 protection, and requires less overhead than RAID 5.
- **XtremIO Virtual Copies (XVC)** – Augment data reduction by enabling multiple writable copies of application datasets that consume zero physical data.

The single X-Brick used for all of our testing was more than sufficient to provide adequate capacity and I/O performance for 2,000 linked-clone desktops and all associated management and monitoring infrastructure servers.

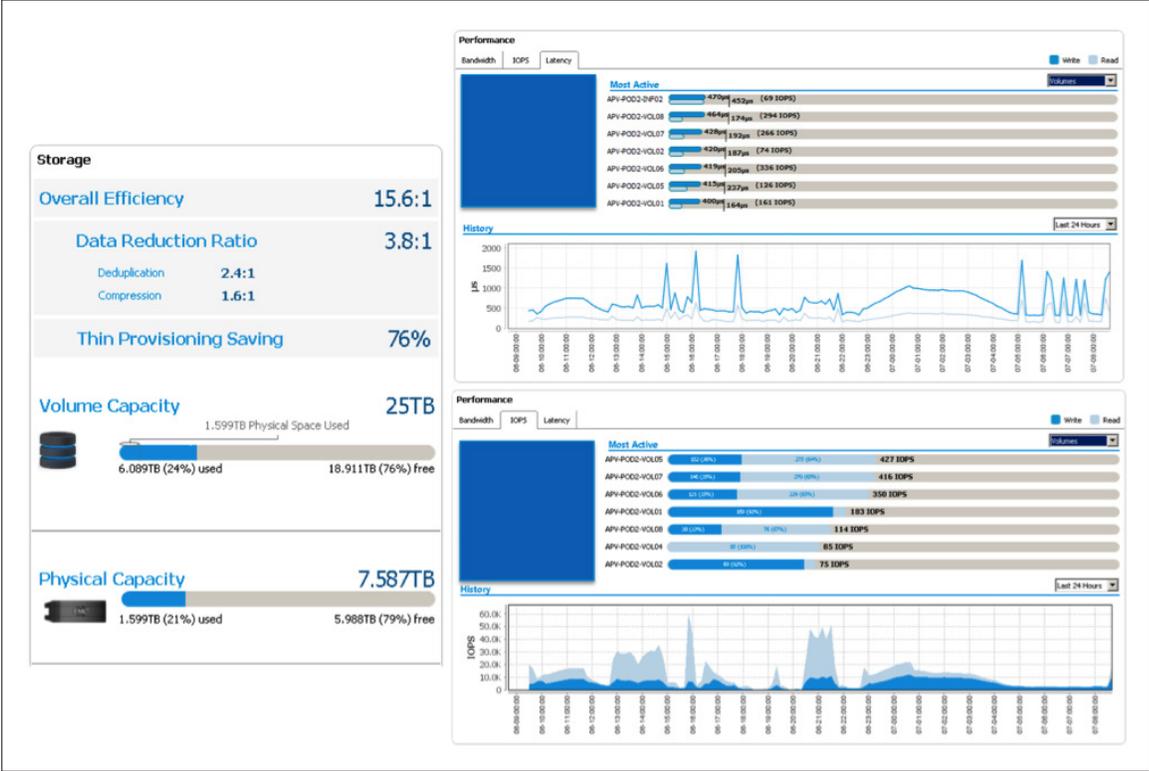


Figure 37: XtremIO Management Dashboard

During the project we provisioned 25 TB worth of shared storage volumes to vSphere, but our actual capacity footprint on the X-Brick was quite modest.

### Network Configuration

The Dell server chassis was configured to allow each server blade to have access to 4x10 GbE adapters.

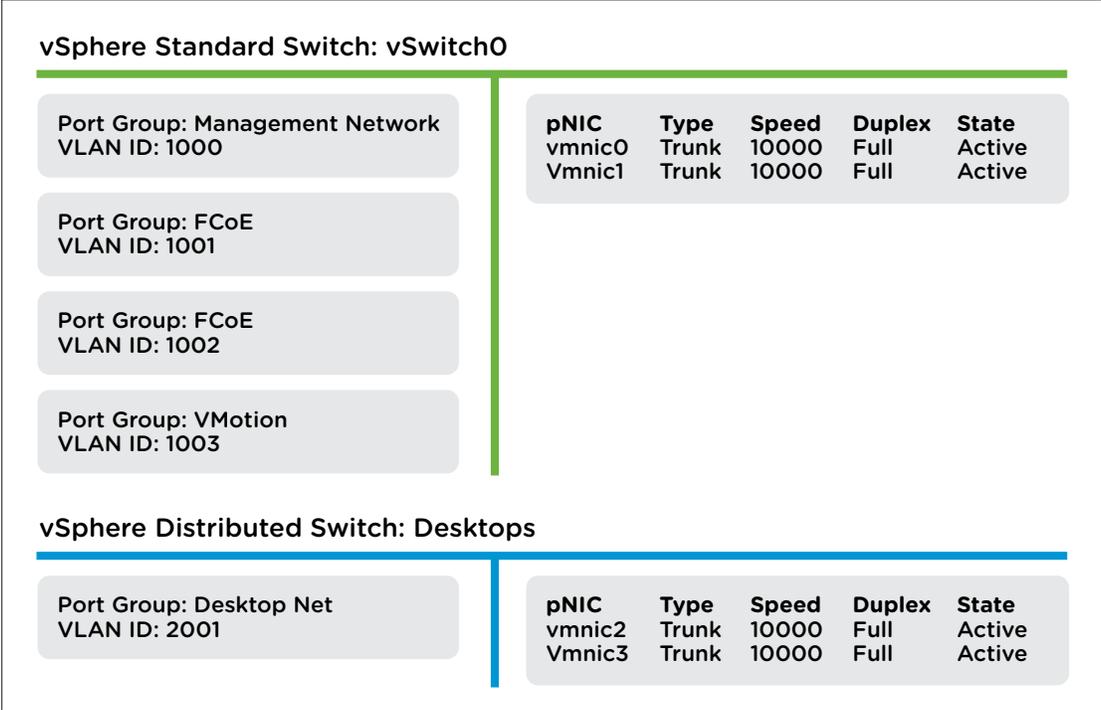


Figure 38: Host Network Configuration

Two adapters were used as uplinks for a vSphere Standard Switch that was used for the host management network, FCoE access, and VMware vSphere vMotion® traffic. Two adapters were used as uplinks for a vSphere Distributed Switch that was used for desktop virtual machines.

### Desktop Master Images

We built target desktop images according to VMware best practices. A master Windows 7 64-bit image with two vCPUs and 2 GB of vRAM was built with VMTools and virtual hardware version to match the vSphere operating environment.

We then optimized the image for VDI with the OS Optimization Tool and made all recommended VDI-specific configuration changes.

For App Volumes provisioning virtual machine usage, this image did not have the View Agent installed. For target desktop images, the View Agent and App Volumes Agents were installed.

Because the host servers were each outfitted with 384 GB of physical RAM, and because we did not want to use high-performance shared storage for virtual machine swap files, each server had a 100 percent memory reservation set.

We used the Active Directory Group Policy (GPO) template to set PCoIP configurations for all desktops.

CONFIGURATION	SETTING
Build to Lossless	Off
Maximum Initial Image Quality	80 percent
Minimum Image Quality	40 percent
Video Frame Rate	30 fps

**Table 6:** PCoIP Configurations

We also configured two other sets of GPO policies in Active Directory.

CONFIGURATION	SETTING
Disable Automatic Updates	Disabled
Don't show messages while viewing a document	Disabled
Show messages while I launch Reader	Disabled
Turn off user participation in the feedback program	Disabled
Enable protected Mode at Startup	Disabled

**Table 7:** Adobe Reader GPO Policies

CONFIGURATION	SETTING
Disable First Run Movie	Enabled
Disable Office First Run on application boot	Enabled
Disable Opt-in Wizard on first run	Enabled

**Table 8:** Microsoft Windows Settings

**Important:** The master image used in this test environment underwent VDI optimization as defined in the VMware [View Optimization Guide for Desktops and Servers in View in VMware Horizon 6](#). VMware strongly recommends that the Windows image be optimized when master images are prepared for use with View in Horizon 6.

## Appendix B: Test Environment Settings

The settings for key components are contained in the tables below.

### Infrastructure Server Virtual Machine Configurations

ATTRIBUTE	DETAILS
OS	Appliance, SUSE Linux 11
vCPU	8
vRAM	24 GB
Storage	320 GB

**Table 9:** vCenter VM Details

ATTRIBUTE	DETAILS
OS	Microsoft Windows Server 2012
vCPU	4
vRAM	10 GB
Storage	82 GB

**Table 10:** View Connection Server VM Details

ATTRIBUTE	DETAILS
OS	Microsoft Windows Server 2012
vCPU	4
vRAM	8 GB
Storage	80 GB

**Table 11:** App Volumes Manager VM Details

## View in Horizon 6 Server Configurations and Settings

We used the following settings for linked-clone testing:

VIEW POLICY	SETTING
Multimedia Redirection	Deny
USB Access	Allow
PCoIP Hardware Acceleration	Allow, priority medium
View Admin Session Timeout	1200 minutes
Forcibly disconnect users	Never
After warning, logoff after	One minute
Display pre-login message	Disabled

**Table 12:** Linked-Clone Test Settings

## Desktop Pool Configurations

We used the following settings for desktop-pool testing:

POOL PARAMETER	SETTING
Pool Type	Automated Desktop Pool
User Assignment	Floating
Desktop Power Policy	Ensure machines are always powered on
Delete or refresh machine on logoff	Refresh Immediately
Automatic Logoff	Never
Display Protocol	PCoIP
Monitors	Two
Monitor Resolution	1920 x 1200
Adobe Flash Quality	Do Not Control
Adobe Flash Throttling	Disabled
Disposable Disk	None
Number of Desktops	2,000
Number of Desktops Powered On	2,000
Provision All Desktops Up Front?	Yes
Linked Clone Overcommit Policy	Aggressive
Dedicated Replica Storage	None (replica in each linked-clone datastore)

**Table 13:** Desktop-Pool Test Settings

## Desktop Image Configuration

We used the following settings for image configuration:

ATTRIBUTE	SPECIFICATION
Desktop OS	Microsoft Windows 7, 64-bit
VMware virtual hardware	Version 11
VMware Tools version	vSphere 6 (up to date)
Virtual CPU	Two
Virtual memory	2048 MB, 100% memory reservation
OS pagefile	1.5 GB starting and maximum
vNICs	One
Virtual network adapter 1	VMXNet3 Adapter
Virtual SCSI controller 0	LSI Logic SAS
Virtual Disk - VMDK	40 GB
Virtual Floppy Drive 1	Removed
Virtual CD/DVD Drive 1	Removed
VMware View Agent	v6.2, build 3005627
VMware App Volumes Agent	v2.10
Installed applications (per Login VSI standards)	<ul style="list-style-type: none"> <li>• Microsoft Office 2013</li> <li>• Internet Explorer 10</li> <li>• Adobe Acrobat Reader XI</li> <li>• Adobe Flash Player XI</li> <li>• Doro PDF Print</li> <li>• FreeMind</li> </ul>

**Table 14:** Desktop Image Configuration Settings

## Login VSI Test Parameters

We used the following Login VSI test parameters:

CONFIGURATION ITEM	SETTING
Enable Session Monitor	Yes
Verify Runtime Files at Start	Yes
Office Version	2013
PDF Printer	Yes
Disable Video Viewing	Yes
Maximum Loops	10
Logoff Session at	Loop
Random Initial Wait	30 seconds
Log application start times	Yes
Log shell execute times	Yes
Overall logon rate	Session count x 4 seconds (one session every four seconds)
Sessions per launcher	10

**Table 15:** Login VSI Test Parameters

## Appendix C: Bill of Materials

This section details the major hardware and software components that were used during this project.

### Hardware BOM

The test configuration bill of materials is summarized in the following table.

AREA	COMPONENT	QUANTITY
Host hardware	<ul style="list-style-type: none"> <li>Dell PowerEdge M620 (2 x Intel E5-2660, 20 core, 2.2 GHz)</li> <li>384 GB of RAM</li> </ul>	24
Storage hardware	<ul style="list-style-type: none"> <li>EMC X-Brick</li> <li>Firmware version v4.0.1 build 41</li> </ul>	1
Network hardware	<ul style="list-style-type: none"> <li>Brocade VDX 8770</li> <li>Cisco MDS 9148 (Fibre Channel switch)</li> </ul>	2 2

**Table 16:** Hardware Test Configuration

### Software BOM

The test configuration bill of materials is summarized in the following table.

COMPONENT	VERSION	BUILD
VMware ESXi	6.0U1	3073146
VMware Horizon 6	6.2	3005368
VMware App Volumes	2.10	1412
VMware vRealize Operations	6.1	3038036
VMware vRealize Log Insight	3.0	3201606
View Composer	6.2	3001314
Login VSI	4.1.4	

**Table 17:** Software Test Configuration

## Appendix D: Additional Resources

### VMware

[Application-Delivery Options in Horizon 6](#)  
[Optimization Guide for Desktops and Servers in View in VMware Horizon 6](#)  
[Reviewer's Guide for View in Horizon 6](#)  
[VMware App Volumes Technical Documentation](#)  
[Horizon 6 Application-Delivery Decision Maker](#)  
[VMware Horizon with View and Virtual SAN Reference Architecture](#)  
[VMware Horizon Architecture Planning Guide](#)  
[VMware Horizon 6 Storage Considerations](#)  
[VMware Horizon 6 Technical Documentation](#)  
[VMware OS Optimization Tool](#)  
[VMware View Security](#)  
[vRealize Operations Technical Documentation](#)  
[vRealize Log Insight Technical Documentation](#)

### Login VSI

[Login VSI Technical Documentation](#)  
[Login VSI VSImax Overview](#)  
[Login VSI Technical Introduction](#)  
[Login VSI Analyzing Results](#)

### EMC XtremIO

[EMC XtremIO Technical Resources](#)  
[VMware Horizon 6 with XtremIO Design Considerations](#)  
[EMC XtremIO Desktop Virtualization Technical Resources](#)  
[Expand Your Horizon with XtremIO](#)  
[VMware Horizon 6 with XtremIO 3.0 Design Considerations](#)  
[VSPEX EUC Design Guide: VMware Horizon View 6.0 with vSphere 5.5 on XtremIO 3.0](#)  
[VSPEX EUC Implementation Guide: VMware Horizon View 6.0 with vSphere 5.5 on XtremIO 3.0](#)

## Blogs

[VMware EUC Blog](#)

[Login VSI Blog](#)

[EMC XtremIO Blog](#)

## VMware Hands-On Labs

[Application Management with VMware App Volumes and Horizon 6 \(HOL-MBL-1458\)](#)

[Advanced Technical Concepts of Horizon 6 from A to Z \(HOL-MBL-1651\)](#)

