VMmark User's Guide

VMmark 3.1

Updated July 14, 2019

vmware®

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Revision: 20190714

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

Figures

Tables

Overview

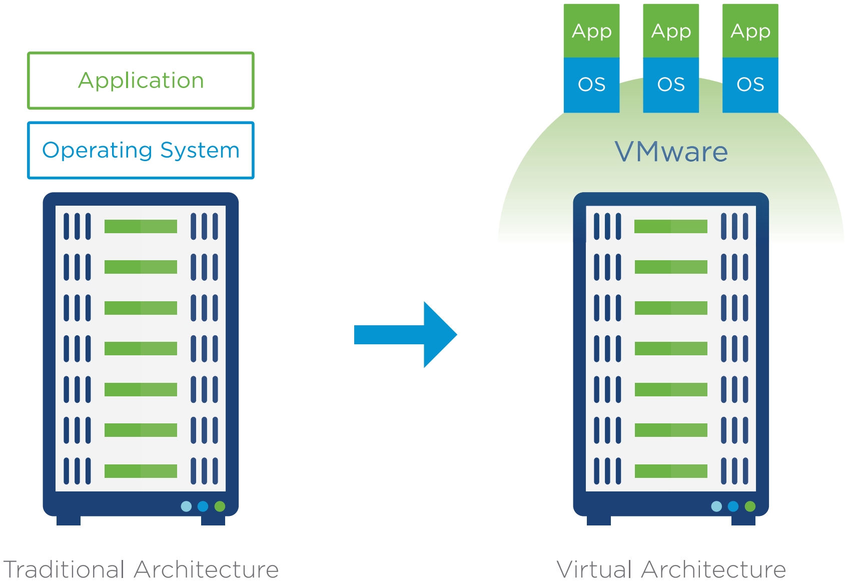
This document describes the VMmark® virtual infrastructure benchmarking utility, it provides instructions for configuring a system for VMmark testing, it details the steps required to perform such a test, and it discusses the interpretation of the data acquired.

Why Virtualization?

Server virtualization is the process of running multiple virtual computer systems (called "virtual machines") on a single physical server, as shown in Figure 1.

Virtualization can increase IT agility, flexibility, and scalability while creating significant cost savings. Workloads get deployed faster, performance and availability increases, and operations can be automated, resulting in IT that's simpler to manage and less costly to own and operate.

Figure 1. Example Organization of a Virtual Environment



Why a Multi-Host Virtual Machine Benchmark?

It is crucial for users to have meaningful and precise metrics in order to effectively compare the suitability and performance of different platforms for virtual environments. Traditional single-workload performance and scalability benchmarks for non-virtualized environments were developed with neither virtual machines nor server consolidation in mind. These metrics focus on achieving maximum system performance for a single workload by driving at least one of the underlying hardware resources to saturation. These types of benchmarks do not provide sufficient insight into the scalability of virtual environments supporting multiple simultaneous workloads.

VMmark 1.x was a multi-workload server consolidation benchmark that measured single-host performance in virtualized environments. It used a unique tile-based implementation in which each "tile" consisted of a collection of virtual machines running a set of diverse workloads. This tile-based approach has been used in all subsequent versions of VMmark.

VMmark 2.x was a multi-host virtual machine benchmark that addressed the increasing virtualization of bursty and heavy workloads, dynamic virtual machine relocation (VMware vSphere® vMotion®), dynamic datastore relocation (VMware vSphere Storage vMotion®), and the automation of many provisioning and administrative tasks across large-scale multi-host environments. In this paradigm, some of the stress on CPU, network, disk, and memory subsystems is generated by the underlying infrastructure operations. Load balancing across multiple hosts can also affect application performance. While still focusing on user-centric application performance, this benchmark also accounted for the effects of this infrastructure activity on overall platform performance.

Continuing the philosophy that multi-host virtual machine benchmarks provide the closest to real world metrics for users to assess virtual environments, VMmark 3 was developed to address the changes in applications and infrastructure operations. In the years since VMmark 2 was released, virtualization has become the norm for applications, and these applications have evolved. Users are no longer questioning if Microsoft Exchange can be virtualized, for example, but are now considering highly scalable workloads (often referred to as platform 3) and running more complex OLTP workloads than ever before. Additionally, with VMware vSphere vMotion without shared storage, and today's highly robust virtual machine deployment workflows, infrastructure operations have changed in load, frequency, and complexity. VMmark 3 was created with this innovation and datacenter evolution in mind and provides a modern benchmarking methodology that accounts for today's and tomorrow's platform performance.

Intended Audience

This document is written for users with a relatively advanced understanding of system administration. However, familiarity with virtualization software and with benchmarking methodology is not assumed.

Legal Notice

This documentation contains information including but not limited to the installation and operation of the Software. Modifications, additions, deletions or other updates ("Modifications") to the information may be incorporated in future releases.

VMware®, Inc., its affiliates or subsidiaries ("VMware") are not responsible for any Modifications made to the published version of this documentation unless performed by VMware. All information is provided "as is" and is believed to be accurate at the time of publication. VMware shall not be liable for any damages arising out of or in connection with the information and recommended actions provided herein (if any), including direct, indirect, consequential damages, loss of business profits or special damages, even if VMware has been advised of the possibility of such damages.

What is the VMmark Benchmark? 1

This chapter provides an overview of the VMmark Benchmark and describes how it works. The chapter consists of the following sections:

Overview of the VMmark 3.x Benchmark

State-of-the-art server consolidation typically collects several diverse workloads onto a virtualization platform—a collection of physical servers accessing shared storage and network resources. Dynamic load balancing techniques ensure that all system resources such as CPU, network, and disk are more efficiently utilized. In fact, virtual environments tend to function more smoothly when resource demands are balanced across multiple resources.

The unit of work for a benchmark of virtualized consolidation environments can be naturally defined as a collection of virtual machines executing a set of diverse workloads. The VMmark 3.x Benchmark follows the convention of VMmark 1.x and VMmark 2.x and refers to this unit of work as a *tile*. The total number of VMmark tiles a multi-host platform can accommodate gives a coarse-grain measure of that platform's consolidation capacity. This concept is similar to some server benchmarks, such as TPC-C (see http://www.tpc.org/tpcc), that scale the workload in a step-wise fashion to increase the system load.

Tiles are relatively heavyweight objects that cannot by themselves capture small variations in platform performance. To address this, both the number of tiles and the performance of each individual workload determine the overall benchmark score.

Each workload within a tile is constrained to execute at less than full utilization of its virtual machine. However the performance of each workload can vary to a degree with the speed and capabilities of the underlying platform. The addition of a fast disk array, for example, might result in disk-centric workloads producing a more favorable score. These variations can capture system improvements that do not warrant the addition of another tile. However the workload throttling forces the use of additional tiles for large jumps in platform performance.

When a tile is added, the performance of the workloads in existing tiles might decrease. If the system has not been overcommitted, however, and the minimum quality-of-service metrics are met, the aggregate score, including the new tile, should increase. The result is a flexible benchmark metric that provides a measure of the total number of workloads that can be supported by a particular multi-host platform as well as the overall performance level within the workload virtual machines.

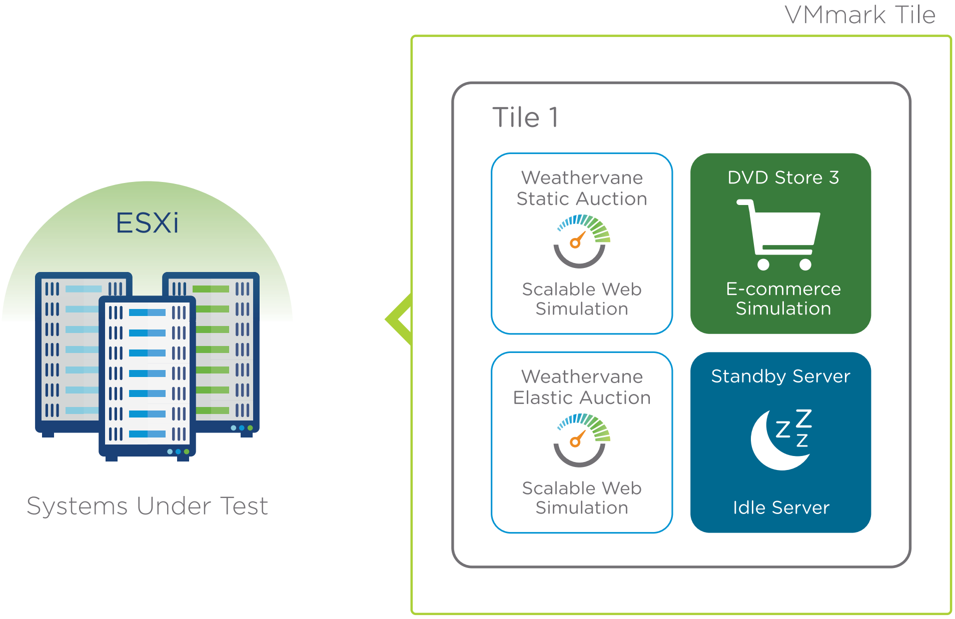
If two different virtualization platforms achieve similar VMmark scores with a different number of tiles, the score with the lower tile count is generally preferred. The higher tile count could be a sign that the underlying hardware resources were not properly balanced. Studying the individual workload metrics is suggested in these cases.

VMmark 3.x generates a realistic measure of platform performance by incorporating a variety of platform-level workloads in addition to traditional application-level workloads. Live migration both of virtual machines and of their underlying disk files has quickly become a powerful and commonplace tool in virtualized datacenters. Likewise, the ability to quickly clone and deploy virtual machines onto available resources has transformed traditional server provisioning. Each of these operations place non-trivial resource demands on the underlying infrastructure and must be included in order to accurately characterize platform performance.

Virtual machine migration, clone and deploy, storage migration, and vMotion without shared storage operations are repeatedly performed on a set of the workload virtual machines to simulate the additional resource demands typical in production datacenters. Additionally, automated load balancing is enabled to ensure application-level workloads are relocated to satisfy their resource needs as the computational loads vary among the individual hosts over time.

VMmark is designed to benchmark the performance of the virtualization software and hardware and is not designed as a benchmark of any other software component.

Figure 1-1. Conceptual Example of a Cluster of Hosts Running a VMmark Tile



VMmark Benchmark Workloads

A meaningful tiled consolidation benchmark is based on a set of relevant datacenter workloads. A survey of datacenter applications led to the inclusion of the following workloads:

- Standby system
- Scalable web simulation
- E-Commerce simulation

Rather than develop workloads from scratch, existing workloads or benchmarks were used where possible. This reduces implementation effort and provides a well-understood foundation upon which to build.

The workloads chosen for use in VMmark are representative of popular applications commonly run by VMware customers. These four workloads are summarized in Table 1-1.

Table 1-1. VMmark 3.x Workload Summary

Workload	Application	Virtual Machine Platform
Scalable Web Simulation	Weathervane Static	Auction Web (2): CentOS 7.2 64-bit, 2 vCPU, 8GB RAM, 16GB disk Auction App (2): CentOS 7.2 64-bit, 4 vCPU, 14GB RAM, 16GB disk Auction LB: CentOS 7.2 64-bit, 2 vCPU, 4GB RAM, 16GB disk Auction MSQ: CentOS 7.2 64-bit, 2 vCPU, 4GB RAM, 16GB disk Auction DB: CentOS 7.2 64-bit, 2 vCPU, 8GB RAM, 16GB disk, 20GB data disk AuctionNoSQL: CentOS 7.2 64-bit, 2 vCPU, 16GB RAM, 16GB disk, 100GB data disk
Scalable Web Simulation	Weathervane Elastic	Elastic Web (2): CentOS 7.2 64-bit, 2 vCPU, 4GB RAM, 16GB disk Elastic App (2): CentOS 7.2 64-bit, 2 vCPU, 8GB RAM, 16GB disk Elastic LB: CentOS 7.2 64-bit, 1 vCPU, 4GB RAM, 16GB disk, 25GB data disk Elastic DB: CentOS 7.2 64-bit, 2 vCPU, 8GB RAM, 16GB disk
E-Commerce Simulation	DVD Store 3	Database: CentOS 7.2 64-bit, 8 vCPU, 32GB RAM, 16GB base disk, 250GB & 100GB data disks Web (3): CentOS 7.2 64-bit, 1 vCPU, 4GB RAM, 16GB disk
Standby	n/a	CentOS 7.2 64-bit, 1 vCPU, 2GB RAM, 16GB disk
Deploy	n/a	CentOS 7.2 64-bit, 4 vCPU, 8GB RAM, 16GB base disk, 10GB data disk

VMmark leverages components of these workload virtual machines to run some common virtualization procedures, which include:

- Virtual machine cloning and deployment
- Dynamic virtual machine relocation between servers
- Dynamic virtual machine relocation across storage
- Simultaneous server and storage virtual machine relocation
- Automated load balancing

The following sections discuss each of these workloads.

Scalable Web Simulation

Scalable web applications are used to provide a wide variety of services, such as social networking and on-line auction web sites. These applications will typically have a core application that implements the business logic, surrounded by a variety of support services such as load balancers, web servers, message servers, and databases. The application logic may be distributed among multiple independently deployed services. Each tier in a scalable web application deployment might stress different infrastructure resources. For example, the application servers might have high CPU demands, while the data services might place high demands on storage or memory.

Weathervane is an application-level benchmark for virtual infrastructure and cloud performance tests. The Weathervane application, named Auction, is a scalable web application that implements a web site for hosting real-time auctions. The Auction application uses a scalable architecture that allows deployments to be easily sized for a large range of user loads. A deployment of the application involves a wide variety of support services, such as caching, messaging, NoSQL data-store, and relational database tiers. Additional information about Weathervane is available at https://github.com/vmware/weathervane.

In VMmark 3.x, the Weathervane workload uses two independent instances of the Weathervane Auction application, a static instance and an elastic instance. The virtual machines used in these application instances, all running Centos 7.2, are shown in Table 1-1. Each instance includes:

- a load balancer running HAproxy 1.5.18
- web servers running Nginx 1.12.0
- application servers running Tomcat 8.5.13 and Java 1.8.0.121
- a message server running RabbitMQ 3.5.3
- a database running PostgreSQL 9.3
- a NoSQL data service running MongoDB 3.0.14

In the static application instance, all of these services run on their own virtual machines. In the elastic application instance, the message server, load balancer, and NoSQL data service share a single virtual machine.

The static application instance, as its name implies, creates a relatively consistent load on the system under test.

The elastic application instance, on the other hand, is both elastic and bursty. Because in today's datacenters it's increasingly common to have self-scaling applications that dynamically add and remove resources to meet demands, VMmark 3.x takes advantage of Weathervane's elasticity-related capabilities to add and remove an application server and a web server throughout the run. This elastic component (along with the cyclical application profile generated by DVD Store 3, described below) allows VMmark 3.x to more accurately represent today's bursty environments.

The load for Weathervane is generated by a workload driver that simulates users interacting with the Weathervane Auction application. The load generated by each user is constant as long as the application can satisfy its quality-of-service (QoS) requirements. These QoS requirements specify the 99th-percentile response-time for each operation, as well as the required mix of operations performed by all users. The performance metrics from Weathervane include the operation throughput, the average response-time for each operation, and the percentage of each operation that completes within the response-time limits.

E-Commerce Simulation

Databases running transactional workloads support a wide array of applications, typically as part of a multi-tier architecture. Databases tend to be resource intensive and exercise most server and infrastructure components. In many cases, database systems also face strict response-time demands. Transaction processing often exhibits bursty behavior, resulting in widely varying resource demands over time. The ability of the underlying platform to support usage spikes is critical to maintaining acceptable performance.

DVD Store Version 3 (DS3) is a complete online e-commerce test application with a back-end database component, a web application layer, and driver programs (see http://github.com/dvdstore/ds3 for additional information). The DS3 driver simulates users logging into a web server and browsing a catalog of products using basic queries. Users may select items for purchase then proceed to check out or continue shopping. Each web server communicates with a database server that maintains user account and inventory data.

The DS3 workload used in VMmark 3.x utilizes four virtual machines in each tile—three web servers and one database server—all running 64-bit CentOS 7.2. The three virtual machines in the DS3 web tier (DS3WebA, DS3WebB, and DS3WebC) each run the Apache 2.4.6 web server and have one virtual CPU and 4GB of memory. The DS3 database tier runs the MySQL database on a virtual machine with eight virtual CPUs and 32GB of memory.

One of the web servers delivers a constant load to the database throughout each benchmark interval. The other two web servers deliver periodic load to the database during the benchmark interval to create a bursty overall load profile and varying resource demands. For VMmark 3.x, each web server is driven by 24 driver threads when active. The performance metric for this workload is the total number of transactions per minute. Minimum quality-of-service metrics must also be met.

Virtual Machine Cloning and Deployment

Creating a new virtual machine and installing a guest operating system and applications can be time consuming. Using virtual machine cloning technology, administrators can make many copies of a virtual machine using a single installation and configuration process. Cloning, configuration, and deployment operations create bursty loads on platform resources, particularly the storage subsystem as the virtual machine files are copied.

This infrastructure workload clones the VMmark template virtual machine, powers-on and pings the clone, takes a snapshot, performs a hot add of CPU and memory, takes another snapshot, creates a small MySQL database, then reverts the snapshots, again pings the clone, and finally deletes it. The benchmark then waits 40 seconds and repeats this process, continuing for the duration of the benchmark period. The number of concurrent clone and deploy operations increases with the number of tiles and the number of hosts in the benchmark cluster. The performance metric used is the number of clone and deploy operations per hour.

Dynamic Virtual Machine Relocation Between Servers

Live migration technology such as VMware vMotion® leverages the complete virtualization of servers, storage, and networking to move an entire running virtual machine seamlessly from one server to another. During a vMotion operation, the active memory and precise execution state of a virtual machine is rapidly transmitted over a high speed network from one physical server to another and access to the virtual machine’s disk storage is instantly switched to the new physical host. This transition can result in bursty loads on platform resources, particularly the networking subsystem. VMmark mimics the manual relocation of a virtual machine, which can be a common task performed by an administrator.

This infrastructure workload acts on one of the AuctionMSQ virtual machines selected in a round-robin fashion from among all the tiles. A destination host is selected at random from among all hosts in the benchmark cluster (other than the virtual machine’s current host), the virtual machine is moved to the destination host, left there for two minutes, then returned to its original host. VMmark then waits another two minutes and repeats this process, continuing for the duration of the benchmark period. The number of concurrent relocation operations increases with the number of tiles and the number of hosts in the benchmark cluster. The performance metric used is the number of relocations per hour.

Dynamic Virtual Machine Relocation Across Storage

Live migration of virtual machine disk files across or within storage arrays enables enormous flexibility for storage maintenance, upgrades, and load balancing. Storage relocations can create bursty loads on platform resources, particularly the storage subsystem.

In this infrastructure workload, VMmark relocates a virtual machine's disk files to a maintenance partition, then returns them to their original location. This round-trip approach models an administrator temporarily evacuating a disk partition, performing maintenance on the storage system, then returning the system to its initial state.

This infrastructure workload acts on one of the standby server virtual machines selected in a round-robin fashion from among all the tiles. The virtual machine’s files are moved to the maintenance partition, left there for two minutes, then moved back to their original location. VMmark then waits another two minutes and repeats this process, continuing for the duration of the benchmark period. The number of concurrent storage relocation operations increases with the number of tiles and the number of hosts in the benchmark cluster. The performance metric used is the number of relocations per hour.

Simultaneous Server and Storage Virtual Machine Relocation

The live migration of virtual machines simultaneously across both servers and storage (vMotion without shared storage) allows even more flexibility than either capability alone. This infrastructure workload produces a combination of the infrastructure loads created by the individual operations.

In this infrastructure workload, VMmark uses vMotion to relocate a virtual machine while simultaneously invoking the storage relocation of the same virtual machine's disk files to a maintenance partition. After 2.5 minutes the virtual machine is returned to its original host and the files are returned to their original location. VMmark then waits another 2.5 minutes and repeats the process. This workload models an administrator temporarily evacuating a host and disk partition, performing maintenance on the host and/or storage system, then returning the system to its initial state.

This infrastructure workload acts on one of the DS3WebA virtual machines selected in a round-robin fashion from among all the tiles. The number of concurrent relocation operations increases with the number of tiles and the number of hosts in the benchmark cluster. The performance metric used is the number of relocations per hour.

Automated Load Balancing

Automatically balancing resource demands among multiple physical servers using technology such as VMware's Distributed Resource Scheduler (DRS) has become a fundamental part of modern virtualized datacenters. Intelligently allocating and balancing resources allows the underlying platform to respond effectively to bursty load conditions even when utilizations are high.

VMmark requires that DRS be enabled and running at (or above) a specific level to insure that rebalancing occurs in a timely manner when utilizations are high. This should improve overall performance by addressing load imbalances occurring during the benchmark interval.

VMmark Power Measurement

Virtualization technologies allow multiple application workloads to be consolidated on to relatively few physical servers. This consolidation can result in significant power savings when compared to deploying each application workload natively on a dedicated physical server. The natural fluctuations in computing demands over time that datacenters experience provide additional power saving opportunities.

Many technologies, such as VMware Distributed Power Management (DPM) and various CPU-based power management features, exist to further reduce server power consumption as computing demands vary over time. In addition new and emerging technologies, such as flash-based storage, can have power demands that are markedly different than traditional solutions.

IT architects might wish to consider trade-offs in performance and power consumption when designing datacenters. To this end, VMmark 3.x can measure power usage while running its full application and infrastructure workloads across a multi-host cluster. It provides two different ways to do this: PTDaemon power measurement and ESXTOP power measurement.

This section provides a brief overview of VMmark power measurement. For additional detail, see "VMmark 3.x Power Results" on page 27 and "Using Power Meters With VMmark" on page 84.

PTDaemon Power Measurement

PTDaemon power measurement, introduced in VMmark 2.5, provides for three test types:

- Performance Only
- Server Power-Performance
- Server and Storage Power-Performance

The Standard Performance Evaluation Corporation (SPEC®) has led an industry-wide effort to enable power measurements for a wide range of performance benchmarks. The power measurement capability in VMmark 3.x uses version 1.8.0 of the SPEC PTDaemon (Power Temperature Daemon). The PTDaemon (or PTD) provides a straightforward and reliable building block with support for the many power analyzers that have passed the SPEC Power Analyzer Acceptance Test (see http://www.spec.org/power/docs/SPECpower-Device_List.html for details).

NOTE Though VMmark uses the SPEC PTDaemon, VMmark results are not SPEC metrics and cannot in any manner be compared to SPEC metrics.

ESXTOP Power Measurement

Introduced in VMmark 3.0, VMmark can now automatically collect ESXTOP power data from the servers hosting VMmark application workloads. When enabled, this provides power usage tracking based on the server's power supply sensor.

Although ESXTOP power measurement offers a convenient way to understand VMmark power usage, it cannot be used in published results or compared to PTDaemon metrics.

VMmark Client Systems

Each tile requires a client system. These systems run software that provides loads for most of the workloads. The prime client also runs the VMmark Harness software that can start, stop, and measure the performance of the workloads.

The client systems must run in a vSphere cluster that is not part of the system under test.

NOTE Running the client systems in the same cluster as the SUT might work; however, doing so is unsupported and any results obtained on such a configuration would be non-compliant.

VMmark Harness

The VMmark Harness is a utility run on the prime client system that can start and stop the applications running on the workload virtual machines and can report the results of a test run.

The VMmark Harness is based on the open-source Software Testing Automation Framework (STAF, see http://staf.sourceforge.net/index.php) and its companion execution engine, STAX. These tools support the development and running of distributed coordinated tests across heterogeneous machines and operating systems.

The VMmark Harness consists of several STAX XML modules, the `VMmark3.properties` file, and several workload-specific configuration files. The main STAX module, `vmmark3_main.xml`, processes the `VMmark3.properties` file to configure the test to be run. Each workload has its own `<`*`workload`*`>_functions.xml` module that contains the workload-specific code needed to initialize the test, run the test, and collect the results.

The `VMmark3.properties` file defines the actual test, identifying all the clients and server virtual machines involved in the test, the number of tiles to be run, and the workloads within each tile.

After the `VMmark3.properties` file has been processed, the VMmark Harness performs pre-run system and timing validation and initiates the setup phase for the VMmark infrastructure operations and for each workload in each tile. After the setup has completed, the VMmark Harness simultaneously initiates the individual workloads in all the tiles. When the workload runs have completed, the harness again validates the timing, then collects the results into a results directory.

Contents of VMmark Package

VMmark is distributed as an ova file named `vmmark3.1–template–`*`date`*`.ova` (where *`date`* is the date of release).

Other relevant files are available on the VMmark download page.

VMmark 3.x Benchmark Scoring Methodology

A VMmark 3.x score aggregates the throughput metrics of all application and infrastructure workloads to create a single overall benchmark score that can be used to quickly compare different platform configurations. Every workload must also pass its minimum quality-of-service requirements for the benchmark result to be considered compliant. The scoring methodology is conceptually straightforward despite the underlying complexity and large scope of the benchmark.

After a VMmark Benchmark test run completes, each individual application and infrastructure workload reports its relevant performance metric. These performance metrics are shown in Table 1-2.

Table 1-2. Individual VMmark Workload Metrics

Workload Name	Application(s)	Metric
Standby server	None	None
DS3WebA	Apache, MySQL	Transactions/minute
DS3WebB	Apache, MySQL	Transactions/minute
DS3WebC	Apache, MySQL	Transactions/minute
Weathervane Static	Auction	Operations/second
Weathervane Elastic	Auction	Operations/second
Clone and deploy	None	Deployed VMs/hour
vMotion	None	VM migrations/hour
Storage vMotion®	None	VM migrations/hour
XvMotion®	None	VM migrations/hour
DRS	N/A	None

NOTE The standby server workload does not produce a metric that affects the benchmark score. However the standby server is required to answer to a periodic heartbeat request in order for the VMmark test to be considered compliant. Likewise, DRS must be enabled to allow the platform to automatically balance resources during the benchmark run. DRS does not produce a metric but will affect the performance of other workloads by managing the overall resource allocations to improve performance and provide stability.

The DVD Store implementation in VMmark 3.x is a multi-tier workload containing three web server virtual machines with varying load patterns accessing a single database virtual machine. For scoring purposes, each web server generates a separate results file to be processed independently from the others. This approach provides a clearer understanding of platform behavior.

The Weathervane workload in VMmark 3.x uses two independent instances of the Weathervane Auction application, a static instance and an elastic instance. Because in today's datacenters it's increasingly common to have self-scaling applications that dynamically add and remove resources to meet demands, the elastic instance takes advantage of Weathervane's elasticity-related capabilities to add and remove an application server and a web server throughout the run. This elastic component (along with the cyclical application profile generated by DVD Store 3) allows VMmark 3.x to more accurately represent today's bursty environments.

These metrics are collected at frequent intervals during the course of the run. The standard VMmark 3.x workload is designed to run for at least 3 hours with workload metrics reported every 60 seconds. This means that rather than having a single number upon completion of a test run, the user will have a series of numbers for each of the workloads. The series of data points for each workload is averaged to generate a single score for that workload which is then listed in the VMmark results file (`Score_`*`N`*`_Tile_Test.txt`).

Normalization allows the integration of the different component metrics into an overall score. A reference system is commonly used for normalization when computing benchmark scores. For example, SPEC CPU2000 (see http://www.spec.org/cpu2000/) takes this approach, using a Sun Ultra5_10 with a 300Mhz processor as its reference platform. VMmark 3.x measures consolidation workloads within virtual environments and therefore requires a reference platform capable of successfully running a single tile.

The steady state for the benchmark is defined as the middle two hours of the three-hour run. The first and last half hours are the ramp-up and ramp-down times, respectively. The steady state is further divided into three 40-minute phases. For each of the 40-minute phases we compute the overall result for the platform and select the median score of the three as the reported score.

After a valid run, the metrics of the application workloads within each tile are computed and aggregated into a score for that tile. This aggregation is performed by first normalizing the different performance metrics (such as Actions/minute and operations/minute) with respect to a reference platform. Then a geometric mean of the normalized scores is computed as the final score for the tile. The resulting per-tile scores are then summed to create the application-workload portion of the final metric.

The metrics for the infrastructure workloads are aggregated separately using the same mathematical technique of normalization with respect to a reference platform followed by the computation of the geometric mean. Unlike the application workloads, the infrastructure workloads are not scaled explicitly by the user. Consequently, the infrastructure workloads are compiled as a single group and no multi-tile sums are required.

The final benchmark score is then computed as a weighted average of the application-workload component and the infrastructure-workload component. VMmark 3.x gives weights of 80% to the application-workload component and 20% to the infrastructure-workload component. These weights were chosen to reflect the relative contribution of infrastructure and application workloads to overall resource demands.

The benchmark helps measure the virtualization overheads of the individual workloads as well as the scalability of the entire system. Therefore results for multi-tile runs are reported as the aggregate score for all tiles, the individual scores for each of the tiles, and the scores for the workloads within the tiles as well as the individual scores for each infrastructure workload.

If any of the workloads within any tile fails to run, produces errors during a run, or fails its minimum quality-of-service requirement, that entire VMmark run is considered to be invalid. This applies to programs running on both the servers and the client systems. Also, the configuration of the workloads, the versions of the benchmarks, operating systems, tools, and all other software used must conform to the specifications in the VMmark documentation.

To illustrate the scoring methodology, consider the following two examples. The first example demonstrates how to compute the score for a single-tile benchmark run, while the second example demonstrates how to compute the scores for a multiple-tile benchmark run.

For these examples, assume the reference system had the scores shown in Table 1-3.

Table 1-3. Example Reference System Workload Scores (Artificial Data)

Workload Name	Score
Standby	None
DS3WebA	1000 transactions/minute
DS3WebB	1000 transactions/minute
DS3WebC	1000 transactions/minute
Weathervane Static	1000 operations/second
Weathervane Elastic	1000 operations/second
Clone and deploy	10 deployments/hour
vMotion	10 VM migrations/hour
Storage vMotion	10 VM migrations/hour
XvMotion	10 VM migrations/hour
DRS	None

Example One: Single-Tile Benchmark Scoring

Suppose you perform a single-tile VMmark run and, according to the methodology described earlier, you remove the first and last thirty minutes, calculate results for each of the three 40-minute phases, and select the median of the three results to obtain the scores shown in Table 1-4.

Table 1-4. Single-Tile Example Test System Workload Scores (Artificial Data)

Workload Name	Score
Standby	None
Weathervane Static	940 operations/second
Weathervane Elastic	910 operations/second
DS3WebA	1020 transactions/minute
DS3WebB	980 transactions/minute
DS3WebC	970 transactions/minute
Clone and deploy	9 deployments/hour
vMotion	11 VM migrations/hour
Storage vMotion	9 VM migrations/hour
XvMotion	8 VM migrations/hour
DRS	None

To compute the score for this tile, you first compute each workload's normalized scores by dividing the score for each workload by the reference score for that workload:

Standby:	n/a	n/a
Weathervane Static	940/1000	= 0.94
Weathervane Elastic	910/1000	= 0.91
DS3WebA	1020/1000	= 1.02
DS3WebB	980/1000	= 0.98
DS3WebC	970/1000	= 0.97
Clone and deploy	9/10	= 0.90
vMotion	11/10	= 1.10
Storage vMotion	9/10	= 0.90
XvMotion	8/10	= 0.80
DRS	n/a	n/a

You would then combine the normalized scores for the application workloads using a geometric mean:

(0.94 * 0.91 * 1.02 * 0.98 * 0.97)^(0.2) = 0.96

Next, you would combine the normalized scores for the infrastructure workloads using a geometric mean:

(0.90 * 1.10 * 0.90 * 0.80)^(0.25) = 0.92

Finally, you would combine these two geometric means using a weighted average (80% for application workloads, 20% for infrastructure workloads):

(0.8 * 0.96) + (0.2 * 0.92) = 0.95

The VMmark score for this tile is 0.95. For reporting, the VMmark results file will include this score as well as the individual scores for all of the workloads (both raw and normalized).

Example Two: Multiple-Tile Benchmark Scoring

This example demonstrates the scoring for multiple tiles. The scores for each tile in a multiple-tile run are calculated just as the score for the tile in a single-tile run is calculated. Suppose you perform a VMmark run containing four tiles and obtain the scores shown in Table 1-5.

Table 1-5. Multiple-Tile Example Test System Workload Scores (Artificial Data)

Workload Name	Tile 1 Score	Tile 2 Score	Tile 3 Score	Tile 4 Score
Standby	n/a	n/a	n/a	n/a
Weathervane Static	940	950	890	940
Weathervane Elastic	910	920	915	930
DS3WebA	950	1000	990	970
DS3WebB	940	1010	1020	970
DS3WebC	930	1020	960	1050
Clone and deploy	21			
vMotion	22			
Storage vMotion	11			
XvMotion	26			
DRS	n/a			

The geometric means of the normalized scores for the application workload tiles would be:

Geometric mean of
normalized scores: Tile 1: 0.93 Tile 2: 0.98 Tile 3: 0.95 Tile 4: 0.97

The application workload portion of the VMmark score would be the sum of these geometric means:

0.93 + 0.98 + 0.95 + 0.97 = 3.84

The infrastructure workload portion of the VMmark score would be:

Geometric mean of
normalized scores: 1.91

The overall VMmark score for the system would be the weighted average of these two numbers:

(0.8 * 3.84) + (0.2 * 1.91) = 3.45

Along with the overall VMmark score of 3.45, the VMmark results file will include both the individual tile scores and the workload scores (both raw and normalized).

VMmark 3.x Power Results

VMmark 3.x allows optional measurement of power during VMmark benchmark runs using either of two methods, PTDaemon power measurement and ESXTOP power measurement.

PTDaemon Power Measurement

When power is to be measured using the PTDaemon, either for server only or for both server and storage, the VMmark harness starts the PTDdaemon, which initiates a connection between the PTD client (or clients) specified in the `VMmark3.properties` file and the power meter (or meters) they are configured to monitor. Once the required connections are established and the benchmark run is underway, the harness captures each of the power meter (or meters) results into a single unified data stream. This data, like that from other VMmark workloads, is broken up into various sections (ramp up, three 40-minute steady-state phases, and ramp down).

The reported VMmark 3.x power consumption is the total average watts consumed during the steady-state phase of the benchmark run that resulted in the median score; the total average watts being the sum of the average watts reported by each power meter used in the run. The final VMmark Performance Per Kilowatt (PPKW) score is the VMmark 3.x score divided by the average power consumption in kilowatts.

ESXTOP Power Measurement

When ESXTOP data collection is enabled for the VMmark application workload hosts, VMmark 3.x will parse the power component of ESXTOP data "Power Usage Now Watts," which is sourced from the server's power supply sensor. Enabling ESXTOP power measurement does not require any additional hardware. These power results are reported on a per-host basis and are broken up into various phases of time.

The reported VMmark 3.x ESXTOP power consumption is the total average watts consumed during the median phase of the benchmark run, which is the sum of the average watts reported by each host during the run. The final VMmark Power Efficiency score is the VMmark 3.x score divided by the average power consumption in kilowatts.

NOTE ESXTOP power metrics, including Power Efficiency scores, are not compliant VMmark 3 power measurements and cannot be compared to the VMmark Performance Per Kilowatt metric. ESXTOP power metrics cannot be included in published VMmark results.

Reference Scores

The VMmark 3.x reference scores were obtained using the environment detailed in Table 1-7.

Table 1-6. Reference System Details

Servers	
Servers	Two HP ProLiant DL380 Gen 9
Processors	Two Intel® Xeon® E5-2643 v3 3.4 GHz 6-core processors per server Four processors total
Cores	Six cores per processor 12 cores per server 24 cores total
RAM	256GB per server 512GB total
Network	
Network connectivity	One QLogic 57810 10Gb Ethernet Adapter per server
Network switch	Extreme Networks Summit x670 10 Gb switch
Storage	
Storage connectivity	One QLogic ISP2532 8Gb Fibre HBA per server
Fibre channel switch	EMC Connectrix DS-6510B
Storage array	EMC VNX5700
Storage drives	16 200GB SSD drives configured as two RAID 1/0 LUNs 24 600GB SAS drives configured as two RAID 5 LUNs
Client	
Servers	Dell PowerEdge R930
Processors	Four Intel Xeon E7-8890 v4 2.2GHz 24-core processors
Cores	24 cores per processor 96 cores total
RAM	2TB total
Network connectivity	Intel 82599 10Gb NIC
Storage connectivity	QLogic 2600 HBA
Storage array	Same as system under test
Software	
Server virtualization	VMware ESXi 6.0, build 3620759
Client virtualization	VMware ESXi 6.0, build 3620759
Datacenter management	VMware vSphere 6.5, build 4602587

The reference scores obtained on this system are shown in Table 1-7.

Table 1-7. Reference System Workload Scores

Workload Name	Score
Standby server	None
Weathervane Static	3597.26 operations/second
Weathervane Elastic	572.18 operations/second
DS3WebA	734.32 transactions/minute
DS3WebB	500.51 transactions/minute
DS3WebC	346.80 transactions/minute

Table 1-7. Reference System Workload Scores (Continued)

Workload Name	Score
Clone and deploy	8.0 deployments/Hour
vMotion	26.0 VM migrations/Hour
Storage vMotion	18.0 VM migrations/Hour
XvMotion	18.0 VM migrations/Hour

VMmark Version Notes

For the latest information about a particular VMmark release, including security notices and any changes that occurred after the release of this guide, view the appropriate Release Notes, available at the VMmark download page.

VMmark 3.1

VMmark 3.1 contains updates to improve workload scalability and enhance security as well as including new requirements regarding security vulnerability mitigations. VMmark 3.1 benchmark scores are considered directly comparable to VMmark 3.0 benchmark scores.

VMmark 3.0

VMmark 3.0 provided a completely new, highly-automated benchmark installation process; it used free or open-source software throughout, eliminating the need to purchase any software; it replaced DVD Store 2 with DVD Store 3 and added Weathervane, a new application-level cloud performance benchmark; and it added XvMotion as a new infrastructure workload. The workloads and load levels of VMmark 3.0 were completely changed from VMmark 2.x and the benchmark scores were in no way considered comparable.

VMmark 2.5.2

VMmark 2.5.2 was a minor maintenance release. It included an updated VMmark Benchmarking Guide. VMmark 2.5.2 benchmark scores were considered directly comparable to VMmark 2.0, 2.1, 2.1.1, 2.5, and 2.5.1 benchmark scores.

VMmark 2.5.1

VMmark 2.5.1 was a minor maintenance release. It included an updated VMmark Benchmarking Guide. VMmark 2.5.1 benchmark scores were considered directly comparable to VMmark 2.0, 2.1, 2.1.1, and 2.5 benchmark scores.

VMmark 2.5

VMmark 2.5 added support for optional power monitoring, support for client systems running additional versions of Windows Server 2008, support for the VMware vCenter Server® Appliance™, a new Turbo Mode, message and results delivery via Growl/Prowl, and a variety of other updates and improvements. VMmark 2.5 benchmark scores were considered directly comparable to VMmark 2.0, 2.1, and 2.1.1 benchmark scores.

VMmark 2.1.1

VMmark 2.1.1 was a minor maintenance release. VMmark 2.1.1 benchmark scores were considered directly comparable to VMmark 2.0 and 2.1 benchmark scores.

VMmark 2.1

VMmark 2.1 added support for client systems running certain versions of Windows Server 2008 (in addition to Windows Server 2003, which was supported in VMmark 2.0) as well as support for virtualized clients (subject to certain conditions). The benchmark harness now scales the storage relocation workload in the same fashion as the other infrastructure workloads. The *VMmark Benchmarking Guide* is updated to reflect these changes. VMmark 2.1 benchmark scores were considered directly comparable to VMmark 2.0 benchmark scores.

VMmark 2.0

While VMmark 1.x was designed as a single-system consolidation benchmark consisting of six isolated single-tier workloads, VMmark 2.0 was designed as a multi-host benchmark reflecting typical, modern-day usage of virtualized infrastructure. VMmark 2.0 consisted of two single-tier application workloads, two multi-tier application workloads, and four infrastructure-level workloads. The workloads and load levels of VMmark 2.0 were completely changed from VMmark 1.x and the benchmark scores were in no way considered comparable.

VMmark 1.1.1

VMmark 1.1.1 was a minor maintenance release. It included updates to the *VMmark Benchmarking Guide*, updated reporting script and disclosure template, and new versions of the Run and Reporting Rules and Review Panel Guidelines and Procedures. VMmark 1.1.1 benchmark scores were considered directly comparable to VMmark 1.0 and 1.1 benchmark scores.

VMmark 1.1

In order to reflect the increasing use of 64-bit operating systems in data centers, VMmark 1.1 used 64-bit operating systems and applications in three of the workload virtual machines (Java server, Web server, and Database server). The workloads and load levels were unchanged from VMmark 1.0, however, and the VMmark 1.0 and 1.1 benchmark scores were thus considered directly comparable.

VMmark 1.0

This was the initial release of Vmmark.

VMmark Run and Reporting Rules

In addition to the guidelines presented in this User's Guide, specific run and reporting rules must be followed for a VMmark test to be considered compliant and for the results of that test to be publishable. These run and reporting rules are available at the VMmark web page:
http://www.vmware.com/products/vmmark.html

NOTE VMware updates the VMmark run and reporting rules from time to time. Before performing a VMmark test, you should confirm that you have the latest available version.

VMmark Benchmark Requirements 2

This chapter describes the hardware and software required in order to perform VMmark benchmark testing. It consists of the following sections:

- "VMmark Version and Settings Requirements" on page 33
- "VMmark Hardware Requirements" on page 33

VMmark Version and Settings Requirements

In order for a VMmark benchmark run to be compliant, the following version and settings requirements must be met:

- Both VMware vCenter® and VMware ESXi™ versions must be currently available or planned to be generally available within 90 days.
- The systems under test must be running ESXi version 6.0 or later.
- The DRS aggressiveness level must be set to the fourth or fifth position (counting from the left). These two positions represent the second-most aggressive setting and the most aggressive setting, respectively.
- There must be no DRS rules set.

VMmark Hardware Requirements

This section describes the hardware required in order to run VMmark.

System Under Test Hardware Requirements

The system under test (SUT) must use server hardware supported by the ESXi version being used and must be configured to meet or exceed the minimum CPU, minimum memory, and other basic requirements for that version. The subsections below provide information about the required storage, memory, and other elements of the test environment.

System Under Test CPU Requirements

Each tile in the VMmark benchmark uses a total of 47 vCPUs (the vCPUs required by the prime and tile clients are not included here because they're not part of the SUT; instead they're addressed in "CPU Requirements for VMmark Clients" on page 35). While CPU overcommitment is allowed, it does have the potential to adversely affect the VMmark score.

System Under Test Memory Requirements

Each tile in the VMmark benchmark allocates a total of either 158GB or 166GB of RAM (the RAM required by the prime and tile clients is not included in this total because it's not part of the SUT; instead it's addressed in "Memory Requirements for VMmark Clients" on page 35). While memory overcommitment is allowed, it does have the potential to adversely affect the VMmark score.

NOTE The first VMmark tile allocates 166GB; subsequent tiles allocate 158GB. Depending on the number of tiles and hosts, one or more additional deploy VMs might be required, each of which allocates 8GB. See "Calculate the Number of Simultaneous Infrastructure Operations" on page 45 for more detail.

System Under Test Storage Requirements

Configure the system under test with enough shared datastore capacity to hold the disks and paging files for all the virtual machines required for the VMmark Benchmark runs. This is approximately 891GB per tile (not counting the storage space required by the prime and tile clients, addressed in "Storage Requirements for VMmark Clients" on page 35).

NOTE In order to provide source and target datastores for the storage relocation operations that are part of the benchmark, VMmark requires a minimum of two datastore partitions.

If using VMware vSAN™ as the primary storage solution, a secondary storage solution will be needed for infrastructure operations.

Additionally, the benchmark requires that all ESXi hosts used in a test have access to the same shared storage.

The VMmark benchmark needs high-throughput, low-latency storage. While the exact bandwidth requirements will vary based on other aspects of the environment, a single VMmark tile can drive about 3500 IOPS (Input/Output Operations Per Second), with additional tiles typically each driving somewhat less. The latency requirements will also vary based on other aspects of the environment; a review of published VMmark results will provide a sense of the storage solutions that work with VMmark. Thus in addition to ensuring that you have enough storage *capacity*, you should also make sure your storage system will have adequate performance.

VMmark can be used with any storage hardware type that meets the above requirements.

Other System Under Test Requirements

In order to support the vMotion and DRS infrastructure operations included in VMmark 3.x, the benchmark requires a minimum of two ESXi hosts and requires that all ESXi hosts used in a test be vMotion compatible.

Client Hardware Requirements

Each VMmark tile requires its own tile client virtual machine and each VMmark deployment requires a single prime client virtual machine. These client virtual machines must meet or exceed the specifications detailed in this section and in the subsections below.

- The client virtual machines should not use resources that are part of the system being tested. This means they should not be run on the same ESXi servers as the benchmark workloads and ideally should use separate storage resources.
- All virtual clients in a VMmark deployment must have identical resource, scheduling, and tuning configurations.
- The servers hosting the VMmark client virtual machines must be running VMware ESXi 6.0 or later.
- The servers hosting the VMmark client virtual machines must be enterprise-class hardware.

CPU Requirements for VMmark Clients

- The CPUs in the servers hosting the virtual clients must not be overcommitted.

  NOTE This means the sum of the vCPUs configured for all the virtual machines running on a host must not exceed the number of logical CPUs that host contains.

  Each hyper-threaded core can be considered two CPUs when calculating the level of CPU commitment.

- In order not to affect the VMmark score, we *recommend* that the hosts' physical processors be equivalent to or faster than Intel Xeon E5-2699, v3 ("Haswell")

  NOTE As a very rough approximation, client systems will need about one-quarter the processing power per tile as the systems under test. In addition, compute resources should be allocated for the single prime client.

- If desired, users can increase the vCPU count for the prime client.
- If desired, users can increase the vCPU count for the tile clients (provided all tile clients are provisioned identically).

Memory Requirements for VMmark Clients

- The prime client requires 16GB of virtual RAM and the tile clients each require 20GB of virtual RAM.
- The memory in the servers hosting the virtual clients must not be overcommitted.

  NOTE This means the sum of the memory allocated to all the virtual machines running on a host must not exceed the amount of physical memory that host contains.

- If desired, users can increase the RAM for the prime client.
- If desired, users can increase the RAM for the tile clients (provided all tile clients are provisioned identically).

Storage Requirements for VMmark Clients

- In typical deployments, the prime client requires 236GB of storage for its disks and paging files. In environments with a large number of ESXi hosts, the prime client can require more storage space.
- Each tile client requires 36GB of storage for its disks and paging files.
- The physical storage used for the virtual clients should be typical of a modern datacenter and should offer performance sufficient to meet the client virtual machines' resource requirements without introducing delays that might affect the benchmark results.

vCenter Hardware Requirements

The vCenter Server can be run on either a physical system or a virtual machine. In either case the underlying system (whether physical or virtual) must be supported by the vCenter version being used and must be configured to meet or exceed the minimum CPU, minimum memory, and other basic requirements for that product and version.

NOTE If the vCenter Server is run on a virtual machine, that virtual machine must not be on a host that is part of the systems under test cluster.

Network Requirements

VMmark tests require a network between the server system and the client systems as well as a separate dedicated vMotion network for infrastructure operations. Some configurations might need multiple network links for optimal performance.

We recommend that the VMmark systems be on dedicated private networks and that the server and client systems be connected with links of at least 1Gb/s speed.

For benchmark runs, IP addresses should be statically assigned. There must be no DHCP server reachable from any of the workload virtual machines.

NOTE Although this User's Guide describes how to configure VMmark to use DHCP, such deployments can not be used to generate compliant benchmark runs.

Network Performance

The physical network infrastructure should be typical of a modern datacenter and should offer performance sufficient to meet the virtual machines' resource requirements without introducing delays that might affect the benchmark results.

Network bandwidth requirements can vary significantly with different environments, but each VMmark tile can potentially consume 600-650 Mb/s of network bandwidth.

Network Topology

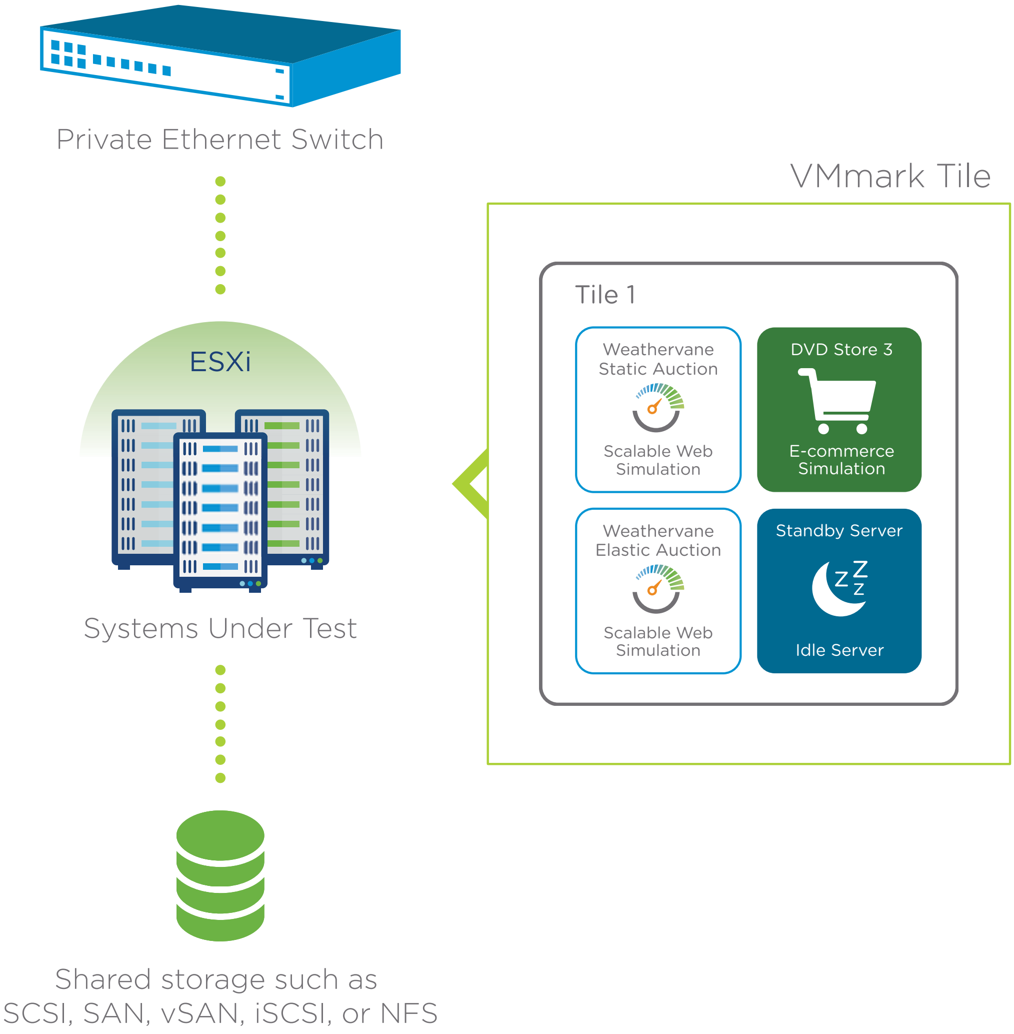
For the most accurate benchmark results, we recommend that you connect the client and server systems through switches that are dedicated to the benchmark tests and are not shared with any other systems.

Additionally, though it is often useful for the VMmark test systems to be connected to a company-wide network during setup, the most accurate benchmark results can be obtained when neither the client systems nor the server systems have network connections to any other network, whether an internal company network or the external Internet, during the VMmark tests. Use of a completely private network ensures that extraneous network traffic does not impact the VMmark tests.

For compliant benchmark runs, configure the ESXi servers, each workload virtual machine, and each client virtual machine with a static IP address. (While you can run VMmark with DHCP-assigned IP addresses, such deployments can not be used to generate compliant benchmark runs.)

Figure 2-1 illustrates a sample network configuration.

Figure 2-1. Sample VMmark Benchmark Network Configuration (Not Shown: vCenter Server, Clients, Dedicated vMotion Network)



Power Metering Requirements

VMmark 3.x tests can optionally include PTDaemon power measurements. Performing such measurements requires one or more compatible power meters, and imposes some additional requirements on the benchmarking environment. This section addresses these points.

Power Meter Requirements

For VMmark tests that will include server power measurement, you will need one or more compatible power meters for the servers in the test environment. For VMmark tests that will include server and storage power measurement, you will also need one or more compatible power meters for the storage device (or devices) in the test environment.

VMmark is compatible with any power meter supported by SPEC. A list of these meters can be found on the SPEC web site:
http://www.spec.org/power/docs/SPECpower-Device_List.html

While VMmark should work with any of these meters, testing has only been performed with the Yokogawa WT210 power meter.

Each power meter used in a VMmark test must meet the following requirements:

- It must be supported by the PTDaemon version used in the VMmark test.
- It must have been calibrated by a standard traceable to NIST (http://nist.gov) if in the USA or to a counterpart national metrology institute if in another country.
- It must have been calibrated within the time period in which its accuracy is guaranteed (typically one year, but can be less for some meters) and in no event more than one year prior to the date of the VMmark test.
- Any Restrictions on Use for the meter must be followed.

Each power meter must be connected to a physical client system. This can be the prime client, a client associated with a VMmark tile, or even another physical system.

NOTE Depending on the connection type used by the chosen power meters, the number of physical ports on a client system might become a bottleneck. For example, many server systems have only one RS-232 serial port. With a power meter that uses an RS-232 connection, such as the Yokogawa WT210, this would allow only one such meter to be directly connected to each client.

Numerous third-party products are available, however, to allow one or more serial devices, such as power meters, to be connected to a single USB port. While we expect that most such devices could be used to connect power meters to VMmark clients, we have performed only limited testing.

Environmental Requirements for Power Measurement

To produce a compliant VMmark 3.x benchmarking result that includes power measurement, the following additional requirements must be met:

Temperature:

- The test environment must be at a temperature of 20°C or higher during the entire test run.

Input power requirements:

- The AC power supplied to the test environment must be between 95 and 126 volts or between 198 and 241 volts.
- The AC power supplied to the test environment must have a frequency of either 50Hz ± 1% or 60Hz ± 1%.

NOTE An uninterruptible power supply (UPS) is allowed as the input power source as long as the UPS outputs a pure sine wave. If a UPS is used, it should be placed between the utility company power and the power meter and its usage must be specified in the notes section of the full disclosure report.

Quality of power readings:

- The results reported by the power meter must be within the accuracy guidelines dictated by the SPEC PTD driver.

Components to be Measured

For VMmark tests that will include server power measurement, the power consumption of the following components must be measured:

- All servers running VMmark workload virtual machines.

For VMmark tests that will include server and storage power measurement, the power consumption of the following components must be measured:

- All servers running VMmark workload virtual machines.
- All external storage devices accessed by the VMmark workload virtual machines.
- All other hardware necessary to connect the servers to the storage subsystems. This includes fibre-channel switches (in the case of SAN storage) or network switches (in the case of NAS storage).

Preparing the Infrastructure for VMmark 3.1 Benchmark Tests

3

This chapter describes the steps required in order to prepare the infrastructure for VMmark 3.1 tests. It consists of the following sections:

Create the Virtualization Infrastructure

The following subsections describe the steps needed to prepare the virtualization infrastructure for VMmark 3.1 tests.

NOTE VMmark, and the benchmark hardware and software components (vCenter, ESXi, and the VMmark template, clients, workload virtual machines, systems under test, storage, etc.) must be used only in an isolated test environment. None of the benchmark hardware or software components should ever be connected to a production network or transitioned to a production environment.

Install vSphere vCenter

Install a version of VMware vSphere vCenter compatible with VMmark 3.1.

VMmark 3.1 is designed to run with vCenter 6.0 or later. In order for a VMmark benchmarking run to be compliant, the installed version of vCenter must either be a released, generally available version, or must meet the pre-release requirements detailed in the VMmark Run and Reporting Rules. Make sure that all relevant updates and patches are installed.

NOTE All ESXi hosts should be able to reach the same shared storage in order to enable vMotion and Storage vMotion infrastructure operations.

NOTE Beginning with vSphere 6.7 Update 1, the HTML5-based vSphere Client is fully featured and is the preferred interface. The instructions in this guide are therefore based on the version 6.7U1 vSphere Client. For the best functionality we recommend using that client version rather than any Flash/Flex-based version (called the "vSphere Web Client").

Install and Configure VMware ESXi

Install and configure VMware ESXi as described in the subsections below.

Install ESXi

Install VMware ESXi version 6.0 or later according to VMware instructions on all servers to be used in the test. The ESXi version must either be a released, generally available version, or must meet the pre-release requirements detailed in the VMmark Run and Reporting Rules. Make sure that all relevant updates and patches are installed.

Make sure that hardware on which you are installing ESXi meets the requirements outlined in "System Under Test Hardware Requirements" on page 33, including the datastore capacity and vMotion compatibility.

Enable ESXi Shell and SSH Access

Enable ESXi Shell and SSH access on your ESXi host systems by performing the following steps on every ESXi system that will be used in the VMmark environment:

1. From the vSphere Client, select an ESXi host in the Inventory pane.
2. Click the **Configure** tab.
3. Under **System**, select **Services**.
4. Configure ESXi Shell to start and stop with the host:
   a. Right click the **Startup Policy** entry for **ESXi Shell**.
   b. Select **Edit Startup Policy**.
   c. Select **Start and stop with host**.
   d. Click **OK**.
5. Configure SSH to start and stop with the host:
   a. Right click the **Startup Policy** entry for **SSH**.
   b. Select **Edit Startup Policy**.
   c. Select **Start and stop with host**.
   d. Click **OK**.

Configure vCenter

1. Using the vSphere Client, create a new datacenter on your vCenter server (in the **Hosts and Clusters** pane right-click the vCenter server, select **New Datacenter...**, enter a name, then click **OK**).

   NOTE If your vCenter server already has a datacenter, you can optionally use that one instead.

2. Create a VMmark systems under test cluster in the datacenter as follows:
   a. Still in the **Hosts and Clusters** view, right-click on the new datacenter and select **New Cluster...**.
   b. At the **New Cluster** window:
      i. Enter a name for the cluster (recording the name for later use; this will be `vCServerVMmark3Cluster` in the `VMmark3.properties` file).

         NOTE Each cluster name should be unique, even across datacenters.

      ii. Next to **DRS** click the box labeled **Turn ON**.
      iii. For **Automation Level** ensure that **Fully automated** is selected.
      iv. For **Migration threshold**, move the slider to the fourth position (counting from the left); this is the second-most aggressive setting.

         NOTE Those wishing to further increase the DRS aggressiveness can, optionally, set the **Migration threshold** slider to the fifth (rightmost) position, which is the most aggressive setting.

   v. Choose the **EVC** setting most appropriate for the hosts in the cluster.

   vi. Choose **Virtual SAN** if appropriate.

   vii. Click **OK**.

3. Add hosts to your systems under test cluster by right clicking on the cluster name and selecting **Add Host...**.

4. In order to enable the vMotion and Storage vMotion infrastructure operations, make sure that all the ESXi hosts in your systems under test cluster can reach the same shared storage.

   NOTE We recommend that the ESXi hosts not be connected to any storage resources that aren't part of the same VMmark installation. This is to avoid potential problems caused by identically named virtual machines on different VMmark installations.

5. Make sure that all hosts can vMotion to all other hosts within the systems under test cluster.

6. Create a VMmark client cluster in the datacenter (recording the name for later use; this will be `vCServerClientCluster` in the `VMmark3.properties` file) and add one or more ESXi hosts to it.

   NOTE Optionally, you can enable DRS for this client cluster.

Configure Time Synchronization on the ESXi Hosts and the vCenter Server

The time on the ESXi hosts and the vCenter Server must be synchronized to a common NTP server. (Client and workload virtual machines should use VMware Tools to set their clocks from their ESXi host.)

NOTE In addition:

- The vCenter Server and the ESXi hosts (as well as the workload virtual machines and the clients, both described later) must all be set to the USA standard date and time format (that is, month/day/year) as opposed to the UK standard (that is, day/month/year).
- The vCenter Server, the workload virtual machines (described later), and the clients (described later) must all be set to the same time zone.

The following sections describe how to configure this time synchronization.

Configure Time Synchronization for ESXi Hosts

To use the vSphere Client to configure ESXi hosts to synchronize with an external NTP server, follow the instructions at this link:

http://kb.vmware.com/kb/57147

To confirm that NTP is working correctly, run the following command from each ESXi host:

```
ntpq -p
```

If NTP is working correctly on the ESXi host, you'll see something similar to the following:

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*118.163.81.61   192.168.0.3      2 u  841 1024  377    1.967    0.414   0.362
```

If NTP is *not* working correctly on the ESXi host, you'll see something more like this:

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
193.168.1.100    .INIT.          16 u    - 1024    0    0.000    0.000   0.000
```

Indications of trouble include a "-" in the "when" column and zeros in many of the other columns.

If needed, you can find more information here:

https://kb.vmware.com/s/article/1005092

Configure Time Synchronization for vCenter Server Appliances

To configure vCenter Server Appliances to synchronize with an external NTP server, follow the instructions at this link:

http://kb.vmware.com/kb/2113610

Configure Time Synchronization for Windows vCenter Servers

To configure Windows-based vCenter Servers to synchronize with an external NTP server, follow these steps:

1 Open the Windows Registry editor (**Start Menu** > **Run** > **Regedit**).

2 Navigate to and expand the following registry folder:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\W32Time

3 Referring to Table 3-1, modify each of the registry entries as follows:

   a Navigate to the entry specified in the path column of Table 3-1.

   b Right-click the entry and select **Modify**.

   c If the Type in Table 3-1 is **REG_DWORD**, select the **Decimal** radio button.

   d Enter the new data from Table 3-1.

   e Click **OK**.

   f Repeat for each entry in Table 3-1.

Table 3-1. Registry Entries to Change

Path	Type	Data
Parameters\Type	REG_SZ	NTP
Config\AnnounceFlags	REG_DWORD	5
TimeProviders\NtpServer\Enabled	REG_DWORD	1

4 Modify the NTP server name:

   a Still within **W32Time**, navigate to **Parameters**.

   b Right click on **NtpServer**.

   c Select **Modify**.

   d Enter the DNS name or IP address of your NTP server. If you enter a DNS name (rather than an IP address) you must append `,0x1` to the end.

   NOTE Use the same NTP server to synchronize your vCenter Server system and all your hosts.

5 Referring to Table 3-2, modify each of the registry entries as follows:

   a Navigate to the entry specified in the path column of Table 3-2.

   b Right-click the entry and select **Modify**.

   c Select the **Decimal** radio button.

   d Enter the new data from Table 3-2.

   e Click **OK**.

f Repeat for each new entry in Table 3-2.

Table 3-2. Registry Entries to Change

Path	Type	Data
TimeProviders\NtpClient\SpecialPollInterval	REG_DWORD	900
Config\MaxPosPhaseCorrection	REG_DWORD	3600
Config\MaxNegPhaseCorrection	REG_DWORD	3600

6 Close the registry editor.

7 Restart the time service by entering the following at a Command Prompt:

```
net stop w32time && net start w32time
```

Calculate the Number of Simultaneous Infrastructure Operations

In any VMmark run, a variety of infrastructure operations are performed. These are clone and deploy, vMotion, Storage vMotion, and XvMotion.

In addition to these baseline infrastructure operations, additional instances of these infrastructure operations are performed in larger environments. If your VMmark system under test cluster contains four or more hosts and you will be running four or more tiles, VMmark will perform these additional instances of infrastructure operations. The number of simultaneous infrastructure operations is half of the smaller of either the number of hosts or the number of tiles, rounded down to an integer; that is:

floor(min(#hosts in VMmark cluster, #tiles)/2)

Calculate this number, then record it for use later in the preparation process. (This number will determine how many copies of the template you'll need to make and how many instances will be required to be entered in the `DeployVMinfo` parameter in the `VMmark3.properties` file.)

Non-VMmark Virtual Machines in a VMmark Cluster

Some environments might require one or more non-VMmark virtual machines to run in the VMmark systems under test cluster. The presence of such non-VMmark virtual machines will generate a warning when a VMmark run is initiated. Such virtual machines must have been approved by VMware in order for a benchmark run to be compliant.

For questions about non-VMmark virtual machines in the VMmark systems under test cluster, contact VMware at `vmmark-info@vmware.com`.

Obtain the VMmark Template and Create the Prime Client

VMmark 3.x uses a single template as the source for all client and workload virtual machines used in the benchmark. This section describes obtaining that template and creating the first client (the “prime client”).

NOTE VMmark, and the benchmark hardware and software components (vCenter, ESXi, and the VMmark template, clients, workload virtual machines, systems under test, storage, etc.) must be used only in an isolated test environment. None of the benchmark hardware or software components should ever be connected to a production network or transitioned to a production environment.

Obtain the VMmark Template

The VMmark template can be obtained using the download link on the VMmark pages of the VMware website:

http://www.vmware.com/products/vmmark.html

Deploy the VMmark Template

Once you’ve obtained the `.ova` format VMmark template (`vmmark3-template*`), follow these steps to deploy it to your VMmark systems under test cluster:

1 From the vSphere Client, in the inventory pane, select **Hosts and Clusters**.

2 If needed, expand the tree on the left hand side to show your clusters.

3 Select your VMmark systems under test cluster.

4 Select **Actions** on the top bar, then select **Deploy OVF Template**.

NOTE If you have not yet installed the VMware Client Integration Plugin, you will need to do so. For details, see http://www.vmware.com/support/pubs/

5 Under **Select template**, click **Local file**, browse to select the .ova template file, then click **Next**.

6 Under **Select name and location**, leave the **Name:** field at the default, select the folder to which you’d like to deploy the template, then click **Next**.

NOTE For VMmark runs, the VMmark 3 template must be placed in the systems under test cluster.

7 Under **Select a resource**, select where you’d like to run the template, then click **Next**.

8 Under **Review details**, click **Next**.

9 Under **Select storage**, change the virtual disk format if desired, select the datastore where you’d like to deploy the .ova, then click **Next**.

NOTE The disk type (that is, Thin, Thick Lazy, Thick Eager) of the *primary* disk for a provisioned VM will be equivalent to the selection used for the template.

The disk type (that is, Thin, Thick Lazy, Thick Eager) of the *secondary* disk for a provisioned VM is controlled by the value of the `ProvisioningSecondaryDiskType` parameter in the `VMmark3.properties` file.

10 Under **Select networks**, select your desired network then click **Next**.

11 Under **Ready to complete**, click **Finish**.
The OVA will take a few minutes to deploy.

NOTE Though this virtual machine is referred to as the “VMmark template,” it should be left as a virtual machine as opposed to converting it to a template in vCenter.

12 Enable VMware Tools time synchronization:

   a From the vSphere Client, right-click on the newly-deployed VMmark template and select **Edit Settings...**.

   b Select the **VM Options** tab.

   c Expand **VMware Tools**.

   d Place a mark in the check box for the **Synchronize guest time with host** option.

   e Click **OK**.

Make Copies of the VMmark Template for Infrastructure Operations

Once you've deployed the VMmark template (`vmmark3-template*`), create enough copies for all the simultaneous infrastructure operations your environment will need (as calculated in "Calculate the Number of Simultaneous Infrastructure Operations" on page 45), naming the copies with the template name, a hyphen, and a sequential number. For example, if the template is named `vmmark3-template-`*`date`*, you'd name these copies `vmmark3-template-`*`date`*`-02`, `vmmark3-template-`*`date`*`-03`, `vmmark3-template-`*`date`*`-04`, and so on.

Note the names of these copies for later use, to be entered under `Deploy/Templates` when you edit the `VMmark3.properties` file.

Create the Prime Client

Follow these steps to clone the VMmark template to a new prime client:

1 From the vSphere Client, right click on the newly-deployed vmmark3-template* virtual machine and select **Clone** > **Clone to Virtual Machine...**.

2 Under **Select a name and folder** enter **`PrimeClient`**, select the location for the prime client, then click **Next**.

3 Under **Select a compute resource**, expand the tree and select your VMmark client cluster (and, if DRS is not enabled for the client cluster, select an ESXi host), then click **Next**.

4 Under **Select storage**, select your desired datastore to provision the prime client, then click **Next**.

   NOTE You can also change the virtual disk format and/or the VM storage policy if desired.

5 Under **Select clone options**, don't select any options (remove the check from **Customize the virtual machine's hardware** if it's checked), then click **Next**.

6 Under **Ready to complete**, select **Finish**.

7 Once the clone process is complete, edit the prime client's settings as follows:

   a Right click on the prime client virtual machine and select **Edit Settings...**.

   b On the **Virtual Hardware** tab, set **CPU** to **8** and **Memory** to **16GB**.

   c Add a second disk:
   Click the **ADD NEW DEVICE** button (at the top), select **Hard Disk**, and set the size to **200GB**.

      NOTE You can select another size for this second disk, but at least 200GB is recommended. More space might be required for environments with a large number of ESXi hosts.

d If you wish to add a second NIC, do so now:
Click the **ADD NEW DEVICE** button, select Network Adapter, and select the desired network for the new NIC.

NOTE The VMmark benchmark should be run on an isolated network. Though the prime client has one NIC by default, in most VMmark deployments the prime client is configured with a second NIC. The first NIC is typically used for external access (usually on the **`VM Network`** network); the second NIC is typically used to communicate with the environment under test over the isolated network (usually on the **`testbed`** network).

e Click **OK**.

Configure the Prime Client

Once the prime client is created, follow these steps to configure it.

NOTE This document is written for environments with two prime client NICs (as described above). Configuring the prime client with just one NIC is fine, as long as the test environment is still isolated.

1 Right click on PrimeClient and select **Power** > **Power On**.

2 Right click on **PrimeClient** and select **Open Console**.

3 Within the console, log in as **`root`**. (The default root password for the template virtual machines is **`vmmark`**.)

4 Configure the network settings for nic0:

NOTE Although this User's Guide describes how to configure VMmark to use DHCP, such deployments can not be used to generate compliant benchmark runs.

a In a terminal window on the prime client, run the command `ifconfig` to determine which network is active.
(The network name will be of the form **enoXXXXXXXX**, typically `eno16777732`.)

b Change to the network scripts directory:
```
cd /etc/sysconfig/network-scripts
```

c Copy the appropriate network script (`ifcfg-sample-static` for networks with static IP address assignment, `ifcfg-sample-dhcp` for networks with DHCP IP address assignment) to be used by the active network identified in Step a above.
For example, to configure a network named eno16777732 for use with a static IP address:
```
cp ~/VMmark3/samples/ifcfg-sample-static ifcfg-eno16777732
```

d Open the network script in an editor.
For example, for a network named eno16777732:
```
vim ifcfg-eno16777732
```

e Update the name, the device, the IP address, and other network settings as needed (i.e., IPADDR, NETMASK, and GATEWAY).

5 If you added a second NIC, configure it now:

a In a terminal window, run the command `ifconfig` to determine which network is active.
(The network name will be of the form **enoXXXXXXXX**, typically `eno16777732`.)

b Run the command `ls /sys/class/net` to determine the name of the new network.
(The new network will be the network name that *isn't* the active one you just identified.)

c Make sure you're still in the `/etc/sysconfig/network-scripts` directory, then copy the appropriate network script (`ifcfg-sample-static` for networks with static IP address assignment, `ifcfg-sample-dhcp` for networks with DHCP IP address assignment) to be used by the new network identified in Step b above.
For example, to configure a new network named eno33559296 for use with a static IP address:

```
cp ~/VMmark3/samples/ifcfg-sample-static ifcfg-eno33559296
```

NOTE Although this User's Guide describes how to configure VMmark to use DHCP, such deployments can not be used to generate compliant benchmark runs.

d Open the new network script in an editor.
For example, for a network named eno33559296:

```
vim ifcfg-eno33559296
```

e Update the name, the device, the IP address, and other IP settings as needed (i.e., IPADDR, NETMASK, and GATEWAY).

f Run the following command:

```
systemctl restart network
```

6 Still in the terminal window, convert this virtual machine into the prime client by running the make-prime script:

```
cd ~/VMmark3/tools
sh make-prime.sh
```

The make-prime script will start an X Windows GUI and display a login dialog.

7 Log in as **`root`**.
(The default login is **test**, but click **Not listed?** and log in as user: **`root`**, password: **`vmmark`**.)

8 Configure passwordless SSH in the VMware ESXi hosts to allow the prime client to log into them without a password:

a From the prime client, copy `id_rsa.pub` to each ESXi system:

```
cd ~/.ssh/
scp id_rsa.pub root@ESXisystem:/id_rsa-client.pub
```

(where *`ESXisystem`* is the name of an ESXi system).

NOTE This assumes you've enabled ESXi Shell access on your ESXi hosts systems, as described in "Enable ESXi Shell and SSH Access" on page 42.

b Still on the prime client, for each ESXi system, add the new key to the authorized-keys list:

```
ssh root@ESXisystem “cat /id_rsa-client.pub >> /etc/ssh/keys-root/authorized_keys”
```

9 Make sure the prime client is configured for the correct time zone:

NOTE The prime client, tile clients, and all workload virtual machines must be set to the same time zone. You can set the tile clients' and workload virtual machines time zone during tile provisioning described below.

a In a terminal window, run the `date` command to determine the prime client's current time zone. If the time zone is correct for your environment, skip to Step 10.

b Make a backup copy of the default `localtime` file:

```
mv /etc/localtime /etc/localtime.bak
```

c Under `/usr/share/zoneinfo`, locate the correct time zone file for your environment.

d Link to the correct time zone file:

```
ln -s /usr/share/zoneinfo/US/Central /etc/localtime
```

(replacing *`US/Central`* with the correct file for your environment).

10 Reboot the prime client.

11 You can now use VNC or the vSphere Client console to connect to this newly-created prime client (using the IP address for the newly-added network adapter). The default login is **test**, but click **Not listed?** and login as user: `root`, password: `vmmark`.

Optional Configuration Changes for the Prime Client

In order to keep your VMmark environment functional, and to remain compliant with the VMmark Run and Reporting Rules, make changes to the prime client (or to the tile clients or workloads) only if they are explicitly allowed in this User's Guide or in the VMmark Run and Reporting Rules. Check with VMware before making any other changes.

Changes that could cause problems include, but are not limited to, installing, updating, or removing software.

If desired, make one or more of the following optional configuration changes to the prime client:

- Optionally, set the resolution:
  Applications > **System Tools** > **Settings** > **Displays**
  (For example, 1680x1050 16:10)

- Optionally, disable screen lock:

  a **Applications** > **System Tools** > **Settings** > **Privacy** > **Screen Lock**.

  b Change **Screen Lock** to **OFF**.

- Optionally, add a keyboard shortcut to minimize all windows:

  a **Applications** > **System Tools** > **Settings** > **Keyboard** > **Shortcuts** > **Navigation**.

  b Click **Hide all normal windows** so that the second column changes to **New accelerator...**.

  c Key in your desired keyboard shortcut; the key combination will appear in the second column.

  d Close the window.

Provision VMmark Tiles

Once the prime client is created and configured, it is used to create all other clients and the workload virtual machines. The subsections below describe this process.

Creating the First VMmark Tile (Tile 0)

Follow these steps to create the first VMmark tile (Tile 0):

1 Use SSH to connect to the prime client, or use VNC or the vSphere Client console and open a terminal window using the graphical user interface.
(The default login is **test**, but click **Not listed?** and log in as user: **`root`**, password: **`vmmark`**.)

2 CD to `~/VMmark3`.

3 Modify the following parameters in the `VMmark3.properties` file according to your environment and needs:

NOTE Also see "VMmark3.properties File Edits Required for First VMmark Provisioning" on page 56 for additional information about editing the `VMmark3.properties` file.

a The various email settings, allowing you to receive email notifications regarding VMmark runs.

b The various `vCServer` settings in the `vSphere Settings` section of the file.

c `ProvisioningSource = <vmmark3-template*>`
(Replace *`<vmmark3-template*>`* with the full template name.)

d
```
ProvisioningDatastores = \
Client:DatastoreName,\
DS3WebA:DatastoreName,\
DS3WebB:DatastoreName,\
DS3WebC:DatastoreName,\
DS3DB:DatastoreName,\
AuctionLB:DatastoreName,\
AuctionMSQ:DatastoreName,\
AuctionWebA:DatastoreName,\
AuctionWebB:DatastoreName,\
AuctionAppA:DatastoreName,\
AuctionAppB:DatastoreName,\
AuctionNoSQL:DatastoreName,\
AuctionDB:DatastoreName,\
ElasticLB:DatastoreName,\
ElasticWebA:DatastoreName,\
ElasticWebB:DatastoreName,\
ElasticAppA:DatastoreName,\
ElasticAppB:DatastoreName,\
ElasticDB:DatastoreName,\
Standby:DatastoreName
```
(Replace each occurence of *`DatastoreName`* in the above example with the datastore onto which that virtual machine should be deployed.)

e Modify any other parameters as needed for your environment. These could include parameters such as `ProvisioningTimeZone`, `ProvisioningNetworkLabel`, and some of the parameters under `Additional Network Settings` (such as `ProvisioningIPstaticStart`, `ProvisioningGateway`, `ProvisioningSubnetMask`, `ProvisioningDnsServerList`, and `ProvisioningDnsSuffixList`).

NOTE Changing the `ProvisioningTimeZone` parameter from the default VMmark time zone of `US/Pacific` to your local time zone (if different) allows you to see the VMmark results in your local time zone.

However, if you change the `ProvisioningTimeZone` parameter, you must also adjust the time zone on the prime client to match (as was described in Step 9 on page 49).

f Edit the `RunName` parameter if needed. (Each provisioning run requires a distinct run name.)

4 To start the provisioning process, from the command line execute:

```
java -jar tools/VMmark3Service.jar -c VMmark3.properties
```

NOTE The provisioning process initiates creation of the DS3DB0 virtual machine, which can take 12 hours or more while data is loaded into the DS3DB0 database. This procedure is not considered part of the provisioning process.

5 Though creation of the DS3DB0 virtual machine can take many hours, the provisioning process will complete much more quickly than that. Once the provisioning process has completed, update the prime client's `hosts` file, as follows:

NOTE The provisioning service does not modify the prime client's hosts file. Once provisioning has completed, the output folder will contain a `hosts-stub.txt` file that can be imported into the prime client's hosts file as described below.

a Populate the prime client's `hosts` file. In a terminal window on the prime client, run the following command:

```
cat ~/VMmark3/provisioning-output/<YourProvisioningRunName>/hosts-stub.txt >> /etc/hosts
```

b Add entries for the systems under test to the prime client's `/etc/hosts` file.

c Test the passwordless SSH configuration by using SSH to manually log in at least once to each host (i.e. `ssh root@[ESXhost1,2...] hostname`).

6 Wait for the DS3DB0 creation process to complete; depending on your environment, this can take 12 hours or more. To determine if it's complete, follow these steps:

a In a terminal window on the prime client, run the following command:

```
ssh <DS3DB0-IP-address> 'tail -f /ds3/VMmark3-DS3installoutput.txt'
```

(where *`<DS3DB0-IP-address>`* is the IP address of the DS3DB0 VM, which can be found in the `hosts-stub.txt` file referenced above).

b Watch for the following text:

```
VMmark3 DS3DB Setup Completed : Rebooting
```

7 *Optionally*, you can save storage space and decrease tile creation time by removing the third virtual disk from the tile 0 DS3DB virtual machine (DS3DB0). This can't be done until after tile 0 is provisioned and the DS3DB0 creation process is complete, but if you choose to remove it you should do so before provisioning additional tiles.

NOTE The storage requirements listed in "System Under Test Storage Requirements" on page 34 assume that this step has been performed.

Follow these steps to remove the tile 0 DS3DB third disk:

a From a terminal window in the DS3DB virtual machine, run the following commands:

```
mv /ds3/* /root/ds3/
umount /ds3
```

b Edit the `/etc/fstab` file, commenting out the `LABEL=DS3` line.

c From a terminal window in the DS3DB virtual machine, run the following command:

```
mv /root/ds3/* /ds3/
```

d Power off the DS3DB virtual machine.

e Using the vSphere Client, delete the third virtual disk from the DS3DB virtual machine.

Update One or More VMmark Workload Virtual Machines

If desired, you can at this time make certain changes to one or more workload virtual machines' virtual hardware version, VMware Tools version, or virtual hardware, as described below.

NOTE If any of the changes in this section are made, those changes must be made to all workload VMs of the same type, across all tiles.

Update Virtual Hardware Version

If desired, you can update the virtual hardware version for one or more workload virtual machines to the latest version supported in your environment.

Update VMmark Tools Version

If desired, you can update VMmark Tools in one or more workload virtual machines to the latest version supported in your environment.

Make One-for-One Replacements of Virtual Hardware

If desired, you can replace virtual hardware storage or network devices in one or more workload virtual machines with devices of the same type on a one-for-one basis. For example:

- One virtual disk can be replaced with one virtual persistent memory (PMem) device.
- One vNIC can be replaced with one vNIC of another type.

If any such replacements are made, you can also install in the guest operating system any supporting software, such as drivers or configuration tools, required to support those devices.

Creating Additional VMmark Tiles (Tiles 1 through n)

If you wish to run more than a single VMmark tile (that is, more than just tile 0), follow these steps to create additional VMmark tiles (that is, tiles 1 through n):

1 Use SSH to connect to the prime client, or use VNC or the vSphere Client console and open a terminal window using the graphical user interface.
(The default login is **test**, but click **Not listed?** and login as user: **`root`**, password: **`vmmark`**.)

2 CD to `~/VMmark3`.

3 Modify the following parameters in the `VMmark3.properties` file according to your environment and needs:

a `ProvisioningSource = tile0`

b `ProvisioningNumTiles =` *`n`*
(where *`n`* is the number of tiles you would like to end up with).

c Edit the `RunName` parameter if needed. (Each provisioning run requires a distinct run name.)

4 Power on tile 0. From a terminal window on the prime client, run:

```
cd ~/VMmark3
java -jar tools/VMmark3Service.jar -c VMmark3.properties -m tilePower -tiles 1
```

(this powers on one tile, that is, tile 0).

NOTE If the provisioning service in the next step finds tile 0 powered off, the service will power it on as part of this process, but it's faster to do it manually, as described in this step.

5 Start the provisioning process by executing the following from the command line:

```
java -jar tools/VMmark3Service.jar -c VMmark3.properties
```

6 Once the provisioning process has completed, update the prime client's hosts file.

NOTE Just as when provisioning tile 0, the provisioning service does not modify the prime client's hosts file. Once provisioning has completed, the output folder will contain a hosts-stub.txt file that can be imported into the prime client's hosts file as described below.

In a terminal window on the prime client, run the following command:

```
cat ~/VMmark3/provisioning-output/<YourProvisioningRunName>/hosts-stub.txt >> /etc/hosts
```

7 Edit the /etc/hosts file:

- If this is the first time you're provisioning additional tiles, you don't need to edit the /etc/hosts file.
- If you've previously provisioned additional tiles, you should open /etc/hosts in a text editor and remove any duplicate IP addresses and host names.
- If you've previously provisioned additional tiles and the value of ProvisioningIPstaticStart has changed, you should open /etc/hosts in a text editor and remove any inaccurate *or* duplicate IP addresses and host names.

Using the UnattendedProvisioning.pl Script

Instead of running the provisioning script manually, as described above, you can use a script to automate provisioning of all tiles. The UnattendedProvisioning.pl script first creates tile 0 then, when the DS3DB0 creation is complete, it creates the remaining tiles.

NOTE When using the UnattendedProvisioning.pl script, there is no option to remove the third virtual disk from the tile 0 DS3DB virtual machine (DS3DB0), as described in Step 7 on page 52. Not removing this virtual disk potentially increases tile creation time and increases storage space usage beyond the values provided in "System Under Test Storage Requirements" on page 34.

When using the script you can reduce storage space usage by removing the third virtual disk from the tiles created once the process has completed.

To use this script, perform the following steps:

1 Edit the VMmark3.properties file, as described in Step 1 through Step 3 in "Creating the First VMmark Tile (Tile 0)" on page 51, customized for your test environment and desired source for tile 0.

2 Create a new properties file, as described in Step 1 through Step 3 in "Creating Additional VMmark Tiles (Tiles 1 through n)" on page 53, but *with a different name*, customized for your test environment and the number of tiles you wish to provision.

NOTE We recommend that this new properties file start out as a copy of the tile 0 VMmark3.properties file, and that it be named VMmark3-*N*tile.properties (where *N* is the value of the ProvisioningNumTiles parameter in the file).

3 Run the ~/VMmark3/tools/VMmark3-UnattendedProvisioning.pl script with the above two files as arguments.

For example, to create three tiles, you'd issue the following command from the ~/VMmark3/tools directory:

```
perl VMmark3-UnattendedProvisioning.pl VMmark3.properties VMmark3-3tile.properties
```

Provisioning Notes

This section contains additional guidance about using the provisioning feature of VMmark 3.x.

Provisioning Additional Tiles

- To provision additional tiles, simply issue a new provisioning service request with a higher number of tiles. For example, if a user had built three tiles (tiles 0, 1, and 2) and wanted to increase to six tiles (adding tiles 3, 4, and 5), a new request would be made with `ProvisioningNumTiles = 6`. (And with a new value for the `RunName` parameter.)

  NOTE For the provisioning service to recognize the already-provisioned tiles (tiles 0, 1, and 2 in the above example), those tiles must be powered on and be reachable over the network.

  Any VMs that are found but not powered on will be powered on by the provisioning service, but leaving this for the provisioning service to do will delay completion of the task.

Recreating All or Part of a Tile

Recreating *All* of a Tile

To recreate an *entire* tile:

1 Remove (or rename) all the VMs in the tile to be recreated.

2 Make sure all the other VMs in your VMmark environment are powered on.

3 Issue a new provisioning request that includes the tile number to be recreated, specifying the provisioning source you want to use:
For tile 0, the provisioning source would be the template.
For tiles 1 through n, the provisioning source would be tile 0.
(This works because, when provisioning, the provisioning service will skip any complete tiles that it can see; that is, tiles that are powered on and reachable over the network.)

Recreating *Part* of a Tile

To recreate only *some* of the VMs in a tile:

1 Delete the VMs you want to recreate.

2 Make sure all the other VMs in your VMmark environment are powered on.

3 Modify the `ProvisioningDatastores` parameter (in the `VMmark3.properties` file) so that only the `VMtype:DatastoreName` pair (or pairs) you wish to recreate are listed.
(By removing the `VMtype:DatastoreName` pair (or pairs) for the VMs you don't want to recreate, you prevent the provisioning service from seeing those VMs and issuing an error message.)

4 Issue a new provisioning request that includes the tile number (or numbers) in which you wish to recreate VMs.

Disk Types for Provisioned VMs

- The disk type (that is, Thin, Thick Lazy, Thick Eager) of the *primary* disk for a provisioned VM will be equivalent to the selection used for the template.
- The disk type (that is, Thin, Thick Lazy, Thick Eager) of the *secondary* disk for a provisioned VM is controlled by the value of the `ProvisioningSecondaryDiskType` parameter in the `VMmark3.properties` file.

The VMmark3.properties File

Most options for a VMmark 3.x run are controlled by the contents of the `VMmark3.properties` file. The sections below show the sections that require editing for a first provisioning of VMmark, the section that requires editing for a first run of VMmark, and finally the entire `VMmark3.properties` file.

NOTE VM names used in the `VMmark3.properties` file should be alphanumeric only, should start with the pre-defined VM server names listed in the file, and should end in a tile number without padding (that is, with no hyphens, added zeros, or anything else besides the tile number). Only VMs with this name format are allowed for provisioning and usage.

VMmark3.properties File Edits Required for First VMmark Provisioning

The lines in the `VMmark3.properties` file that require editing for a first provisioning of VMmark, highlighted in magenta in "Entire VMmark3.properties File" on page 57, are:

```
PrimeClient =
vCServer =
vCServerUsername =
vCServerPassword =
vCServerDatacenter =
vCServerVMmark3Cluster =
vCServerClientCluster =
ProvisioningSource =
ProvisioningDatastores =
```

NOTE This is a listing of the lines that require editing in *all* VMmark deployments. There might be other lines that need to be changed for your specific environment.

VMmark3.properties File Edits Required for First VMmark Run

The lines in the `VMmark3.properties` file that require editing for a first, single-tile, performance-only run of VMmark, highlighted in green in "Entire VMmark3.properties File" on page 57, are:

```
SVmotion/SVMotionLUNs =
XVmotion/XVMotionLUNs =
Deploy/DeployLUNs =
Deploy/Templates =
Deploy/DeployVMinfo =
EmailList =
EmailSMTP =
EmailFrom =
EmailPort =
```

NOTE Some of the edits required for a first provisioning of VMmark are also required for a first VMmark run. This section assumes you've already made all the edits shown in "VMmark3.properties File Edits Required for First VMmark Provisioning" on page 56, and thus doesn't highlight them again.

NOTE This is a listing of the lines that require editing in *all* VMmark deployments. There might be other lines that need to be changed for your specific environment.

NOTE Depending on the number of simultaneous infrastructure operations your environment will need (as calculated in Chapter 3, "Calculate the Number of Simultaneous Infrastructure Operations," on page 45), some of these lines might each need more than one entry, separated by commas.

Entire VMmark3.properties File

A copy of the entire `VMmark3.properties` file is included here for reference. As described above, the lines that require editing for a first provisioning of VMmark are highlighted in magenta; the additional lines that require editing for a first run of VMmark are highlighted in green. Depending on your environment, additional settings might also need to be edited.

```
//                Example VMmark(r) Properties File
//  Copyright 2006-2019 VMware, Inc. All rights reserved.
//
//   This file is part of the VMmark Harness.
//
//    The VMmark Harness is free software; you can redistribute it and/or modify
//    it under the terms of the GNU General Public License, Version 2.0,
//    as published by the Free Software Foundation.
//
//    The VMmark Harness is distributed in the hope that it will be useful,
//    but WITHOUT ANY WARRANTY; without even the implied warranty of
//    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
//    GNU General Public License for more details.
//
//    You should have received a copy of the GNU General Public License
//    along with the VMmark Harness; if not, write to the Free Software
//    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
//
               ---------------------------------------------------------------------------------
               -------------------------------------------------
// VMmark3.1 : 12/17/2018
// **** Read Me First ****
// This file is sensitive to case, spacing, and line endings.  Follow the example formats.
//   Use VM names that are alphanumeric only, starting with alpha, ending in a tile number.
//   Only VMs with the below name formatting are allowed for provisioning and usage.
//   Comma separated lists (CSV) should not end with a comma.  Correct format is: Ex
               item1,item2,item3
//
               ---------------------------------------------------------------------------------
               -------------------------------------------------
// General Settings #general
//
// Boolean EmailNotifications : Default true : Controls if service will attempt to utilize email
               notifications
// EmailNotifications = true
// CSV String EmailList : Default empty : Ex = user1@vmware.com,user2@vmware.com
EmailList =
// String EmailSMTP : Default empty : Ex = email.vmware.com
EmailSMTP =
// String EmailFrom : Default empty : Ex = user1@vmware.com
EmailFrom =
// String EmailPort : Default empty : Ex = 25
EmailPort =
// String PrimeClient : Default empty : Ex: 192.168.1.1 : Used to create PrimeClient entries in
               the hosts files on the workload VMs.
PrimeClient =
// String RMQserver : Default localhost : Should typically be localhost
// RMQserver = localhost
//
               ---------------------------------------------------------------------------------
               -------------------------------------------------
// vSphere Settings #vsphere
// This section details the connection settings for the vCServer under test.
//
// String vCServer : Default empty : Ex: 10.115.155.239
vCServer =
//  String vCServerUsername : Default empty : Ex vCServerUsername = administrator@vsphere.local
vCServerUsername =
//  String vCServerPassword : Default empty : Ex vCServerPassword = vmware
vCServerPassword =
// String vCServerDatacenter : Default empty : Ex vCServerDatacenter = Datacenter
```

```
vCServerDatacenter =
// String vCServerVMmark3Cluster : Default empty : Ex vCServerVMmark3Cluster = Cluster1
vCServerVMmark3Cluster =
// String vCServerClientCluster : Default empty : Ex vCServerClientCluster = ClientCluster1 :
               This is the cluster of hosts running your
//    Client VMs and is where provisioning will place new client VMs.
vCServerClientCluster =
//
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// VMmark Run Configuration #run
// This section specifies many of the required settings for running the VMmark3 benchmark.
//   Note: Tile 0 placeholders have been added to this section to assist users.
// int Tiles : Default empty : Number of Tiles to Run
Tiles = 1
// int RunTimeSeconds : Default 10800 : This is the run time in seconds.  For compliant runs this
               must be 10800
// RunTimeSeconds = 10800
// Clients: These are the VMs that generate the external load.  They should be provisioned into a
               separate client cluster.
//   CSV String Clients : Default empty : Ex Client0,Client1,Client2
Clients = Client0
// Servers: These are the VMs that load is generated upon.  They make up the individual
               components of a tile.  All are required for compliance.
//   CSV String Workload/Servers : Default empty : Ex Standby/Servers =
               Standby0,Standby1,Standby2
// Note: the script 'VMmark3/tools/VMmark3-PrintWorkloadConfigurationStub.pl' automates the
               generation of these values for multiple tiles.
//   see the VMmark user's guide for additional information
Standby/Servers = Standby0
AuctionWebA/Servers = AuctionWebA0
AuctionWebB/Servers = AuctionWebB0
AuctionAppA/Servers = AuctionAppA0
AuctionAppB/Servers = AuctionAppB0
AuctionDB/Servers = AuctionDB0
AuctionMSQ/Servers = AuctionMSQ0
AuctionLB/Servers = AuctionLB0
AuctionNoSQL/Servers = AuctionNoSQL0
DS3WebA/Servers = DS3WebA0
DS3WebB/Servers = DS3WebB0
DS3WebC/Servers = DS3WebC0
DS3DB/Servers = DS3DB0
ElasticLB/Servers = ElasticLB0
ElasticWebA/Servers = ElasticWebA0
ElasticWebB/Servers = ElasticWebB0
ElasticAppA/Servers = ElasticAppA0
ElasticAppB/Servers = ElasticAppB0
ElasticDB/Servers = ElasticDB0
//
// Infrastructure Ops Settings
// This section must be filled out prior to beginning benchmark runs.
// String SVmotion/SVMotionLUNs : Specifies which LUNs to use for Storage VMotions.  Must be
               shared storage. : Default empty
SVmotion/SVMotionLUNs =
// String XVmotion/XVMotionLUNs : Specifies which LUNs to use for XVMotions.  Must be shared
               storage. : Default empty
XVmotion/XVMotionLUNs =
// String Deploy/DeployLUNs : Specifies which LUNs to use for Deploys.  Must be shared storage. :
               Default empty
Deploy/DeployLUNs =
// The deploy workload is a type of provisioning and thus leverages the provisioning settings
               specified below.
//   Validate that they are up to date for your environment prior to execution.  The following
               settings are necessary.
// String Deploy/Templates : Specifies the template to deploy from.  Should be the current
               VMmark3 template; otherwise
//   resulting runs will be non-compliant. Default empty : Ex vmmark3-template-103016
// Note for multiple simultaneous infrastructure operations, multiple templates are required : Ex
               vmmark3-template-103016-1,vmmark3-template-103016-2
Deploy/Templates =
// CSV String Deploy/DeployVMinfo : Specifies the name and IP address of the deployed VMs. Ex
               DeployVM0:192.168.1.245,DeployVM1:192.168.1.246
// Note if set during provisioning, the user defined values will be added to the hosts-stub.txt
Deploy/DeployVMinfo =
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// Benchmark Workload Settings  #workload
// This section specifies which infrastructure and application workloads to be used.  Changing
               these parameters is non-compliant.
// CSV InfrastructureList : Default vmotion,svmotion,xvmotion,deploy
// InfrastructureList = vmotion,svmotion,xvmotion,deploy
// CSV WorkloadList : Default
               Standby,DS3WebA,DS3WebB,DS3WebC,DS3DB,AuctionLB,AuctionWebA,AuctionWebB,AuctionApp
               A,AuctionAppB,AuctionDB,AuctionMSQ,AuctionNoSQL,ElasticLB,ElasticWebA,ElasticWebB,
               ElasticAppA,ElasticAppB,ElasticDB,
// WorkloadList =
               Standby,DS3WebA,DS3WebB,DS3WebC,DS3DB,AuctionLB,AuctionWebA,AuctionWebB,AuctionApp
               A,AuctionAppB,AuctionDB,AuctionMSQ,AuctionNoSQL,ElasticLB,ElasticWebA,ElasticWebB,
               ElasticAppA,ElasticAppB,ElasticDB,
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// String ManualHosts : Default empty : If empty the harness will automatically use the names of
               hosts found in the cluster under test for ESXTOP and Reporter Routines.
//   Otherwise a user can specify name or IP address.  Ex:  ManualHosts = host1,host2
// ManualHosts = host1,host2
// String ManualClientHosts : Default empty : If empty the harness will automatically use the
               names of hosts found in the client cluster specified for Reporter Routines.
//   Otherwise a user can specify name or IP address.  Ex:  ManualClientHosts =
               clienthost1,clienthost2
// ManualClientHosts = clienthost1,clienthost2
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// ESXTOP Configuration
// Boolean EsxtopCollection : Default false : Enables ESXTOP collection on the hosts under test.
               If enabled the ESXtopLUN variable must also be populated.
// EsxtopCollection = false
// String EsxtopLUN : Default empty : Absolute path to LUN where esxtop should be collected.
               Must be shared storage for the systems under test.
//   Note: Space is a consideration as ESXTOP files can grow large.  Ex:
               /vmfs/volumes/MY-LUN-NAME
// EsxtopLUN =
//
// - Optional Settings - Defaults shown
// - Warning: Some of these settings must be left at their default values for a run to be
               compliant.
//
// Workload Delays
// This section allows a user to specify the delay (in seconds) for specific workloads.  Changing
               these parameters is compliant.
// DS3WebA/DELAYTIME = 8
// DS3WebB/DELAYTIME = 11
// DS3WebC/DELAYTIME = 15
// AuctionLB/DELAYTIME = 8
// Standby/DELAYTIME = 2
// ElasticLB/DELAYTIME = 8
//
// Workload Skip Restores
// This section allows a user to skip the restore process for specific workloads.  Changing these
               parameters is non-compliant.
// DS3WebA/SKIPRESTORE = false
// DS3WebB/SKIPRESTORE = false
// DS3WebC/SKIPRESTORE = false
// DS3DB/SKIPRESTORE = false
// AuctionLB/SKIPRESTORE = false : This value cannot be changed due to application design
// AuctionAppA/SKIPRESTORE = false
// AuctionAppB/SKIPRESTORE = false
// AuctionDB/SKIPRESTORE = false
// AuctionMSQ/SKIPRESTORE = false
// AuctionNoSQL/SKIPRESTORE = false
// AuctionWebA/SKIPRESTORE = false
// AuctionWebB/SKIPRESTORE = false
// ElasticLB/SKIPRESTORE = false
// ElasticAppA/SKIPRESTORE = false
// ElasticAppB/SKIPRESTORE = false
// ElasticWebA/SKIPRESTORE = false
// ElasticWebB/SKIPRESTORE = false
// ElasticDB/SKIPRESTORE = false
//
// Misc Workload Settings
// Changing these parameters is non-compliant
// int Standby/INTERVAL : Specifies how long to wait between STAF pings. : Default 60
// Standby/INTERVAL = 60
// int Weathervane/AuctionUsers : Specifies the number of Weathervane users : Minimum Weathervane
               Auction users = 60 : Default = 9500
// Weathervane/AuctionUsers = 9500
// int Weathervane/ElasticUsers : Specifies the number of Weathervane Elastic users : Minimum
               Weathervane Elastic users = 60 : Default = 1500
// Weathervane/ElasticUsers = 1500
// int DS3DB/DS3DBsize : Specifies the DS3DBsize in GB : Default = 100
// DS3DB/DS3DBsize = 100
// int DS3Web/DS3thinktime : Specifies the DS3Web think time for operations: Default = 1
// DS3Web/DS3thinktime = 1
//
// Workload Delays
// This section allows a user to specify the delay for specific workloads.  Changing these
               parameters is compliant.
// SVmotion/DELAYTIME = 16
// Vmotion/DELAYTIME = 10
// XVmotion/DELAYTIME = 16
// Deploy/DELAYTIME = 17
//
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// Advanced Infrastructure Operations Settings.  Changing these parameters is non-compliant.
//
// This section allows the user to specify the amount of time between infrastructure operations.
// SVmotion/SleepBetween = 120
// Vmotion/VMotionDelayBetween = 120
// XVmotion/SleepBetween = 150
// Deploy/DeployDelayBetween = 40
//
// This section allows the user to specify the workload used for various infrastructure
               operations.
// SVmotion/Workloads = Standby
// Vmotion/Workloads = AuctionMSQ
// XVmotion/Workloads = DS3WebA
//
// This section allows the user to specify the burst queue size (minimum 1) used for various
               infrastructure operations.
// SVmotion/BurstQueueSize = auto
// Vmotion/BurstQueueSize = auto
// XVmotion/BurstQueueSize = auto
// Deploy/BurstQueueSize = auto
//
// This section allows the user to specify the amount of free space verified for various
               infrastructure operations.
// SVmotion/FreeSpaceFactor = 2
// XVmotion/FreeSpaceFactor = 2
//
```

```
// This sections allows the user to specify the size (in GB) of the Deploy VM's secondary disk.
// Deploy/DeployDBsize = 20
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// Miscellaneous Settings and Options.  Changing these parameters is compliant.
// Boolean ScrubConfigFile : Default false : Triggers the removal of passwords from the VMmark3
               properties file.
// ScrubConfigFile = false
// int TileDelay : Default 60 : Delays start of Tile N by TileDelay * N seconds
// TileDelay = 60
// int DebugLevel : Default 0 : Value that controls the display/logging of info/debug messages.
               Valid options are 0-3
// DebugLevel = 0
// Additional Misc Settings.  Changing these values is non-compliant
// Boolean TurboMode : Enables a quick VMmark3 run of 30 minutes, with automatic score
               generation.  Useful for preliminary testing, not compliant for submissions :
               Default false
// TurboMode = false
// Boolean Reporter : Enabled the collection of necessary system information.  Required for
               publications. : Default true
// Reporter = true
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// Power (PTD) Settings #power #ptd
// This section should be configured with the use of the VMmark3 Benchmarking Guide.  Notes and
               parameters definitions are provided there.
// PTD=True/False: Valid values [True/False] Enable/Disable
// PTD_Clients: STAF-enabled systems connected to power meters
//   Note: The number of PTD_Clients determines the total number of PTD
//   processes activated. 2 clients would require 2 ports, etc.
// PTD_Target: Classification of metered items (SERVER or EXT_STOR [storage])
// PTD_Ports: Port used for each power meter.
// PTD_MeterTypes: Meter Type for each power meter.
// PTD_COMs: COM port for each power meter on its PTD client.
// PTD_Amp_Ranges: Allows defining of different Ampere ranges per meter.
// PTD_Volt_Ranges: Allows defining of different Volt ranges per meter.
// PTD_Enable_GPIB: Allows the use of GPIB devices, set to 1 to enable
// Ex: PTD_Client:PTDclient1 -> Target:SERVER -> Port:8888 -> MeterType:8 (Yokogawa) -> COM:COM1
               -> Range:4.0
// Ex: PTD_Client:PTDclient2 -> Target:SERVER -> Port:8890 -> MeterType:8 (Yokogawa) -> COM:COM1
               -> Range:4.0
//
// PTD = false
// PTD_Clients = ptdclient61,ptdclient43,ptdclient44,ptdclient45
// PTD_Target = SERVER,SERVER,EXT_STOR,EXT_STOR
// PTD_Ports = 1016,1017,1018,1019
// PTD_MeterTypes = 8,8,8,8
// PTD_COMs = /dev/ttyS0,/dev/ttyS1,/dev/ttyS2,/dev/ttyS3
// PTD_Amp_Ranges = 5.0,5.0,20.0,20.0
// PTD_Volt_Ranges = 120.0,120.0,120.0,120.0
// PTD_Enable_GPIB = 0,0,0,0
// - Optional Settings - Defaults shown
// - Warning: Changing values may make resulting runs non-compliant or cause issues.
// PTD_RampTime=1800
// PTD_Local_Hostnames="testbed,testbed,testbed,testbed"
// PTD_Local_Ports="0,0,0,0"
//
               ==================================================================================
               =======================================================
//
// Automated Build Out Provisioning #provisioning
//
// ProvisioningDatastores is used to specify which VMs (workloads) to work with.  Because each
               Workload requires a datastore
//   they're bundled together so users don't have to repeat the same information in multiple
               places.
```

```
// -- Provisioning uses some of the settings from above.  Validate that those settings are up to
               date. --
//
// Boolean AllowRunOverwrite : Default false : Controls if service will halt if output folder is
               found
AllowRunOverwrite = false
// String runName : Name of provisioning output, stored in ~/VMmark3/provisioning-output/<name> :
               Default vmmark3-output : Ex VMmark3-BuildOut1
RunName = vmmark3-output
// String ProvisioningSource : default empty: This is the mechanism for users to specify which
               tile is being created.
//   For provisioning Tile 0 this should be a valid VMmark3 template : Ex: 'ProvisioningSource =
               vmmark3-template-103016'
//   For provisioning Tile 1 - N, Tile0 is used as the source and this should be the keyword
               'tile0' : Ex 'ProvisioningSource = tile0'
//   Default empty
ProvisioningSource =
// int ProvisioningNumTiles : 1 or more tiles : Default 1 : If ProvisioningSource is a template
               this is automatically set to 1 tile.
// ProvisioningNumTiles = 1
// String ProvisioningDatastores : Default empty : This variable is used to decode the datastore
               placement of the provisioned VMs.
//   Format is "VMtype1:DatastoreName,VMtype2:DatastoreName,VMtype3:DatastoreName"
//   Accepted VMTypes
               [client,DS3WebA,DS3WebB,DS3WebC,DS3DB,AuctionLB,AuctionMSQ,AuctionWebA,AuctionWebB
               ,AuctionAppA,AuctionAppB,AuctionNoSQL,
//
               AuctionDB,ElasticLB,ElasticWebA,ElasticWebB,ElasticAppA,ElasticAppB,ElasticDB,Stan
               dby]
//   Note: For ease of readability, line wrapping is allowed for ProvisioningDatastores.  Below
               are examples of both formats allowed.
//   Traditional Format Example:
//     ProvisioningDatastores =
               Client:HDD-Lun1,DS3WebA:HDD-Lun2,DS3WebB:HDD-Lun2,DS3WebC:HDD-Lun2,DS3DB:SSD-Lun6,
               AuctionLB:SSD-Lun5,AuctionMSQ:SSD-Lun5,AuctionWebA:HDD-Lun2,AuctionWebB:HDD-Lun3,A
               uctionAppA:HDD-Lun1,AuctionAppB:HDD-Lun2,AuctionNoSQL:SSD-Lun6,AuctionDB:SSD-Lun6,
               ElasticLB:iSCSI-Lun0,ElasticWebA:iSCSI-Lun1,ElasticWebB:iSCSI-Lun1,ElasticAppA:iSC
               SI--Lun0,ElasticAppB:iSCSI-Lun0,ElasticDB:iSCSI-Lun0,Standby:HDD-Lun1
//   Wrapped Format Example:  (Note an example is provided below, users should replace the
               example LUN values prior to provisioning)
ProvisioningDatastores = \
Client:HDD-Lun1,\
DS3WebA:HDD-Lun2,\
DS3WebB:HDD-Lun2,\
DS3WebC:HDD-Lun2,\
DS3DB:SSD-Lun6,\
AuctionLB:SSD-Lun5,\
AuctionMSQ:SSD-Lun5,\
AuctionWebA:HDD-Lun2,\
AuctionWebB:HDD-Lun3,\
AuctionAppA:HDD-Lun1,\
AuctionAppB:HDD-Lun2,\
AuctionNoSQL:SSD-Lun6,\
AuctionDB:SSD-Lun6,\
ElasticLB:iSCSI-Lun0,\
ElasticWebA:iSCSI-Lun1,\
ElasticWebB:iSCSI-Lun1,\
ElasticAppA:iSCSI--Lun0,\
ElasticAppB:iSCSI-Lun0,\
ElasticDB:iSCSI-Lun0,\
Standby:HDD-Lun1
// String ProvisioningSecondaryDiskType : Default ThickEager : While the disk type (Thin,
               ThickLazy, ThickEager) of provisioned VMs' primary disks is equivalent to the
               selection used for the template,
//    the disk type of provisioned VMs' secondary disks can be set using the
               ProvisioningSecondaryDiskType configuration variable.
//    Ex: ProvisioningSecondaryDiskType = ThickLazy
// ProvisioningSecondaryDiskType = ThickEager
// int ProvisioningTimeoutMinutes : default 60 : Maximum time (in minutes) that provisioning will
               wait on a task to complete (minutes) Minimum = 1:  Ex ProvisioningTimeoutMinutes =
               40
// ProvisioningTimeoutMinutes = 60
// String ProvisioningDomain : Domain used for network settings : Default vmmark3.eng.vmware.com
// ProvisioningDomain = vmmark3.eng.vmware.com
// String ProvisioningTimeZone : Timezone used for provisioned VMs : Default US/Pacific : Per the
               spec this is case sensitive
//   The VMmark3 service does not attempt to validate this value
//    For examples see:
               https://www.vmware.com/support/developer/vc-sdk/visdk41pubs/ApiReference/timezone.
               html
// ProvisioningTimeZone = US/Pacific
// String ProvisioningIPmode : network mode of provisioning : Default static, DHCP is optional
               but non-compliant
// ProvisioningIPmode = static
// String ProvisioningNetworkLabel : Network label to use for provisioned VMs : Default VM
               Network : Note that vSphere will allow you to set this to
//   a nonexisting network label.
// ProvisioningNetworkLabel = VM Network
// String ProvisioningNetworkType : standard or dvswitch modes of provisioning : Default standard
//   Note:  dvswitch mode requires the use of Ephemeral distributed port groups.
// ProvisioningNetworkType = standard
//
// Additional Network Settings
// String ProvisioningIPstaticStart : First IP address to use. Valid ranges are 1-245.  245 and
               above are reserved for gateways and deploy operations VMs.  For reserving more
               simply start
//   the range later (i.e., 192.168.1.3 will always reserve the first two IP addresses;
               Ex:[192.168.1.1, 192.168.1.2, 192.168.2.1, 192.168.2.2] won't be assigned).
//   Default : 192.168.1.1
// ProvisioningIPstaticStart = 192.168.1.1
// String ProvisioningGateway : Gateway to use for static VM provisioning : Default 192.168.1.253
// ProvisioningGateway = 192.168.1.253
// String ProvisioningSubnetMask : Subnet mask to use for static VM provisioning : Default
               255.255.252.0
// ProvisioningSubnetMask = 255.255.252.0
// int ProvisionPingMaxTries : Default 24 : Specifies the number of ping attempts before failing.
               Each ping attempt waits 30 seconds.
// ProvisionPingMaxTries = 24
// int ProvisionPingIpSeconds : Default 600 : Specified the number of seconds to wait for a VM to
               get an IP address.
// ProvisionPingIpSeconds = 600
// int ProvisioningMaxSimultaneousOperations : Default is the number of workload VMs being
               operated on.  This can be used to slow down the deployment requests.
// ProvisioningMaxSimultaneousOperations =
// int RebootTimeSeconds : Default 90 : Time to allow VMs to reboot before continuing the
               provisioning process.
// RebootTimeSeconds = 90
// Each of the following settings requires two, and only two, entries
// String ProvisioningDnsServerList : DNS severs to use for VM provisioning : Default
               10.20.20.1,10.20.20.2
// ProvisioningDnsServerList = 10.20.20.1,10.20.20.2
// String ProvisioningDnsSuffixList : DNS suffix to use for VM provisioning : Default
               eng.vmware.com,eng.vmware.com
// ProvisioningDnsSuffixList = eng.vmware.com,eng.vmware.com
//
//
               ---------------------------------------------------------------------------------
               --------------------------------------------------
// Extra Config Options #extra
// Changing the following parameters is compliant but it may make resulting runs unstable if done
               incorrectly.  Use with care.
//
// This section specifies the TimeOuts and Delays for various operations: Note all values are
               specified in seconds
// int VMmark3ServiceInitializationSeconds : Default 10 : Time to wait for VMmark3 Service to
               start
```

```
// VMmark3ServiceInitializationSeconds = 10
// int ClusterMappingTimeOut : Default 3600 : Time to wait for cluster mapping to complete
// ClusterMappingTimeOut = 3600
// int PostProcessTimeOut : Default 300 : Time to wait for Tilescore.pl to run
// PostProcessTimeOut = 300
// int CancelTestTimeOut : Default 1200 : Time to wait for cancel routines to finish during an
               error
// CancelTestTimeOut = 1200
// int CancelTestBlockSize : Default 10 : The number of tiles which should be processed
               concurrently
// CancelTestTimeOut = 1200
// int CheckTimeTimeOut : Default 180 : Time to check the time of clients
// CheckTimeTimeOut = 180
// int vCServerTimeTimeOut : Default 180 : Time to check the time of vCServer
// vCServerTimeTimeOut = 180
// int ReporterTimeOut : Default 5400 : Time to run reporter routines
// ReporterTimeOut = 5400
// int ReporterCleanupTimeOut : Default 180 : Time to run reporter cleanup routines
// ReporterCleanupTimeOut = 180
// int HostSshTimeOut : Default 180 : Time to create host ssh connections
// HostSshTimeOut = 180
// int VCsupportTimeOut : Default 2400 : Time to collect VC support data
// VCsupportTimeOut = 2400
// int CollectEventsTimeOut : Default 900 : Time to collect VC events
// CollectEventsTimeOut = 900
// int EsxtopSetupTimeOut : Default 180 : Time to set up esxtop on cluster hosts under test
// EsxtopSetupTimeOut = 180
// int EsxtopRuntimeTimeOut : Default 600 : Additional Time to allow esxtop on cluster hosts
               under test to complete
// EsxtopRuntimeTimeOut = 600
// int ClusterReportTimeOut : Default 900 : Time to create cluster report
// ClusterReportTimeOut = 900
// int EnvInfoTimeOut : Default 600 : Time to collect EnvInfo
// EnvInfoTimeOut = 600
// int ListHostsTimeOut : Default 300 : Time to run the list hosts routine
// ListHostsTimeOut = 300
// int ListVMsTimeOut : Default 300 : Time to run the list VMs routine
// ListVMsTimeOut = 300
// int ListLUNsTimeOut : Default 300 : Time to run the list LUNs routine
// ListLUNsTimeOut = 300
// int CheckDrsTimeOut : Default 300 : Time to run the CheckDrs routines
// CheckDrsTimeOut = 300
// int MiscInfoTimeOut : Default 300 : Time to allow for guest info, Config file, and Misc Info
               collections
// MiscInfoTimeOut = 300
// int PtdMiscTimeOut : Default 300 : Time to allow for various Ptd Setup Routines to complete
// PtdMiscTimeOut = 300
// int PtdWorkloadMinTimeOut : Default 120 : Minimum Time to allow for Ptd workload to complete
               in.
// PtdWorkloadMinTimeOut = 300
// int InfrastructureWorkloadMinTimeOut : Default 1800 : Minimum Time to allow for Infrastructure
               workloads to complete in.
// InfrastructureWorkloadMinTimeOut = 1800
// int StandyWorkloadMinTimeOut : Default 300 : Minimum Time to allow for Standby workloads to
               complete in.
// StandyWorkloadMinTimeOut = 300
// int DS3WorkloadMinTimeOut : Default 300 : Minimum Time to allow for DS3WebA DS3WebB DS3WebC
               workloads to complete in.
// DS3WorkloadMinTimeOut = 300
// int AuctionWorkloadMinTimeOut : Default 3600 : Minimum Time to allow for Auction workloads to
               complete in.
// AuctionWorkloadMinTimeOut = 3600
// int SVmotion/SVmotionSetupTimeOut : Default 900 : Time to check the configuration of SVmotion
// SVmotion/SVmotionSetupTimeOut = 900
// int XVmotion/XVmotionSetupTimeOut : Default 900 : Time to check the configuration of XVmotion
// XVmotion/XVmotionSetupTimeOut = 900
// int Deploy/DeploySetupTimeOut : Default 900 : Time to check the configuration of Deploy
// Deploy/DeploySetupTimeOut = 900
// int EsxtopFileWriteDelay : Default 5 : Time to wait for compressing/writing of esxtop output
               before collecting
// EsxtopFileWriteDelay = 5
// int FileWriteDelay : Default 5 : Time to wait for output files to be written before collecting
// FileWriteDelay = 5
// int WorkloadDelay : Default 5 : Time to wait for workloads to initialize before starting run
// WorkloadDelay = 5
// String VCscratchDir : Directory used for various temporary harness activities : Default
               /root/VMmark3/samples/
// VCscratchDir = /root/VMmark3/samples/
// Boolean ErrorImmediate : Default false : Causes harness to exit immediately if an error
               occurs.  Note that even when set to false, some errors will nevertheless result in
               immediate termination of the run.
// ErrorImmediate = false
// String TmpDirectory : Temporary director for some operations, can be changed to accommodate
               space issues : Default /root/VMmark3/tmp
// TmpDirectory = /root/VMmark3/tmp
// int StafMaxThreads : Maximum number of parallel staf threads to execute for operations. :
               Default the number of tiles or 100, whichever is greater
// StafMaxThreads = 100
// String HostResourcePool : Used for advanced provisioning where created VMs will be placed
               within an existing resource pool : Default is unused
// HostResourcePool =
// String HostVMfolder : Used for advanced provisioning where created VMs will be placed within
               an existing VM folder : Default is unused
// HostVMfolder =
// String ClientResourcePool : Used for advanced provisioning where created client VMs will be
               placed within an existing resource pool : Default is unused
// ClientResourcePool =
// String ClientVMfolder : Used for advanced provisioning where created client VMs will be placed
               within an existing VM folder : Default is unused
// ClientVMfolder =

//
// - Warning: Changing the following parameters will make resulting runs non-compliant or
               unstable.
// Boolean DoClusterMapping = Default true :  Flag that enables the cluster mapping routines
// DoClusterMapping = true
// String ScriptName : Name of Master Reporter script : Default VMmark-ReporterMaster.sh
// ScriptName = VMmark-ReporterMaster.sh
// String VCScriptName : Name of VC Reporter script : Default VMmark-Reporter-VC.sh
// VCScriptName = VMmark-Reporter-VC.sh
// String ReporterDirectory : String where Reporter scripts are located : Default
               /root/VMmark3/tools/
// ReporterDirectory = /root/VMmark3/tools/
// String VMmarkDirectory : Location of VMmark3 : Default /root/VMmark3/
// VMmarkDirectory = /root/VMmark3/
// String ClientRootDirectory : Location of VMmark3 Client directory : Default
               /root/VMmark3/client
// ClientRootDirectory = /root/VMmark3/client
// String ProvisioningSecondaryDisks : This setting controls the size of disks to be added to
               Tile0 VM provisioning operations.  Note that this must match
//   what is expected by the current make-(VM) scripts in the VM's ~/VMmark3/tools/ directory.
               Any modification will cause subsequent runs to be non-compliant.
//   Lists the VMtype:SizeInGB of secondary disks
//   Default : DS3DB:250;100,AuctionNoSQL:100,ElasticLB:25,AuctionDB:20,Deploy:10
// ProvisioningSecondaryDisks =
               DS3DB:250;100,AuctionNoSQL:100,ElasticLB:25,AuctionDB:20,Deploy:10
// String ProvisioningDBsize  : This changes the size of DB DS3DB0 will be created with : Default
               100
// ProvisioningDBsize = 100
//
               ----------------------------------------------------------------------------------
               --------------------------------------------------
// Deprecated:
// POSTPROCESSFLAG = false
// RESCUE = false
// CLEANUPFLAG = false
// ProvisioningAutomation = /root/VMmark3/tools/AutoWorkloadConfig.pl
SSHname = root
SSHpassword = vmmark
SSHport = 22
//
```

Running the VMmark Benchmark 4

This chapter describes the process of running the VMmark benchmark. It consists of the following sections:

- "Starting a VMmark Run" on page 67
- "VMmark Results Files" on page 70

Preparing for a VMmark Run

To prepare for a VMmark run, open the `VMmark3.properties` file and edit the VMmark Run Configuration section to reflect your environment, following the directions included in that file.

VMmark includes a simple script to generate the workload configurations needed for the `VMmark3.properties` file for n tiles using the default naming. It simply prints the output to the screen and can be redirected or copied into your `VMmark3.properties` file.

For example, for two tiles:

```
perl tools/VMmark3-PrintWorkloadConfigurationStub.pl 2
```

This will output something like:

```
Clients = Clients0,Clients1
Standby/Servers = Standby0,Standby1
```

etc.

Starting a VMmark Run

Once the `VMmark3.properties` file has been appropriately edited, you can start a VMmark run using either the GUI or a command line, as described in the two sections below.

NOTE When performing a benchmarking run, it's best to have powered on only those VMs that are being used in the run. To power on a subset of the tiles in an environment, open a terminal window on the prime client and run:

```
cd ~/VMmark3
java -jar tools/VMmark3Service.jar -c VMmark3.properties -m tilePower -tiles n
```

(where *n* is the number of tiles you want to power on).

This will ensure that the power for only the VMs in the first *n* tiles (that is, tiles 0 through *n*-1) is on.

You can also use this command to power all VMs off by setting *n* to 0.

Starting a VMmark Run Through the GUI

To start a VMmark run through the GUI, follow these steps:

1 From the prime client's VNC or vSphere Client console, start STAX using one of these methods:

- To start STAX using the GUI, double click on the **VMmark3-StartSTAX** icon on the desktop.

- To start STAX using the command line, open a terminal window and enter the following:
  `sh /root/VMmark3/tools/VMmark3-STAX.sh`

2 From within the STAX window:

a Click **Save** to save the default properties.

b Click on **Submit New Job**.

c In the **Job Info** tab:
XML Job File > **Local Machine** > **Filename**
Browse to `/root/VMmark3/xml/vmmark3_main.xml`

d In the **STAX 3 Job Monitor** window, under the **Job Info** tab, under **Job Options**, enter a job name.

NOTE The STAX job name must not include spaces.

e Optionally, to use a properties file with a different filename than the default `VMmark3.properties`:

NOTE The default `VMmark3.properties` filename is used throughout this User's Guide and other VMmark documentation. If you chose to use an alternate properties filename, be sure to substitute the alternate name as required.

i Click **Job Wizard...**.

ii Under **Functions**, click **Main (default)**.

iii Under **Arguments for function Main**, click **Edit**.

iv Specify the full path to the alternate properties filename, and click **Save**.

v Click **Save** again and **Yes** to confirm.

f Click **Submit New Job** to start a VMmark run.

The **STAX Job Monitor** window opens and STAX starts the VMmark harness. The current status of the running workloads is shown in this window.

NOTE After the above information has been entered, you can use the **Resubmit Previous Job** button. The names entered are remembered across restarts of the Monitor. STAX will locate the latest `VMmark3.properties` file and the XML Harness code each time.

Starting a VMmark Run Without a GUI

To start a VMmark run without a GUI, follow these steps:

1 SSH to the prime client:
`ssh root@`*`primeclient`*
(replacing *`primeclient`* with the name of your prime client).

2 Change to the `/root/VMmark/tools/` directory:
`cd /root/VMmark3/tools/`

3 Optionally, to use a properties file with a different filename than the default VMmark3.properties:

NOTE The default `VMmark3.properties` filename is used throughout this User's Guide and other VMmark documentation. If you chose to use an alternate properties filename, be sure to substitute the alternate name as required.

a Open `VMmark3-STAX-console.sh` in an editor.

b Change the line that reads:
`configfile=/root/VMmark3/VMmark3.properties`
to the reflect the alternate properties filename.

c Save the file.

4 Issue the following command:

```
./VMmark3-STAX-console.sh jobname
```

(replacing *jobname* with a job name, which must not include spaces).

Run Notes

- To manually request that a `hosts-stub.txt` file be created in your current directory, issue the following command:

```
java -jar tools/VMmark3Service.jar -c VMmark3.properties -m hostsFile -tiles
<numtiles>
```

VMmark Results Files

After a VMmark benchmark test run has completed, the VMmark harness automatically collects the results data from the client system (or systems) and places those results in the results directory on the prime client, then processes the results to score the test.

The benchmark captures the 60-second throughput measurements for each workload and stores them in files with the `.wrf` (Workload Results File) suffix in the `Results_<datestamp>` directory. At the end of a compliant test run, the average throughput scores for each workload are recorded in the `Score_N_Tile_Test.txt` file (where *N* is the number of tiles in the test) along with the normalized scores and the composite VMmark metric.

The `Score_N_Tile_Test.txt` file, a sample of which is included below, contains additional details of the test, such as duration, start time, and end time as well as:

- `TILE_N_Scores`: The actual workload scores for each VM during each of the three test phases (p0, p1, and p2).
- `TILE_N_Ratios`: The scores normalized against the VMmark reference platform.
- `TILE_N_QoS`: Quality of Service (QoS) metrics for each workload.
  - For DVD Store workloads, this metric represents average application latencies in milliseconds. When the average application latencies exceed the workload's QoS limit, the run is marked non-compliant, as indicated by an asterisk for the test phase in question.
  - For Weathervane Static and Weathervane Elastic, two metrics are displayed, separated by a bar:
    - The metric to the left of the bar represents the normalized average response times for all the various operation types Weathervane performs during a run. Each operation's response time is normalized against that operation's QoS limit.

      When any Weathervane operation type's 99th percentile response time exceeds that operation's QoS limit, an asterisk on the relevant test phase marks the run as non-compliant.
    - The metric to the right of the bar is related to the percentage of operations that failed QoS requirements. VMmark calculates the percentage of operations within each operation type that failed QoS requirements then displays in this position the maximum percentage across any operation type. When this value is greater than 1.0, an asterisk on the relevant test phase marks the run as non-compliant.
    - Additionally, to ensure benchmark consistency from run to run, the proportion of each operation type in the overall mix of operations must match a proportion requirement. When this proportion is violated, a plus sign on the relevant test phase marks the run as non-compliant.

The composite VMmark score at the bottom of the `Score_N_Tile_Test.txt file` is the median of the sums of each phase's geometric mean per tile.

The `VMmark3-Graphs.html` file plots the throughput and quality of service (QoS) results of each VMmark workload over time for each tile. This provides an in-depth and visually intuitive look at workload performance during the run. Compliant workloads are graphed in blue and non-compliant workloads are graphed in red, so you can easily isolate characteristics of non-compliant workloads and can compare performance across tiles. Note that the plotted workload includes the ramp-up and ramp-down periods. The `VMmark3-Graphs.html` file also contains details of the test, such as duration, start time, and end time, so that the most important information about the run is accessible in one file.

The `STAX_Job_N_User.log` uses a default format that can be awkward to read. An easier-to-read processed version, `STAX_Job_N_User-Parsed.log`, is included in the results directory. The only difference between the two is their format.

Sample VMmark Results File

The below sample VMmark results file is from a VMmark 3.0 run on the reference system described in "Reference Scores" on page 28.

```
VMmark 3.0 TileScore : v1.0 05222017
Computing_results_for_test_with_tile_count: 2 ...
Esxtop Power Mode Enabled
First Sample: 1493677260 Mon May  1 15:21:00 2017
Info: 1 : 1493677320 : 2 : 6
Run_start 1493677260  : Mon May  1 15:21:00 2017
Start_time 1493677320 : Mon May  1 15:22:00 2017
End_time 1493687880 : Mon May  1 18:18:00 2017
Run_end 1493688240 : Mon May  1 18:24:00 2017
Duration_in_minutes : 176.00
Steady_state_start 1493679120 : Mon May  1 15:52:00 2017
Steady_state_end 1493686320 : Mon May  1 17:52:00 2017
Phase_0_begin 1493679120 : Mon May  1 15:52:00 2017
Phase_1_begin 1493681520 : Mon May  1 16:32:00 2017
Phase_2_begin 1493683920 : Mon May  1 17:12:00 2017


TILE_0_Scores:  WeathervaneAuction  WeathervaneElastic    DVDStoreA     DVDStoreB     DVDStoreC      Standby
p0                         3598.91              571.02       756.23        517.23        368.20         1.00
p1                         3599.75              569.29       775.95        519.62        347.45         1.00
p2                         3600.91              572.15       743.70        510.40        346.50         1.00

TILE_0_Ratios: WeathervaneAuction  WeathervaneElastic  DVDStoreA  DVDStoreB  DVDStoreC  Standby  Geo.Mean
p0                           1.00                1.00       1.03       1.03       1.06     1.00      1.02
p1                           1.00                0.99       1.06       1.04       1.00     1.00      1.02
p2                           1.00                1.00       1.01       1.02       1.00     1.00      1.01

TILE_0_QoS:    WeathervaneAuction% WeathervaneElastic%    DVDStoreA     DVDStoreB     DVDStoreC
p0                     0.54 | 0.03         1.14 | 0.78      1538.08       1816.89       2071.95
p1                     0.53 | 0.00         1.13 | 0.78      1414.18       1685.71       1946.58
p2                     0.52 | 0.00         1.24 | 0.94      1594.79       1862.68       2118.15


TILE_1_Scores:  WeathervaneAuction  WeathervaneElastic    DVDStoreA     DVDStoreB     DVDStoreC      Standby
p0                         3599.09              573.55       741.67        480.98        336.50         1.00
p1                         3600.19              571.90       745.12        483.27        324.80         1.00
p2                         3592.24              572.89       723.80        494.98        348.38         1.00

TILE_1_Ratios: WeathervaneAuction  WeathervaneElastic  DVDStoreA  DVDStoreB  DVDStoreC  Standby  Geo.Mean
p0                           1.00                1.00       1.01       0.96       0.97     1.00      0.99
p1                           1.00                1.00       1.01       0.97       0.94     1.00      0.98
p2                           1.00                1.00       0.99       0.99       1.00     1.00      1.00

TILE_1_QoS:    WeathervaneAuction% WeathervaneElastic%    DVDStoreA     DVDStoreB     DVDStoreC
p0                     0.52 | 0.00         1.07 | 0.84      1619.87       1955.02       2234.74
p1                     0.51 | 0.00         1.15 | 0.85      1582.86       1934.51       2181.29
p2                     0.50 | 0.00         1.07 | 0.82      1665.33       1959.29       2256.10

p0_score =   2.01
p1_score =   2.00
p2_score =   2.00

Infrastructure_Operations_Scores:       vMotion      SVMotion      XVMotion        Deploy
Completed_Ops_PerHour                     25.00         18.00         14.00          8.00
Avg_Seconds_To_Complete                   22.56        193.30        254.38        407.23
Failures                                   0.00          0.00          0.00          0.00
Ratio                                      0.96          1.00          0.78          1.00
Number_Of_Threads                             1             1             1             1


EsxtopPower_Results:
p0      Avg_Watts        Target
           300.46        prme-perf-vmmark3.eng.vmware.com
```

```
          284.06       prme-perf-vmmark4.eng.vmware.com
p1     Avg_Watts       Target
          299.63       prme-perf-vmmark3.eng.vmware.com
          286.46       prme-perf-vmmark4.eng.vmware.com
p2     Avg_Watts       Target
          300.50       prme-perf-vmmark3.eng.vmware.com
          287.33       prme-perf-vmmark4.eng.vmware.com

Warnings Messages::
  p0 : WeathervaneAuction0 Exceptions : 1
  p0 : WeathervaneElastic0 Exceptions : 4
  p1 : WeathervaneAuction0 Exceptions : 1
  p1 : WeathervaneElastic0 Exceptions : 1
  rampdown : WeathervaneAuction0 Exceptions : 4
  p0 : WeathervaneAuction1 Exceptions : 2
  p0 : WeathervaneElastic1 Exceptions : 2
  p1 : WeathervaneAuction1 Exceptions : 2
  p1 : WeathervaneElastic1 Exceptions : 1
  p2 : WeathervaneAuction1 Exceptions : 2
  p2 : WeathervaneElastic1 Exceptions : 5
  rampdown : WeathervaneElastic1 Exceptions : 1

Summary ::
Run_Is_Compliant
Median_Phase : p2

Unreviewed_VMmark3_EsxtopPower_Avg_Watts :   587.83
Unreviewed_VMmark3_Applications_Score    :     2.00
Unreviewed_VMmark3_Infrastructure_Score  :     0.93
Unreviewed_VMmark3_Score                 :     1.79
Unreviewed_VMmark3_Power_Efficiency*     :   3.0412

* This is NOT a compliant VMmark 3 Power Measurement and cannot be compared to VMmark 3 PPKW metrics.

Workload Graphs generated, see VMmark3-Graphs.html
```

Creating and Submitting a Benchmark Results Package

Preparing VMmark results for publication according to the run and reporting rules involves the following tasks:

- Make sure your prime client is configured to log in without a password to every ESXi hosts in your VMmark environment, as described in Step 8 on page 49.
- Prepare for reporter post-processing.
- Enable reporting in the `VMmark3.properties` file and run a compliant test.
- Create a name-mapping file.
- Generate an HTML-formatted table of results and ratios.
- Edit the `disclosure.HTML` file.
- Submit the benchmark results for review.

The following sections detail these steps.

Prepare for Reporter Post-Processing

The VMmark reporter scripts postprocesses the `vm-support.tgz` and `vc-support.tgz` files to reduce their size and make them more manageable for VMmark users. Despite efforts to minimize the resource requirements for this process, it nevertheless requires more than 3GB of disk space on the prime client. The actual amount of disk space required is highly variable; the process could require significantly more than 3GB (which would typically occur in environments with a large number of ESXi hosts.)

The VMmark harness uses the `VCscratchDir` parameter (in the `VMmark3.properties` file) to specify the directory where the reporter files should be post processed. If you run out of space on your prime client, rather than deleting files or extending a partition you can use the secondary disk on your prime client or you can create a third disk for this purpose.

To use the already-created secondary disk, follow these steps:

1 On the prime client open a terminal and execute the following command:

```
mkdir /root/VMmark3/results/scratch
```

2 Open the `VMmark3.properties` file and modify the `VCscratchDir` parameter:

```
VCscratchDir = /root/VMmark3/results/scratch
```

3 Save and close the `VMmark3.properties` file.

To create a third disk, see "Out of Disk Space or "tee: [..]/output_err.txt: No such file or directory" During Reporter Post-Processing" on page 92.

Enable Reporting in the `VMmark3.properties` File

When you are satisfied with your benchmark run and are ready to submit the results, you need to run a full test with reporting enabled. To do so:

1 Edit the `VMmark3.properties` file and set the following parameter:

```
Reporter = true
```

2 Rerun a compliant VMmark test.

The harness automatically gathers information about the workload virtual machines, the ESXi virtualization hosts, and the vCenter server, and places that information in the results directory.

Create a Name-Mapping File

Results to be submitted to VMmark for review must include a file showing the mapping of workloads to `.vmx` file names. A template is provided in the `reportingtools` directory for this purpose.

1. On the prime client change to the appropriate results directory:
   `cd ~/VMmark3/results/Results_`*`datestamp`*
   (where *`datestamp`* refers to the results to be submitted).

2. Copy the template `VM-name-map.txt` file from the `tools` directory to the `results` directory:
   `cp ../../tools/VM-name-map.txt .`

   NOTE This file is not present in the beta release.

3. Edit the `VM-name-map.txt` file, entering in the right column the path and name of each `.vmx` file that corresponds to the workload in the left column.

   The relevant section of the resulting file will look something like this example from a one-tile system:

```
Workload Name  .vmx File Name
------------------------------------------
PrimeClient    ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/PrimeClient/PrimeClient.vmx
vmmark3-template ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/vmmark3-template-01062017/
               vmmark3-template-01062017.vmx

Client0        ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/Client0/Client0.vmx
DS3DB0         ./vmfs/volumes/57604dc9-c75c8d7b-0e53-ac162d70c6f8/DS3DB0/DS3DB0.vmx
DS3WebA0       ./vmfs/volumes/57604dc9-c75c8d7b-0e53-ac162d70c6f8/DS3WebA0/DS3WebA0.vmx
DS3WebB0       ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/DS3WebB0/DS3WebB0.vmx
DS3WebC0       ./vmfs/volumes/57604dc9-c75c8d7b-0e53-ac162d70c6f8/DS3WebC0/DS3WebC0.vmx
AuctionLB0     ./vmfs/volumes/56445811-4fe1a2e8-8695-ecf4bbd14dc0/AuctionLB0/AuctionLB0.vmx
AuctionNoSQL0  ./vmfs/volumes/56445825-6356e060-8abb-ecf4bbd14dc0/AuctionNoSQL0/AuctionNoSQL0.vmx
AuctionAppA0   ./vmfs/volumes/5644571a-174b5ecc-da03-ecf4bbd14dc0/AuctionAppA0/AuctionAppA0.vmx
AuctionAppB0   ./vmfs/volumes/564457bd-be88f05a-01e0-ecf4bbd14dc0/AuctionAppB0/AuctionAppB0.vmx
AuctionDB0     ./vmfs/volumes/56445825-6356e060-8abb-ecf4bbd14dc0/AuctionDB0/AuctionDB0.vmx
AuctionMSQ0    ./vmfs/volumes/56445811-4fe1a2e8-8695-ecf4bbd14dc0/AuctionMSQ0/AuctionMSQ0.vmx
AuctionWebA0   ./vmfs/volumes/564457bd-be88f05a-01e0-ecf4bbd14dc0/AuctionWebA0/AuctionWebA0.vmx
AuctionWebB0   ./vmfs/volumes/564457da-42c1408c-fb94-ecf4bbd14dc0/AuctionWebB0/AuctionWebB0.vmx
ElasticLB0     ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/ElasticLB0/ElasticLB0.vmx
ElasticAppA0   ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/ElasticAppA0/ElasticAppA0.vmx
ElasticAppB0   ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/ElasticAppB0/ElasticAppB0.vmx
ElasticDB0     ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/ElasticDB0/ElasticDB0.vmx
ElasticWebA0   ./vmfs/volumes/57604dc9-c75c8d7b-0e53-ac162d70c6f8/ElasticWebA0/ElasticWebA0.vmx
ElasticWebB0   ./vmfs/volumes/57604dc9-c75c8d7b-0e53-ac162d70c6f8/ElasticWebB0/ElasticWebB0.vmx
Standby0       ./vmfs/volumes/57604df6-8630cf20-a345-ac162d70c6f8/Standby0/Standby0.vmx
```

4. Save the edited `VM-name-map.txt` file.

Generate an HTML-Formatted Table of Results and Ratios

After successfully completing a benchmark run with reporting enabled, the first step in preparing for publication is to generate an HTML-formatted table of results and ratios. This table will later be inserted into the full disclosure report.

1. On the prime client change to the appropriate results directory:
   `cd ~/VMmark3/results/Results_`*`datestamp`*
   (where *`datestamp`* refers to the results to be submitted).

2. Issue the command:
   `../../tools/tilescore2html.sh` *`n filename`*
   (where *`n`* is the number of tiles included in the results to be formatted and *`filename`* is the desired name of the HTML file (the `.html` will be automatically appended)).

   NOTE VMmark 3.x supports running the `tilescore.pl` and `tilescore2html.sh` scripts only on the prime client. These scripts are located in the `~/VMmark3/tools` directory.

Depending on the entries in your `VMmark3.properties` file, this command will generate a *`filename`*`.html` file (in all cases), a *`filename`*`_serverptd.html` file (if the run was configured to include server power), and a *`filename`*`_allptd.html` file (if the run was configured to include server and storage power).

If you generated one or both of the file types that include power measurement (*`filename`*`_serverptd.html` or *`filename`*`_allptd.html`), but want to create a submission that doesn't include the data from all the generated files, you can delete the file (or files) you don't want included.

Edit the `disclosure.HTML` File

WARNING See "Required Disclosure Report Updates for Security Mitigations" on page 108 for additional disclosure requirements.

After generating the HTML-formatted table of results and ratios, the next step is to create a full disclosure report.

1 On the prime client change to the appropriate results directory:
`cd ~/VMmark3/results/Results_`*`datestamp`*
(where *`datestamp`* refers to the results to be submitted).

2 Copy the template `disclosure.html` file from the `tools` directory to the `results` directory:
`cp ../../tools/disclosure.html .`

3 Edit the `disclosure.html` file.

a If the submission includes PTDaemon power measurement (that is, if it is a Server Power-Performance or Server and Storage Power-Performance submission) replace `VMmark V3.1 Score` with either `VMmark V3.1 Server PPKW Score` or `VMmark V3.1 Server and Storage PPKW Score` (as appropriate) in the disclosure file header.

b Replace `### @ ### Tiles` with the score and tile count from the *`filename`*`.html` file, the *`filename`*`_serverptd.html` file, or the *`filename`*`_allptd.html` file.

c In the section titled `Performance`, replace `###` with the contents of one of the HTML files you generated with `tilescore2html.sh` as described in "Generate an HTML-Formatted Table of Results and Ratios" on page 74.

d Search for all remaining instances of `###` and replace `###` with the appropriate details of your system configuration and data about the benchmark run. Keep in mind the following points:

- For benchmark runs with non-identical servers, make additional copies of the relevant sections of the template `disclosure.html` file as needed to describe the servers.
- For additional information about completing the Security Mitigations section of the disclosure, see Appendix C, "Security Mitigations," on page 103.
- `Total Number of` Virtual `Clients` should include the prime client.
- `Networking Notes` must disclose assignment of virtual machines and vmnics to virtual switches.
- For benchmark runs with multiple non-identical storage solutions, make additional copies of the relevant sections of the template `disclosure.html` file as needed to describe the storage.
- All disclosures of non-default values or settings should also disclose the default value or setting:
`<parameter> = <non-default value> (default <default value>)`
- If the submission includes PTDaemon power measurement, disclose the assignment of power meters to metered devices (that is, servers, external storage devices, switches, etc.), and disclose the voltage, frequency, and phase (that is, single-phase or three-phase) of the power source.
- `MM-DD-YYYY` strings are included in the template as examples and should be replaced with the relevant month, day, and year.

e Save the `disclosure.html` file, naming it according to the test type as follows:
For Performance-only tests, use `<filename>.html`
For Server Power-Performance tests, use `<filename>_serverptd.html`
For Server and Storage Power-Performance tests, use `<filename>_allptd.html`

NOTE A single VMmark benchmarking run can be used for multiple submission types if the required data was measured during the run.

That is, if server power was measured with PTDaemon, the run can be used for Performance Only and Server Power-Performance. If both server *and* storage power were measured, the run can be used for all three test types (Performance Only, Server Power-Performance, or Server and Storage Power-Performance).

When using a single VMmark benchmarking run for multiple submission types, you should create a separate disclosure file for each desired submission type.

Submit the Benchmark Results for Review

Once all the necessary files are in the appropriate results directory on the prime client, you are ready to submit the benchmark results for review. This involves the following steps:

- Compress the results folder.
- Generate an MD5 sum on the submission file.
- FTP the file to the submission site.
- To receive an emailed acknowledgment, email `benchmark@vmware.com`.

For specific submission instructions send email to `benchmark@vmware.com`.

Optional Configurations and Settings A

This appendix provides information about optional configurations for many of the programs and utilities used to perform the VMmark benchmarking tests. It consists of the following section:

Running a Subset of the Workloads (a Partial Tile)

Any test runs conducted with a subset of the workload suite (that is, with one or more partial tiles) do not count as valid runs for the purposes of calculating a VMmark benchmark score. There might be times, however, such as debugging, or measuring the effects of various workloads on the overall performance, when it is desirable to perform partial-tile runs.

Partial-tile runs can be started manually, by starting only those workloads desired, or can be started using the VMmark harness. To start a partial-tile run using the VMmark harness, modify the `VMMARK3.properties` file, located in the `/root/VMmark3` directory on the prime client, changing the `WorkloadList and InfrastructureList` variables to include only those workloads you want to run.

NOTE **Partial-tile runs are only for diagnostic or research purposes and must not be used to calculate VMmark Benchmark results.**

Manually Generating a VMmark Score

In some cases you may wish to manually generate a VMmark score. This might be useful if an error occurred during the automatic score generation, for example. This will typically not be a compliant score, but would instead be useful for diagnostic or research purposes.

Using the `tilescore.pl` command options, you can generate a full score, a score that excludes some tiles, or a score that excludes some workloads.

From a terminal window on the prime client, run `~/VMmark3/tools/tilescore.pl -h` to display a list of command options. Of most interest in this case will likely be the various skip options (`-W`, `-D`, `-E`, `-V`, `-S`, `-X`, `-C`, `-Y`, `-T`, `-P`) and the option to score a subset of the tiles (`-t`).

NOTE VMmark 3.x supports running the `tilescore.pl` and `tilescore2html.sh` scripts only on the prime client. These scripts are located in the `~/VMmark3/tools` directory.

Adjust StafMaxThreads Parameter

In order to speed up various VMmark operations such as results processing, a new parameter, `StafMaxThreads`, has been added to the `STAF.cfg` file. This parameter sets the maximum number of simultaneous threads for all non-run parallel operations (that is, operations that can be accomplished in parallel, other than those during benchmark runs, which are never throttled).

The default value of this parameter is the larger of 100 or the number of tiles in the VMmark environment.

Because this value is automatically scaled up with the number of tiles, there will not likely ever be a need to manually increase it. However, in an environment with fewer than 100 tiles, if slowdown is experienced performing operations such as ping, this might be due to too many concurrent STAF operations; in this case, decreasing this value might improve performance.

Add a Web Browser to the Prime Client

For licensing reasons, as of VMmark 3.1 the VMmark template no longer includes a web browser. If you want to install a browser you can do so by running `yum install firefox`.

Back Up the DVD Store 3 Database

The most time-consuming part of preparing a VMmark 3 environment is the creation of the first DVD Store 3 database (that is, the database portion of the DS3DB virtual machine for tile 0, or DS3DB0). Depending on your environment, this can take 12 hours or more. (Subsequent tiles are cloned from the first one, and their creation is thus far faster.)

It's possible when upgrading to a new VMmark 3 version to avoid the need to repeat this process by using the `ds3backup_restore.sh` script to back up the existing DS3 database, then to restore the backup to a newly-created DS3DB virtual machine.

Using the `ds3backup_restore.sh` Script

Follow these steps to preserve the DS3 database when upgrading or reinstalling VMmark 3.x:

1 With the DS3DB0 virtual machine powered on, copy two files from the DS3DB0 virtual machine to the prime client by executing the following two commands on the prime client:

```
scp root@ds3db0:/ds3/mysqlds3/build/mysqlds3_cleanup_100GB.sql /ds3/mysqlds3_cleanup_100GB.sql
scp root@ds3db0:/ds3/data_files/prod/inv.csv /ds3/inv.csv
```

2 Shut down the DS3DB0 virtual machine.

3 Rename the DS3DB virtual machine to DS3DB0_backup.

4 Optionally, edit the `ds3backup_restore.sh` script to reflect your environment (as described in "Customization and Options for the `ds3backup_restore.sh` Script," below). If you don't edit this file, the script will prompt you for the information it requires.

5 Make a backup of the DS3 database using the `ds3backup_restore.sh` script:
`~/VMmark3/tools/ds3backup_restore.sh`
then select option **`B`** and follow the prompts. For guidance about what's required here, see "Customization and Options for the `ds3backup_restore.sh` Script," below.

6 Edit the `VMmark3.properties` file, setting `ProvisioningDBsize = 1`.

7 Execute the VMmark3 provisioning service for tile0.

8 After the DS3DB0 virtual machine is completely provisioned, copy two files from the prime client to the the newly provisioned DS3DB0 virtual machine by executing the following two commands on the prime client:

```
scp /ds3/mysqlds3_cleanup_100GB.sql root@ds3db0:/ds3/mysqlds3/build/mysqlds3_cleanup_100GB.sql
scp /ds3/inv.csv root@ds3db0:/ds3/data_files/prod/inv.csv
```

9 Power off the newly created DS3DB0 virtual machine.

10 Optionally, verify that the variables `ds3dbvm` and `ds3dbvmdk` are set to the correct settings for the newly created DS3DB0 virtual machine (as described in "Customization and Options for the `ds3backup_restore.sh` Script," below). If you don't edit this file, the script will prompt you for the information it requires.

11 Restore the DS3 database using the `ds3backup_restore.sh` script.
`~/VMmark3/tools/ds3backup_restore.sh`
then select option **`R`** and follow the prompts. For guidance about what's required here, see "Customization and Options for the `ds3backup_restore.sh` Script," below.

12 Edit the `VMmark3.properties` file, restoring the default setting of `ProvisioningDBsize = 100`.

13 Verify that the newly created DS3DB0 virtual machine with the DS3 database restored from backup is functioning properly as follows:

a Power on the newly created DS3DB0 virtual machine.

b Remote logon to DS3DB0 virtual machine with SSH and execute `DS3_checker.sh` from the `/root/VMmark3/client/DS3DB` directory.

c Verify that you see the following output:
"It looks like database DS3 is built correctly and in initialized state."

14 After verifying that the newly created DS3DB0 with the DS3 database restored from backup is functioning properly, delete the DS3DB0_backup virtual machine.

Customization and Options for the `ds3backup_restore.sh` Script

The `ds3backup_restore.sh` script is located on the prime client in the `~/VMmark3/tools/` directory. Before running the script you can edit it (using a standard text editor) to uncomment and preload variables that control the script's behavior. Alternately, if you leave the variables commented out, you'll be prompted for them when the script is run.

These variables are:

- The ESXi host to use for backup or restore:
  `dbbackuphost =`
- The LUN to use for DS3 database backup files:
  `ds3backup_lun =`
  This is just the LUN name, not the full path. For example, if the LUN's full path is `/vmfs/volumes/VMstorage1`, set this to `VMstorage1`.
- The directory to use for DS3 database backup files:
  `ds3backup_dir =`
  This is the full path to the directory. For example, if `DS3BackupDir` is used, set this to `/vmfs/volumes/VMstorage1/DS3BackupDir`.
- The DS3DB0 source VM LUN:
  `ds3source_lun =`
  This is just the LUN name, not the full path. For example, if the LUN's full path is `/vmfs/volumes/VMstorage2`, set this to `VMstorage2`.
- The DS3DB0 source VM directory name:
  `ds3dbvm =`
  This is just the directory name, not the full path. For example, if DS3 DB is located on `/vmfs/volumes/VMstorage2/DS3DB0`, set this to `DS3DB0`.
- The DS3DB0 250GB DS3 database source .vmdk file:
  `ds3dbvmdk =`
  You can use the vSphere Client UI to determine the name of the .vmdk file for the 250GB second hard disk, then use the name of that file without the .vmdk extension. Select the .vmdk file without `-flat`. For example, if the source file is `[VMstorage2]/DS3DB0/DS3DB0_1.vmdk`, set this to `DS3DB0_1`.

Executing the `~/VMmark3/tools/ds3backup_restore.sh` script will present three options:

```
Select [H] to display the help message.
Select [B] to backup the DS3 DB to a backup directory.
Select [R] to restore a DS3 DB backup to a new DS3DB0 directory.
```

Selecting `H` will provide the following information:

```
=============================================================================
INTRODUCTION
------------
Script to backup or restore the DS3DB0 DS3 DB. The DB is normally created during
the initial provisioning and can take 10 - 20 hours to build based on system
being used. The backup or restore procedure should take less than 20 minutes
based on the system being used.

The user can select either:
- backup a completed DS3DB0 DS3 DB disk to a directory on the same ESXi host.
- restore a backup of the DS3DB0 VM DB disk to a DS3DB0 VM directory.

====================================================================================
##  NOTE: The DS3DB0 VM needs to be shutdown prior to execution of this script.  ##
##        Otherwise, a 'Failed to lock the file (XXXXXX)' error will result.     ##
===================================================================================

================================================================================================
VARIABLES
---------
User can preset these variables before execution; otherwise, you will be prompted
for input at run time.

    dbbackuphost = ESXi host to use for backup or restore.
    Example: sut1-esxihost.eng.vmware.com

    ds3backup_lun = Storage LUN to use for backup files.
    Example: If the LUN is /vmfs/volumes/VMstorage1 then use VMstorage1.

    ds3backup_dir = Directory on the backup storage LUN used for the DS3 DB backup.
    Example: If DS3BackupDir is used then /vmfs/volumes/VMstorage1/DS3BackupDir is the working
    directory.

    ds3source_lun = Source LUN where the existing DS3 DB to be backed up resides.
    Example: If the LUN is /vmfs/volumes/VMstorage2 then use VMstorage2.

    ds3dbvm = Source directory on LUN of the existing DS3 DB VM to be backed up.
    Example: If DS3 DB is located on /vmfs/volumes/VMstorage2/DS3DB0, then use DS3DB0.

    ds3dbvmdk = DS3 DB source VM vmdk file (250GB hard disk) WITHOUT the .vmdk extension.
    Example: You can use the vSphere Client UI to determine the name of the Hard disk 2 250GB
    .vmdk file where Disk File  = [VMstorage2]/DS3DB0/DS3DB0_1.vmdk use DS3DB0_1.

================================================================================================
```

Deploying an Updated Prime Client

The VMmark download page might offer updates to VMmark 3.x, such as bug fixes or security updates, that require creating a new prime client but don't require re-deploying the workload virtual machines.

If such an updated prime client is required, but you want to preserve data (results, configuration files, etc.) from your existing prime client, follow these steps:

1 Rename your existing prime client virtual machine (to **`PrimeClient-old`**, for example) by right-clicking on it and selecting **Rename...**.

2 If you're using a second NIC with a static IP address to communicate over the isolated network to the environment under test, change that IP address so it doesn't conflict with the new prime client:

   a Change to the network scripts directory:
   ```
   cd /etc/sysconfig/network-scripts
   ```

   b Open the appropriate network script in an editor. For example, for a network named eno33559296:
   ```
   vim ifcfg-eno33559296
   ```

   c Change the static IP address (the new prime client will use the existing IP address). Note both the original and new IP addresses.

   d Reboot the prime client.

3 Follow the steps in "Obtain the VMmark Template and Create the Prime Client" on page 46 to deploy a new template and create a new prime client. If you're using a second NIC with a static IP address, use the IP address from the old prime client so the existing workload and client virtual machines will still be able to communicate with the new prime client.

4 With both prime clients powered on and configured to have unique IP addresses, copy the data you want from the old prime client to the new prime client as follows:

   From the new prime client, use `scp` to copy the desired data from the old prime client. For example:

   - To copy the `results` directory from the old prime client to a `results-old` directory on the new prime client:
     ```
     mkdir ~/VMmark3/results-old/
     scp -r PrimeClient-old-IP:~/VMmark3/results/ ~/VMmark3/results-old/
     ```
     (where *`PrimeClient-old-IP`* is the IP address of the old prime client that you set in Step 2c above).

   - To copy the `VMmark3.properties` file from the old prime client to a `VMmark3.properties.old` file on the new prime client:

```
scp PrimeClient-old-IP:~/VMmark3/VMmark3.properties ~/VMmark3/VMmark3.properties.old
```

   (where *`PrimeClient-old-IP`* is the IP address of the old prime client that you set in Step 2c above).

   To use parameters from your `VMmark3.properties.old` file, copy the original values from that file and paste them into the new `VMmark3.properties` file. This will ensure that any new parameters and documentation are available for future runs.

5 Once you're sure you've retrieved from the old prime client all the data you want, it can be powered off (and optionally deleted, if you need the storage capacity).

Redeploying a VMmark Environment

To redeploy your VMmark 3.1 environment, whether for an upgrade or for any other reason, follow the steps in this section.

1 If desired, make a backup of the DS3 database, as described in "Back Up the DVD Store 3 Database" on page 79. This is optional, but for environments with slow storage subsystems can save time during the redeployment.

2 If desired, preserve the data in the prime client, as described in "Deploying an Updated Prime Client" on page 82. This is optional, but can preserve existing VMmark results and `VMmark3.properties` file customizations.

3 Delete from your environment (or rename) all workload and tile virtual machines (including the prime client).

NOTE If you choose to back up the DS3 database, you might want to rename rather than delete the DS3DB0 virtual machine, then delete it after the restore process is successful. (This process is described in the "Back Up the DVD Store 3 Database" section.)

4 Create a new prime client, as described in "Obtain the VMmark Template and Create the Prime Client" on page 46.

5 Provision additional VMmark tiles, as described in "Provision VMmark Tiles" on page 51.

Using Power Meters With VMmark

VMmark 3.x allows the optional measurement of power usage while running its full application and infrastructure workloads across a multi-host cluster. This section describes the power meters that can be used for that measurement and how to install and configure them.

For more information about power measurement in VMmark 3.x, see "VMmark Power Measurement" on page 19.

Compatible Power Meters

For VMmark tests that will include server power measurement, you will need one or more compatible power meters for the servers in the test environment. For VMmark tests that will include server and storage power measurement, you will also need one or more compatible power meters for the storage device (or devices) in the test environment.

NOTE As long as it is not loaded beyond its rated limits, a single power meter can supply power to more than one server or to more than one storage device. For tests that are measuring Server and Storage Power-Performance, the server power and the storage power must be measured separately, thus requiring at least two power meters (one or more for server power and one or more for storage power).

For VMmark tests that will include server power measurement, each server running VMmark workload virtual machines will need to be connected to a compatible power meter. For VMmark tests that will include server and storage power measurement, each storage device accessed by the VMmark workload virtual machines will need to be connected to a compatible power meter.

VMmark is compatible with any power meter supported by version 1.8 of the SPEC PTDaemon (Power Temperature Daemon). A list of these meters can be found on the SPEC web site:
http://www.spec.org/power/docs/SPECpower-Device_List.html

While VMmark should work with any of these meters, testing has only been performed with the Yokogawa WT210 power meter.

Each power meter must be connected to a physical client system. This can be the prime client, a client associated with a VMmark tile, or even some other physical system.

NOTE Depending on the connection type used by the chosen power meters, the number of physical ports on a client system might become a bottleneck. For example, many server systems have only one RS-232 serial port. With a power meter that uses an RS-232 connection, such as the Yokogawa WT210, this allows only one such meter to be directly connected to each client.

Numerous third-party products are available, however, to allow one or more serial devices, such as power meters, to be connected to a single USB port. While we expect that most such devices could be used to connect power meters to VMmark clients, we have performed only limited testing.

For instructions to install and configure a power meter, see:
http://www.spec.org/power/docs/SPEC-Power_Measurement_Setup_Guide.pdf

Included in the sections below are instructions for the use of one common power meter, the Yokogawa WT210.

Configuring VMmark to Use Power Meters

After configuring your power meter and connecting it to the host running your prime client, perform the following steps to configure VMmark to use it:

1 Shut down the prime client.

2 In vCenter Server, right click the prime client, select Edit Settings, then select **Add** > **Serial Port** > **Use physical serial port on the host**.

3 Select either `/dev/char/serial/uart0` or `/dev/char/serial/uart1`.
(This should be the UART corresponding to the physical serial port to which you connected the power meter.)

4 Power on the prime client.

5 In a terminal window on the prime client, run:

```
dmesg | grep tty
```

6 Look for a line similar to the following:

```
ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
```

7 Convert the hex number (`0x3f8` in the above example) to decimal by running the command:

```
printf "%d\n" YourHexNumber
```

8 View the PTD help output by running the command:

```
/root/VMmark3/client/ptd/ptd-linux-x86 -h
```

9 In the PTD help output find the number representing the Power Analyzer device type you're using. For example, a Chroma 66202 power meter would be **31**.

10 Make the following edits to the `VMmark3.properties` file:

a Enter the decimal number you got from the `printf` command in the `PTD_Ports` field.

b Enter the tty number (`ttyS0` in the above example) in the `PTD_COMs` field as a device relative to root. So for `ttyS0` this would be:

```
/dev/ttyS0
```

c Enter the Power Analyzer device type in the `PTD_MeterTypes` field.

d Set `PTD = true`.

e Uncomment all lines in the PTD section that are above `Optional Settings - Defaults shown`.

Troubleshooting B

This appendix provides assistance in troubleshooting the VMmark Benchmarking tests. For further assistance, visit the VMmark section of the VMware Communities at:
http://communities.vmware.com/community/vmtn/general/performance/vmmark

Workload Troubleshooting

This section contains guidance for troubleshooting problems with specific workloads.

Deploy Remediation Warning

Deploy remediation is a series of attempts to correct for various issues found during the deploy operation. These issues might be failing to get an IP address, not having a connected vmnic, or not booting successfully. As systems start to saturate, it appears that this can happen with more frequency.

When the VMmark harness detects any of these issues, it tries to salvage the current deploy operation (as opposed to marking it a failure) but doing so takes time. Extra time means a slower deploy operation, which results in a lower benchmark score.

Each instance of deploy remediation (either start ping or final ping) is counted as one warning; each deploy operation can thus have up to two warnings.

Error “VeryLow-Disk-Space”

If a Weathervane workload has a large number of errors in a VMmark run, it might fill the disk of its tile client virtual machine with error logs. If this occurs, a subsequent VMmark run might generate the error message “VeryLow-Disk-Space”.

If you encounter this problem, for each affected tile client virtual machine (as identified by the harness):

1 Manually log into the virtual machine.

2 Delete all file files from the /tmp directory.

```
rm -f /tmp/*
```

3 Reboot the virtual machine.

Once these steps have been performed for all affected virtual machines, start additional VMmark runs as usual.

DVD Store 3 Troubleshooting

This section includes guidance for troubleshooting DVD Store 3 operations.

”unable to ping DS3-DB” Error Message in STAX Window

Messages in the STAX window similar to ”unable to ping DS3-DB” are typically caused by networking issues in your environment. Make sure all VMs in the tile can ping each other.

NOTE For environments with advanced configurations, such as NSX enablement, you might need to review the networking settings for VM to VM communication (these are outside the scope of VMmark).

”Unknown database” Message in `ConfigDS3DB` Text File

If the `ConfigDS3DB` text file contains the following message:

```
Unknown database 'DS3'
```

delete all the DS3 VMs for that tile then recreate them as described in "Recreating All or Part of a Tile" on page 55.

Power Meter Troubleshooting

This section includes guidance for troubleshooting issues with power meters in VMmark 3.x.

Power Meter Current Display Screen Shows --oL-

If you see **--oL-** on a power meter's **Current** display screen, this is an indication that the measured current is outside that power meter's configured range. If your measured current falls outside the default current range of 2.0 to 4.0 Amps, make sure you've correctly set the `PTD_Ranges` variable in the `VMmark3.properties` file to reflect the expected maximum Amperage.

PTD Connection Times Out

Occasionally one or more of the PTD connections to the PTD_Clients will time out. When this happens the `ptd_stats.wrf` log file will show messages such as:

```
2013-04-08 14:09:21:779 [ERROR]: Attempt to connect to 'ptdclient45 failed. IOException
                encountered: java.net.ConnectException: Connection timed out: connect
2013-04-08 14:09:21:780 [ERROR]: Attempt to connect to 'ptdclient43 failed. IOException
                encountered: java.net.ConnectException: Connection timed out: connect
2013-04-08 14:09:21:781 [ERROR]: Attempt to connect to 'ptdclient44 failed. IOException
                encountered: java.net.ConnectException: Connection timed out: connect
```

With prime clients that have more than one NIC this error might be caused by the ptddriver attempting to communicate via the wrong NIC.

Follow these steps to correctly specify which NIC the ptddriver should use for each power meter:

1 Edit the prime client hosts file, adding an entry for the network interface the ptddriver should use. For example, such an entry might look like:
```
192.168.1.7 testbed
```

2 Within the `VMmark3.properties` file, edit the per meter settings to use the new hosts file entry. For example:
```
PTD_Client "ptdclient43 ptdclient44 ptdclient45"
PTD_LOCAL_Hostnames = "testbed testbed testbed"
PTD_LOCAL_Ports = "0 0 0"
```

In this example we're telling the ptddriver to communicate with each of three PTD_Clients (ptdclient43, ptdclient44, and ptdclient45) via the NIC identified in the hosts file as `testbed` (in this example, IP address 192.168.1.7) using port 0.

3 Save and close the `VMmark3.properties` file.

Miscellaneous Troubleshooting

This section includes guidance for troubleshooting miscellaneous issues.

Problems Deploying the VMmark Template

If you encounter issues deploying the VMmark template template from the original `.ova` file, make sure you're using the latest version of the HTML5-based vSphere Client rather than the Flash/Flex-based version (called the "vSphere Web Client"). If you still encounter issues, you can use VMware PowerCLI to deploy the template, as follows:

1 If it's not already present, install PowerShell on the system from which you wish to perform the installation.

PowerShell is a cross-platform (Windows, Linux, and macOS) automation and configuration tool. It can be obtained from various sources, including GitHub: https://github.com/PowerShell/PowerShell#get-powershell

NOTE PowerShell should not be installed in any of the VMmark virtual machines (the prime client, the tile clients, or the workload virtual machines).

2 Install the VMware PowerCLI module in PowerShell.

PowerCLI is available from VMware; here's version 11.2.0 (the latest version as of this writing): https://code.vmware.com/web/tool/11.2.0/vmware-powercli

NOTE Check the compatibility matrix to make sure PowerCLI will work in your operating system and with your version of PowerShell.

3 Use the Import-VApp cmdlet to deploy the template to a SUT host (as opposed to the SUT cluster).

NOTE For guidance on using Import-VApp see: https://code.vmware.com/docs/7788/doc/Import-VApp.html

All steps required to deploy the VMmark template and create the prime client can be accomplished using the Import-VApp cmdlet.

Problems Provisioning a VMmark Tile

If you encounter issues provisioning a VMmark tile, this could be due to problems with the deployed version of the template. To rule out such problems, redeploy the VMmark template from the original `.ova` file using PowerCLI, as described in "Problems Deploying the VMmark Template" on page 90.

Infrastructure Operations Troubleshooting

If you're having issues with the VMmark infrastructure operation routines, check the following:

- Make sure your vCenter Server is pingable from the prime client.
- On the vCenter Server, go to **Server Manager** > **Services**, and make sure **VMware VirtualCenter Server** is **Started**.
  It can sometimes take up five minutes for the service to respond to requests after restarting either the vCenter system or the service itself.
- Before starting the benchmark, verify that the time protocol on the vCenter Server is running and that the vCenter Server time is synchronized with the testing environment.

STAF and STAX Issues

STAF logs

Issues with STAF can sometimes be solved by reviewing the STAF log file. You can access the STAF log file with the following command:

```
journalctl _SYSTEMD_UNIT=startstaf.service
```

STAF Crashes When Reporter Runs Out of Threads

If STAF crashes, especially in very large environments, this might be due to an insufficient number of threads. You can try increasing the number of threads beyond the VMmark default of 200 by editing the following line in the prime client's `STAF.cfg` file:

```
service stax library JSTAF execute \
              C:\STAF\services\stax\STAX.jar OPTION J2=-Xverify:none \
              parms "numthreads 200 logtcelapsedtime enabled logtcstartstop enabled"
```

STAF Complains About Trust Level

If STAF complains that systems are at trust level 3, this suggests that the trust level settings in the `STAF.cfg` file are not being correctly processed. To correct this, make sure each `STAF.cfg` file contains the following:

```
trust machine 192.168.*.* level 5
```

(substituting the first two octets of the network to which the machines are connected).

STAF is Unable to Connect to a Particular Server

If STAF is unable to connect to a particular server, this might be due to one of the following causes:

- A communication or configuration problem.

  From a terminal window on the client, type:
  `staf` *`alias`* `ping ping`
  (where *`alias`* is the alias in the client's hosts file (for example, DS3WebA0)).

  The proper response is **PONG**.

- Dissimilar versions of STAF are installed.

  Make sure all clients and all workload virtual machines are running the same version of STAF.

- Review the `/etc/hosts` files on both systems for accuracy.

Guestinfo Collection Fails With Error Message "Get Guest Info did not run correctly"

If guestinfo collection consistently fails with error message "Get Guest Info did not run correctly", restart STAF on the prime client and on the affected virtual machine. If that fails, reboot the prime client.

To restart STAF on the prime client or on a workload or tile client virtual machine:

1. Access a shell prompt on the system in question:
   For the prime client, this will likely be a terminal window.
   For a workload or tile client virtual machine, you can use `ssh root@`*`VMname`* (where *`VMname`* is the name of the affected virtual machine).

2. Shutdown STAF:
   `staf local shutdown shutdown`

3. Wait 30 seconds, then restart STAF:
   `nohup /usr/local/staf/startSTAFProc.sh`

4. Verify that STAF is now running and that it has only recently started (which indicates that the shutdown was successful):
   `cat nohup.out | grep time`

Unable to Start STAX

If you get an error message when attempting to start STAX, make sure the your prime client has the Java version it had when it was created and has no other Java version.

Provisioning Issues

Provisioning Service Hangs During the Automation Phase

If the provisioning service hangs during the automation phase — as indicated by a lack of progress for more than ten minutes — this might be caused by IP address issues, such as the presence of a duplicate IP address on the network, or an incompatible subnet mask.

You can check for a duplicate IP address by running this command:
`ifup` *`enoXXXXXXXX`*
(where **`enoXXXXXXXX`** is the network name, typically `eno16777732`).

Watch for output reporting a duplicate IP address.

Benchmark Execution Issues

Out of Disk Space or "tee: [..]/output_err.txt: No such file or directory" During Reporter Post-Processing

As described in "Prepare for Reporter Post-Processing" on page 73, the post-processing operation can run out of disk space. To address this, you can use the already-created secondary disk (as described in that section), or you can create a third disk, as follows:

1 Create an additional virtual disk on the prime client:

NOTE The following steps can be performed while the prime client remains powered-on.

   a Right click on the prime client virtual machine in the vSphere Client and select **Edit Settings...**.

   b Under **New device**, select **New Hard Disk**, then click **Add**.

   c Choose whatever size you wish.

   d Click **OK**.

2 Partition and format the new disk.
```
fdisk /dev/sdc
```
(Respond to the prompts as follows: **`n`**, **`p`**, **`<Enter>`**, **`<Enter>`**, **`<Enter>`**, **`w`**.)

3 Format the new disk:
```
mkfs.ext4 –L scratch /dev/sdc1
```

4 Create a `scratch` directory under `/root/VMmark3/results`:
```
mkdir /root/VMmark3/results/scratch
```

5 Mount the new disk on `/root/VMmark3/results/scratch`:
```
mount /dev/sdc1 /root/VMmark3/results/scratch
```

6 To configure the system to automatically mount the new disk during boot, open `/etc/fstab` and add the following line:
```
/dev/sdc1 /root/VMmark3/results/scratch ext4 defaults 1 3
```

7 Open the `VMmark3.properties` file and modify the `VCscratchDir` parameter:
```
VCscratchDir = /root/VMmark3/results/scratch
```

8 Save and close the `VMmark3.properties` file.

The VMmark Harness Complains That Clients Are Out of Sync

If the VMmark harness complains that clients are out of sync, this is likely due to client clocks not being synchronized to within the allowable threshold.

Networking Issues

This section addresses network issues sometimes encountered while running VMmark.

Setting the Number of Ports on a Virtual Switch in ESXi

Virtual network switches in VMware ESXi start out with a fixed number of ports by default. If your test uses more than this number of ports, you might experience difficult-to-diagnose errors. Increasing the number of ports should alleviate these issues.

Follow these steps to determine the number of ports a virtual switch is configured with and increase that number if required:

1 Within the vSphere Client Inventory pane, select the ESXi host.

2 Select the **Configuration** tab.

3 In the **Hardware** window, select **Networking**.

4 To the right of the virtual switch to be configured choose **Properties**.

5 Under the **Ports** tab, the **vSwitch Properties** section lists the number of ports.

6 If the number of ports listed is fewer than your setup needs, add ports as follows:

    a Click **Edit**

    b Increase the number of ports.

    c Click **OK**.

    NOTE The changes will take effect the next time the ESXi host system is rebooted.

7 Click **Close**.

Setting the Network Name

If a virtual machine is unable to connect to the network, make sure that it is configured to use the correct network (that is, the correct virtual switch), that the virtual switch is configured to the correct physical NIC, and that the NIC is connected to the intended network.

For further information, refer to the relevant VMware product documentation.

Setting Network to Connect at Power-on

If a virtual machine is unable to connect to the network, make sure that its network adapter is configured to connect at power on. This setting can be found in **Summary** > **Commands** > **Edit Settings** > **Hardware** > **Network Adapter 1**.

Strange Issues on the Prime Client (Networking Related or Otherwise)

Strange issues on the prime client (networking related or otherwise) could be caused by the SELinux security policy. To address these:

1 Open `/etc/sysconfig/selinux`.

2 Change "SELINUX=enforcing" (the default) to "SELINUX=disabled".

3 Reboot the prime client VM.

Results Troubleshooting

Missing vcsupport*.tgz file from Results Directory

If the vcsupport*.tgz file is missing from the results directory, this might be caused by pschealthd core files (core.pschealthd.*) gathering in the /var/core/ directory.

As a workaround, you can set up an automated cron job to delete these files as follows:

1 SSH to the vCenter Server Appliance.

2 Run the following:
```
sed -i -e 's/password-auth/system-auth/g' /etc/pam.d/crond
```

3 Run the following:
```
crontab -e
```

4 Insert the following line:
```
10 * * * rm -rf /storage/core/core.pschealthd.*
```

5 Write the changes, then quit.

Another option would be to allow more time to collect files from the vCenter Server Appliance by increasing the VCsupportTimeOut value in the VMmark3.properties file. A reasonable value to try would be 4800 (double the default value of 2400).

Troubleshooting Low Scores or Compliance Issues

Low scores can be caused by a scarcity of resources — such as CPU capacity, memory, storage bandwidth, or network bandwidth — in the systems under test. Resource scarcity can show up as low VMmark scores or as Quality of Service (QoS) failures for one or more of the applications during the run. For information about interpreting QoS results, see “VMmark Results Files” on page 70.

To confirm whether this is due to resource contention, you can try running VMmark with a subset of the workloads, as described in “Running a Subset of the Workloads (a Partial Tile)” on page 77. If the QoS values improve, that suggests the issue was resource related.

Missing .wrf File

If you see the following error:

“Error: could not resolve start-time. Results data missing from one or more *.wrf files or time on clients not synchronized”

this might indicate a missing or invalid .wrf file (a workload results file).

To verify that it is, in fact, a missing or invalid .wrf file, and to determine which one it is, you can manually run the ~/VMmark3/tools/tilescore.pl script, omitting various tiles or workloads until the script completes without error, thus locating the problem.

Use tilescore.pl -h to display a list of command options. Of most interest in this case will likely be the various skip options (-W, -D, -E, -V, -S, -X, -C, -Y, -T, -P) and the option to score a subset of the tiles (-t).

NOTE VMmark 3.x supports running the tilescore.pl and tilescore2html.sh scripts only on the prime client. These scripts are located in the ~/VMmark3/tools directory.

Performing a Short Run

When first configuring a VMmark environment, it can be useful to perform one or more short VMmark test runs to make sure everything is functioning correctly. While useful for testing and diagnostic purposes, short runs are not compliant and do not produce publishable scores.

One situation in which these short runs are particularly helpful is when setting up the first tile. We recommend creating one complete tile (for example, Tile0) and making sure it is working smoothly, then making backups of this working configuration before cloning the workload virtual machines to create the additional tiles.

Turbo Mode Short Run

One way to configure a short run is to enable Turbo Mode in the `VMmark3.properties` file. Turbo Mode is a special VMmark running mode that will automatically configure the run to last 1800 seconds (30 minutes) and will then attempt to automatically post-process the results via `tilescore.pl`, marking any such results non-compliant.

Here is the relevant section of the `VMmark3.properties` file:

```
// Boolean TurboMode : Enables a quick VMmark3 run of 30 minutes, with automatic score
generation.  Useful for preliminary testing, not compliant for submissions : Default
false
// TurboMode = false
```

Manual Short Run

A short run can be manually configured by reducing the value of the RUNTIME variable in the `VMmark3.properties` file from its default of 10800 seconds (3 hours) to something no less than 900 second (15 minute).

Running VMmark Without Weathervane

While the Weathervane workload is a required part of a compliant VMmark run, there can be situations in which it can be useful to perform a VMmark run without Weathervane. These situations might include testing the VMmark harness or testing a workload other than Weathervane. Running without Weathervane can save time, because the Weathervane restore can take more than 20 minutes.

Though in other workloads the restore can be skipped, this is not possible with Weathervane due to the application design. Instead, disable Weathervane in the `VMmark3.properties` file by editing the `WorkloadList` parameter to remove all the Weathervane workloads (`AuctionLB`, `AuctionWebA`, `AuctionWebB`, `AuctionAppA`, `AuctionAppB`, `AuctionDB`, `AuctionMSQ`, `AuctionNoSQL`, `ElasticLB`, `ElasticWebA`, `ElasticWebB`, `ElasticAppA`, `ElasticAppB`, and `ElasticDB`).

The VMmark3-ConfigChecker Script

The VMmark3-ConfigChecker script (`VMmark3-ConfigChecker.pl`) is a Perl script that can be used by VMmark submitters to review several components of a submission for potential compliance and configuration issues. It is not intended to be an exhaustive check, but is rather a framework that can be built upon. The script's output also does not indicate the presence or absence of errors or noncompliant items in the benchmark configuration. The script simply highlights areas to evaluate for such issues.

NOTE The VMmark3-ConfigChecker script supports only submissions from runs executed entirely on hosts running ESXi 6.0 or greater.

Running the VMmark3-ConfigChecker Script

The VMmark3-ConfigChecker script is supplied as part of the VMmark package (in the `~/VMmark3/tools` directory).

To run the VMmark3-ConfigChecker script, follow these steps:

1. On the prime client change directories to the results folder of the run to be reviewed (typically `~/VMmark3/results/`*`Run-name`*`/`).
2. Execute the script with the `-v` (verbose) parameter. For example:
   ```
   perl ~/VMmark3/tools/VMmark3-ConfigChecker.pl -v
   ```
3. Review the output of the script. The detailed results of each check are written to the `vmmarkconfigchecker-results` subdirectory.

NOTE The script decompresses reporter files found in the submission under review. We advise users to delete the decompressed output from the results folder being reviewed by the VMmark3-ConfigChecker script before submitting.

Additional points:

- For help and command line options, run:
  ```
  perl ~/VMmark3/tools/VMmark3-ConfigChecker.pl -h
  ```
- You are encouraged to extend the mechanisms utilized in the script to add additional functionality. If you need assistance extending the script, please contact VMware at `vmmark-info@vmware.com`.

Manually Running the VMmark Reporter Scripts

VMmark results for publication must include detailed information about the ESXi and vCenter Server instances in which the test was run. This information is usually obtained automatically by reporter scripts. However, if the scripts do not complete correctly, you can run them manually.

Run only the reporter script that is needed to replace the one that did not complete correctly. Before running a VMmark reporter script manually, ensure that the testbed configuration is identical to the VMmark benchmark submission.

The following two sections describe running the host reporter script (`VMmark-ReporterMaster.sh`) and vCenter server reporter script (`VMmark-Reporter-VC.sh`), respectively.

Manually Running the Host Reporter Script

From the prime client, run the `VMmark-ReporterMaster.sh` script once for each host for which the reporter bundle is needed. The script takes several minutes to complete; you can run the script in multiple terminal windows on the prime client at the same time if you need to run it for more than one host.

For each ESXi host in the systems under test cluster, run this command and replace *`<hostname>`* with the host's hostname or IP address:

```
sh /root/VMmark3/tools/VMmark-ReporterMaster.sh <hostname> /root/VMmark3/results
              /root/VMmark3/results /root/VMmark3/VMmark3.properties sut
              >/root/VMmark3/results/Reporter-<hostname>-manual.out 2>&1
```

For each ESXi host in the client cluster, run this command and replace *`<hostname>`* with the hostname or the host's IP address:

```
sh /root/VMmark3/tools/VMmark-ReporterMaster.sh <hostname> /root/VMmark3/results
              /root/VMmark3/results /root/VMmark3/VMmark3.properties client
              >/root/VMmark3/results/Reporter-<hostname>-manual.out 2>&1
```

The `VMmark-ReporterMaster.sh` script will take a few minutes to complete and the new `.tgz` bundle, `Reporter-`*`<hostname>`*`-manual.out` and `ReporterResult-`*`<hostname>`*`.txt` will be saved into the `/root/VMmark3/results` directory.

You can display the directions to run the reporter manually by running the following on the prime client:

```
sh /root/VMmark3/tools/VMmark-ReporterMaster.sh
```

Manually Running the vCenter Server Reporter Script

From the prime client, run the `VMmark-Reporter-VC.sh` script:

```
sh /root/VMmark3/tools/VMmark-Reporter-VC.sh "/root/VMmark3/VMmark3.properties"
"/root/VMmark3/results/VC-GenerateLogs.txt" "/root/VMmark3/results/"
"/root/VMmark3/results" >/root/VMmark3/results/ReporterVC-manual.out 2>&1
```

The `VMmark-Reporter-VC.sh` script will take a few minutes to complete and the new `.tgz` bundle, `ReporterVC-manual.out` and `VC-GenerateLogs.txt` will be saved into the `/root/VMmark3/results` directory.

You can find directions for running the VC reporter by running

```
sh /root/VMmark3/tools/VMmark-Reporter-VC.sh
```

Don't run the VC reporter manually more than once in a 30 minute period. Running it more frequently than this can cause the VC Reporter `.tgz` file to get quite large (multiple GB in size).

How to Enable and Analyze esxtop Performance Data

To help diagnose problems or simply to obtain the best performance from a virtualized platform it can be useful to get and analyze esxtop performance data from your VMmark run. This section describes how to do so.

Obtain a Tool to View the esxtop Performance Data

There are various ways to view esxtop performance data. In this section we'll describe using the NMONVisualizer tool, though you can also use perfmon or other tools.

NMONVisualizer can be downloaded from here:
https://nmonvisualizer.github.io/nmonvisualizer/

The tool can be run on MacOS, Windows, and Linux. If desired, it can be run on the Prime Client.

Enable esxtop Performance Data Collection in VMmark

Follow these steps to enable esxtop performance data collection in VMmark:

1. Set the `EsxtopCollection` parameter to `true` in the VMmark3.properties file:
   `EsxtopCollection = true`
2. Specify a LUN on which the esxtop file should be saved (it must be shared storage for the systems under test):
   `EsxtopLUN = /vmfs/volumes/`*`LUN-NAME`*
   (where *`LUN-NAME`* is your datastore name).

Capture the esxtop Performance Data

Follow these steps to capture the esxtop performance data:

1. Run VMmark.
2. Transfer the VMmark results folder to wherever you put the NMONVisualizer `.jar` file.
3. The VMmark results folder will include a file like `HOSTNAME-Esxtop.csv.tgz`. Unzip this `.tgz` file. It will create a `vmfs` folder.

   NOTE When unzipped, these files can be very large (200MB or more).

Open the esxtop Performance Data for Analysis

Follow these steps to open the esxtop performance data for analysis:

1. Open the NMONVisualizer program `.jar`.
2. In NMONVisualizer, select **File** > **Load**.
3. Browse to the `vmfs` folder that was created in the VMmark results folder when you unzipped the `.tgz` file. In that folder, locate a `.csv` file. It will be something like:
   `vmfs/volumes/LUN-NAME/esxtop/hostname-Esxtop.csv`
4. Click **Parse**.

   NOTE For large data sets, this operation can be slow.

Example esxtop Performance Data Analysis: CPU Utilization

To view CPU utilization levels from the VMmark run, in the NMONVisualizer left sidebar select your hostname > Physical Cpu > Total > % Core Util Time. This will show the core utilization time during the run.

% Core Util Time is the metric typically used to represent CPU utilization, though there are several other metrics available that measure utilization slightly differently.

The bottom pane in NMONVisualizer shows the average, minimum, maximum, and so on over the interval. Note that this interval includes the benchmark ramp-up, not just the steady state. If you want to see just the utilization over the steady state, the start and end time of the steady state period can be found in `Score_N_Tile_Test.txt` and `VMmark3–graphs.html`.

In addition to CPU utilization, all the other performance metrics are in the same file and can be analyzed in a similar manner.

Security Mitigations C

This appendix provides information about required and recommended mitigations for various vulnerabilities in systems used to perform VMmark tests, as well as required updates to the disclosure report. It consists of the following sections:

- "Required Mitigations for VMmark System Under Test" on page 103.
- "Recommended Mitigations for the VMmark Environment" on page 106.
- "Verify that all Required Mitigations are Enabled" on page 107.
- "Required Disclosure Report Updates for Security Mitigations" on page 108.

For VMmark 3.1, all microcode, hypervisor, and guest mitigations specified in this appendix must have been applied to the system under test in order for a VMmark result to be compliant.

Required Mitigations for VMmark System Under Test

In order to be compliant, a VMmark result must be obtained on systems with the following security vulnerabilities mitigated:

NOTE This list is current as of the publication date of this User's Guide, but could change in the future. Check for the latest version of this User's Guide for any updates.

- CVE-2017-5715 (Spectre Branch Target Injection)
- CVE-2017-5753 (Spectre Bounds Check Bypass)
- CVE-2017-5754 (Meltdown Rogue Data Cache Load)
- CVE-2018-3615 (Foreshadow L1 Terminal Fault - SGX)
- CVE-2018-3620 (Foreshadow-NG L1 Terminal Fault - OS)
- CVE-2018-3639 (Spectre-NG Speculative Store Bypass)
- CVE-2018-3640 (Spectre-NG Rogue System Register Read)
- CVE-2018-3646 (Foreshadow-NG L1 Terminal Fault - VMM)

The subsections below address these mitigations at the microcode, ESXi, and guest OS level. An additional subsection addresses CVE-2018-3615, which is a special case.

Processor Microcode Mitigations

The processors in the servers that are part of the SUT need to be running microcode that includes mitigations for the following vulnerabilities:

- CVE-2017-5715 (Spectre Branch Target Injection)
- CVE-2018-3640 (Spectre-NG Rogue System Register Read)

In many cases the latest firmware updates available from the server vendor will include these mitigations; this is the recommended way to apply the mitigations. In other cases, running an appropriate version of ESXi will load microcode patches that include these mitigations. Regardless of how the mitigations are applied, however, it is the responsibility of the benchmarker to confirm that the mitigations are enabled and to update the disclosure report accordingly.

VMware ESXi Mitigations

The version(s) of ESXi running on the SUT needs to include mitigations for the following vulnerabilities:

- CVE-2017-5715 (Spectre Branch Target Injection)
- CVE-2017-5753 (Spectre Bounds Check Bypass)
- CVE-2017-5754 (Meltdown Rogue Data Cache Load)
- CVE-2018-3639 (Spectre-NG Speculative Store Bypass)
- CVE-2018-3646 (Foreshadow-NG L1 Terminal Fault - VMM)

The first four of these mitigations are automatically included with the appropriate ESXi version or patch. The fifth mitigation (for CVE-2018-3646) requires additional actions, which are described below.

It is the responsibility of the benchmarker to confirm that these mitigations are enabled and to update the disclosure report accordingly.

NOTE The instructions in this Appendix are for Intel Skylake and newer processors. For other processors, contact VMware at `vmmark-info@vmware.com` for assistance.

Mitigating CVE-2018-3646

To mitigate CVE-2018-3646:

1. Download and install the patches appropriate for your ESXi version as described in VMware Security Advisory VMSA-2018-0020:
   https://www.vmware.com/security/advisories/VMSA-2018-0020.html
2. Set `VMkernel.Boot.hyperthreadingMitigation = true` as described VMware KB article 55806:
   https://kb.vmware.com/s/article/55806
3. Reboot the ESXi system.

Guest OS Mitigations

The guest operating system(s) running in the VMmark workload virtual machines need to include mitigations for the following vulnerabilities:

- CVE-2017-5715 (Spectre Branch Target Injection)
- CVE-2017-5753 (Spectre Bounds Check Bypass)
- CVE-2017-5754 (Meltdown Rogue Data Cache Load)
- CVE-2018-3620 (Foreshadow-NG L1 Terminal Fault - OS)
- CVE-2018-3639 (Spectre-NG Speculative Store Bypass)

The workload virtual machines created from the VMmark 3.1 template will already include these guest operating system mitigations. Nevertheless, it is the responsibility of the benchmarker to confirm that the mitigations are enabled and to update the disclosure report accordingly.

CVE-2018-3615 Mitigation

CVE-2018-3615 (Foreshadow L1 Terminal Fault - SGX) is a vulnerability of the SGX feature in some Intel processors. At the time of this publication, VMware ESXi neither uses the Intel SGX feature nor supports its virtualization. For this reason, neither ESXi systems nor the virtual machines running on them are susceptible to exploits of this vulnerability.

Because this satisfies the CVE-2018-3615 mitigation requirement for the VMmark SUT, the relevant cells in the VMmark disclosure report template are pre-filled with "N/A" (not applicable), and can be left that way.

For the latest information on this subject, see VMware KB article 54913 (https://kb.vmware.com/s/article/54913).

Recommended Mitigations for the VMmark Environment

The vCenter server, the hardware and ESXi version used to run the client virtual machines, and the client virtual machines themselves are not defined as part of the VMmark SUT by the VMmark Run and Reporting Rules. It is therefore ***not required*** that these non-SUT portions of the VMmark environment include mitigations for the above-referenced vulnerabilities.

However, VMware ***strongly recommends*** that the mitigations listed in "Required Mitigations for VMmark System Under Test" on page 103 also be applied to these non-SUT portions of the environment, and that additionally the vCenter server used for the VMmark runs include all relevant mitigations, as described in VMware Security Advisory VMSA-2018-0020:
https://www.vmware.com/security/advisories/VMSA-2018-0020.html

Verify that all Required Mitigations are Enabled

After applying all required security mitigations, verify that they are all enabled by running the spectre-meltdown-checker script and the HTAwareMitigation module, as described below.

Run the spectre-meltdown-checker.sh Script

The `spectre-meltdown-checker.sh` script is included in the `/root/` directory of the VMmark template, and is thus in every workload and client virtual machine.

On each ESXi host, from one of the workload virtual machines:

1 In the guest VM, run `spectre-meltdown-checker.sh`:

```
sh /root/spectre-meltdown-checker.sh
```

2 Inspect the report and verify that the summary at the bottom is all green and indicates “OK” for all required mitigations. For example:

SUMMARY: CVE-2017-5753:OK CVE-2017-5715:OK CVE-2017-5754:OK CVE-2018-3640:OK CVE-2018-3639:OK CVE-2018-3615:OK CVE-2018-3620:OK CVE-2018-3646:OK

A result of “KO” indicates that a mitigation is not active.

Run the HTAwareMitigation Module

The HTAwareMitigation module can be downloaded from VMware.

From a system on which PowerCLI is installed:

1 Follow the instructions in VMware KB article 56931 (https://kb.vmware.com/s/article/56931) to obtain and install the HTAwareMitigation module.

2 Use the HTAwareMitigation module to scan the VMmark systems under test cluster:

```
Get-HTAwareMitigationConfig -ClusterName VMmarkSUT_Cluster_Name
```

3 Inspect the HTAwareMitigation `output.html` report and verify that **RuntimeHTAMSetting** is **TRUE** for all ESXi hosts in the cluster.

Required Disclosure Report Updates for Security Mitigations

Security mitigations affect two parts of the disclosure report, both addressed below.

Include Non-Default Settings in Relevant Disclosure Report Sections

Any changes made to the default settings or configuration of the system under test must be described in the relevant section of the disclosure report. Such changes include firmware updates, ESXi updates, guest operating system updates, and so on.

Even if a change is implied in the Security Mitigations section, it must *also* be included in the relevant section for the component being changed.

Update Security Mitigations Section of Disclosure Report

New for VMmark 3.1, the disclosure report contains a Security Mitigations section (see Table C-1 for a sample). After running the `spectre-meltdown-checker.sh` script and HTAwareMitigation module described in "Verify that all Required Mitigations are Enabled" on page 107, you should update the Security Mitigations section to reflect the results, changing every occurrence of "###" to either "Yes" or "No," as appropriate.

NOTE The row for CVE-2018-3615 contains "N/A" in all three mitigation columns (Server Firmware, ESXi, and Guest OS) and can be left that way. See "CVE-2018-3615 Mitigation" on page 105 for more information.

Table C-1. Sample Security Mitigations Section of Disclosure Report

Security Mitigations						
				Mitigated		
Vulnerability	**CVE**	**Exploit Name**	**Public Vulnerability Name**	**Server Firmware**	**ESXi**	**Guest OS**
Spectre	2017-5753	Variant 1	Bounds Check Bypass	N/A	### [Yes/No]	### [Yes/No]
Spectre	2017-5715	Variant 2	Branch Target Injection	### [Yes/No]	### [Yes/No]	### [Yes/No]
Meltdown	2017-5754	Variant 3	Rogue Data Cache Load	N/A	### [Yes/No]	### [Yes/No]
Spectre-NG	2018-3640	Variant 3a	Rogue System Register Read	### [Yes/No]	N/A	N/A
Spectre-NG	2018-3639	Variant 4	Speculative Store Bypass	N/A	### [Yes/No]	### [Yes/No]
Foreshadow	2018-3615	Variant 5	L1 Terminal Fault - SGX	N/A	N/A	N/A
Foreshadow-NG	2018-3620	Variant 5	L1 Terminal Fault - OS	N/A	N/A	### [Yes/No]
Foreshadow-NG	2018-3646	Variant 5	L1 Terminal Fault - VMM	N/A	### [Yes/No]	N/A