VMware vSphere 6 Fault Tolerance

Architecture and Performance

TECHNICAL WHITE PAPER

Table of Contents

Introduction

VMware vSphere® Fault Tolerance (FT) provides continuous availability to virtual machines, eliminating downtime and disruption—even in the event of a complete host failure. This white paper gives a brief description of the vSphere FT architecture and discusses the performance implication of this feature with data from a wide variety of workloads.

vSphere FT Architecture

vSphere FT enables a virtual machine to survive a physical server failure by creating an exact replica virtual machine on another host that can take over at the time of failure. During failover, the transition of a vSphere FT virtual machine from one physical server to another is similar to a migration using vSphere vMotion®: it is completely seamless. That means there is zero downtime, zero data loss, zero connection loss, continuous service availability, and complete transaction integrity.

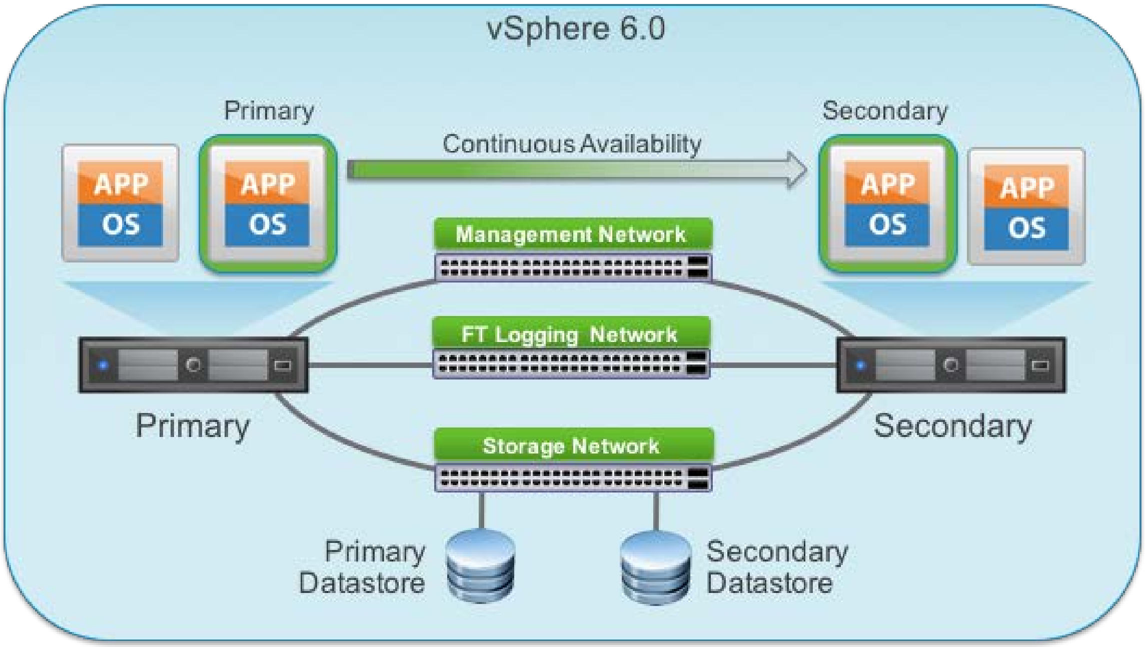


Figure 1. Fast Checkpointing keeps the secondary virtual machine in sync with the primary virtual machine.

vSphere FT works by continuously replicating an entire running virtual machine from one physical server to another. The result is that an FT-protected virtual machine has two replicas: the primary virtual machine and the secondary virtual machine, each running on distinct ESXi hosts. These replicas are logically identical —they represent a single virtual machine state and a single network identity, but they are physically distinct. Each replica has its own set of virtual machine files (including VMX and VMDK files), which vSphere FT automatically keeps in sync. When a physical server fails, one of the replicas will resume execution, and the virtual machine state, the network identity, and all active network connections for the virtual machine will be identical, ensuring a seamless failover process.

vSphere FT is made possible by four underlying technologies: storage, runtime state, network, and transparent failover.

Storage

vSphere FT ensures the storage of the primary and secondary virtual machines is always kept in sync. When vSphere FT protection is enabled, an initial synchronization of the VMDKs happens using VMware vSphere Storage vMotion® to ensure the primary and secondary have the exact same disk state.

This initial Storage vMotion happens whenever FT is turned on while the virtual machine is running, whenever FT protection is re-established after a failover occurs, or whenever a powered-off FT virtual machine powers on. The FT virtual machine is not considered FT-protected until the Storage vMotion completes.

After this initial synchronization, vSphere FT will mirror VMDK write operations between the primary and secondary over the FT network to ensure the storage of the replicas continues to be identical.

Runtime State

vSphere FT ensures the runtime state of the two replicas is always identical. It does this by continuously capturing the active memory and precise execution state of the virtual machine, and rapidly transferring them over a high-speed network, allowing the virtual machine to instantaneously switch from running on the primary ESXi host to the secondary ESXi host whenever a failure occurs.

Network

The networks being used by the virtual machine are also virtualized by the underlying ESXi host, ensuring that even after a failover, the virtual machine identity and network connections are preserved. Similar to vSphere vMotion, vSphere FT manages the virtual MAC address as part of the process. If the secondary virtual machine is activated, the secondary host sends a gratuitous ARP so the network switch is aware of the new physical location of the virtual MAC address. Since vSphere FT preserves the storage, the precise execution state, the network identity, and the active network connections, the result is zero downtime and no disruption to users should an ESXi host failure occur.

Transparent Failover

vSphere FT ensures that the primary always agrees with the secondary about the state of the virtual machine. This is achieved by holding externally visible output from the virtual machine, and only releasing it when an acknowledgement is made from the secondary affirming that the state of the two virtual machines is consistent (for the purposes of vSphere FT, externally visible output is network transmissions). The cost for maintaining this zero data loss consistency is that network transmissions from the virtual machine are delayed until the two virtual machines are in a consistent state.

Best Practices

Based on performance testing described in the next section, a wide range of virtual machine workloads work well under FT protection. Among these are CPU-bound workloads, I/O-bound workloads, servers, and complex database workloads. One of the most common performance observations of virtual machines under FT protection is a variable increase in the network latency of the virtual machine (increasing both latency and jitter). But as the performance testing in the next section demonstrates, good application performance is typically possible under FT protection. Workloads that are particularly latency sensitive, such as high frequency trading (HFT) or real-time systems may experience more overhead.

See the *Performance Best Practices for VMware vSphere 6.0* [2] guide for recommendations regarding vSphere FT setup.

Additionally, this paper applies the recommendations found in the following KB articles to improve performance.

- KB 2039495
  Large packet loss at the guest OS level on the VMXNET3 vNIC in ESXi

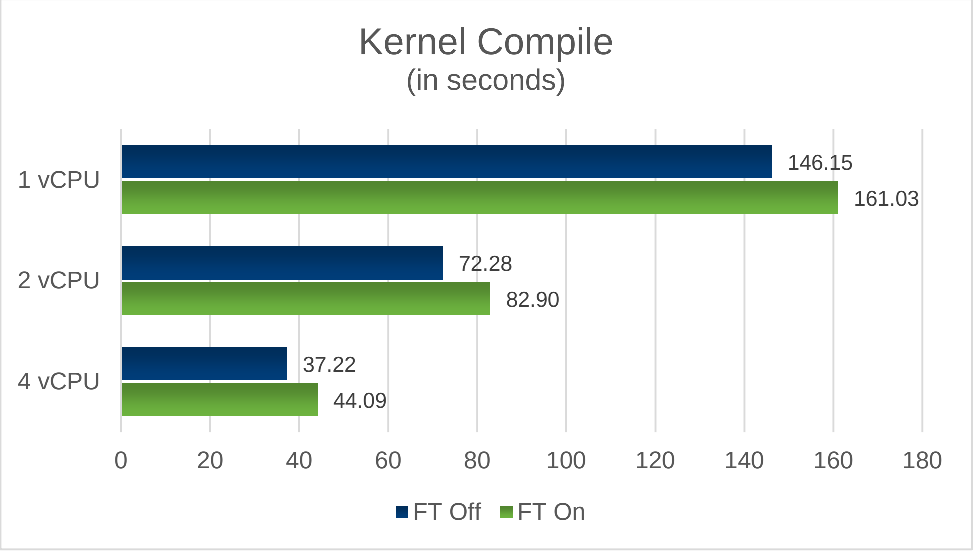
- KB 1010071
  The output of esxtop shows dropped receive packets at the virtual switch
- KB 2111976
  After you enable Fault Tolerance on a Windows VM (Windows XP and later) configured with Virtual Hardware version 11 and vSphere 6.0 VM Tools, the virtual machine's workload throughput will drop significantly

Fault Tolerance Performance

This section discusses the performance characteristics of FT virtual machines using a variety of micro-benchmarks and real-life workloads. Micro-benchmarks were used to stress CPU, disk, and network subsystems individually by driving them to saturation. Real-life workloads, on the other hand, were chosen to represent what most customers would run, and these workloads have been configured to have a CPU utilization of 60 percent in steady state. Identical hardware testbeds were used for all the experiments, and the performance comparison was done by running the same workload on the same virtual machine with and without FT enabled. The hardware and experimental setup details are provided in Appendix A: Benchmark Setup and Appendix B: Workload Details. For each experiment (where applicable), the traffic on the FT logging NIC during the steady state portion of the workload is also provided as a reference.

Kernel Compile

This experiment shows the time taken to do a parallel compile of the Linux kernel. This is a both a CPU- and MMU-intensive workload due to the forking of many parallel processes. During this benchmark the CPU was 100 percent utilized. This workload did some disk reads and writes, but generated no network traffic. As seen in Figure 2, FT protection increases the kernel compile time a small amount—about 7 seconds.



FT Traffic:
1 vCPU: 2.1 Gbps
2 vCPU: 2.6 Gbps
4 vCPU: 2.4 Gbps

Figure 2. Kernel compilation performance (lower is better)

Netperf

Netperf is a micro-benchmark that measures the performance of sending and receiving network packets. Netperf was configured with several scenarios and network speeds to demonstrate the throughput and latency performance of TCP/IP under FT. The results were achieved while applying the recommendations provided in the following KB articles: KB 2039495, KB 1010071, and KB 2111976.

Throughput

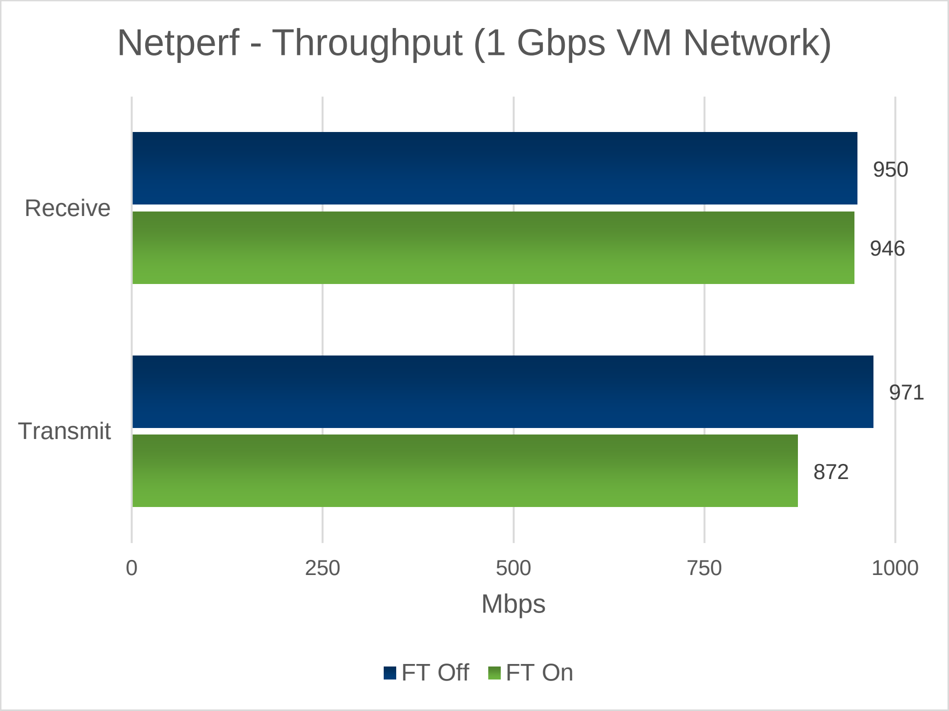
This netperf experiment measures unidirectional TCP/IP throughput. One experiment was done in each direction, when the virtual machine was either receiving or transmitting data. The speed of the virtual machine network is an important factor for performance, and one experiment was done on a 1 Gbps uplink (Figure 3) and another was done for a 10 Gbps uplink (Figure 4).

The throughput experiments reveal some important points about network performance when vSphere FT is enabled:

- Workload traffic under FT protection can meet or exceed 1 Gbps for both 1 Gbps and 10 Gbps uplinks.
- The FT network traffic is minimal for workloads where the virtual machine is transmitting a lot of data, but not receiving much data.
- The FT network traffic is high for workloads where the virtual machine is receiving a lot of data, but not transmitting much data.

Receive-heavy workloads (where the virtual machine is receiving large amounts of data) tend to increase FT traffic due to the requirement to keep the replicas in sync. The influx of data into the primary causes large differences between the replicas, thus requires more data to be sent over the FT network. Transmit-heavy workloads (where the virtual machine is transmitting a lot of data), on the other hand, cause very few differences between the replicas, thus very little FT traffic. Therefore, transmit-heavy applications, such as Web servers or read-bound databases, tend to have fewer FT traffic requirements.

As shown in Figure 4, the receive workload on a 10 Gbps virtual machine network is challenged when FT protected, achieving less than 3 Gbps.

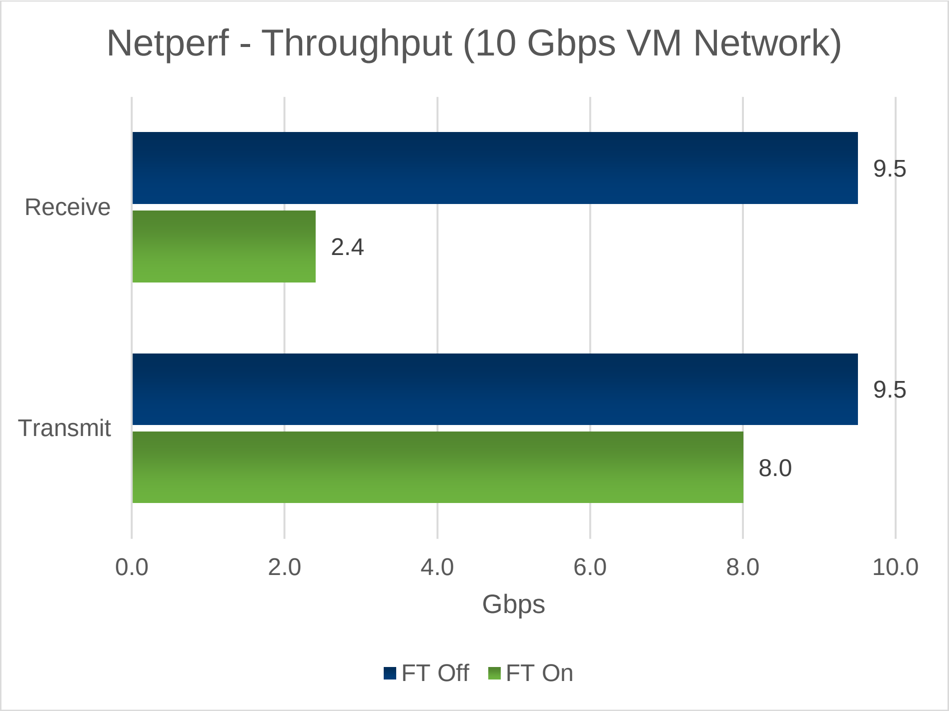


FT traffic:

Receive run:
3.1 Gbps

Transmit run:
331 Mbps

Figure 3. Netperf throughput for receive and transmit on a 1 Gbps VM network (higher is better)



FT traffic:

Receive run:
5.9 Gbps

Transmit run:
529 Mbps

Figure 4. Netperf throughput on a 10 Gbps VM network for receive and transmit (higher is better)

Latency

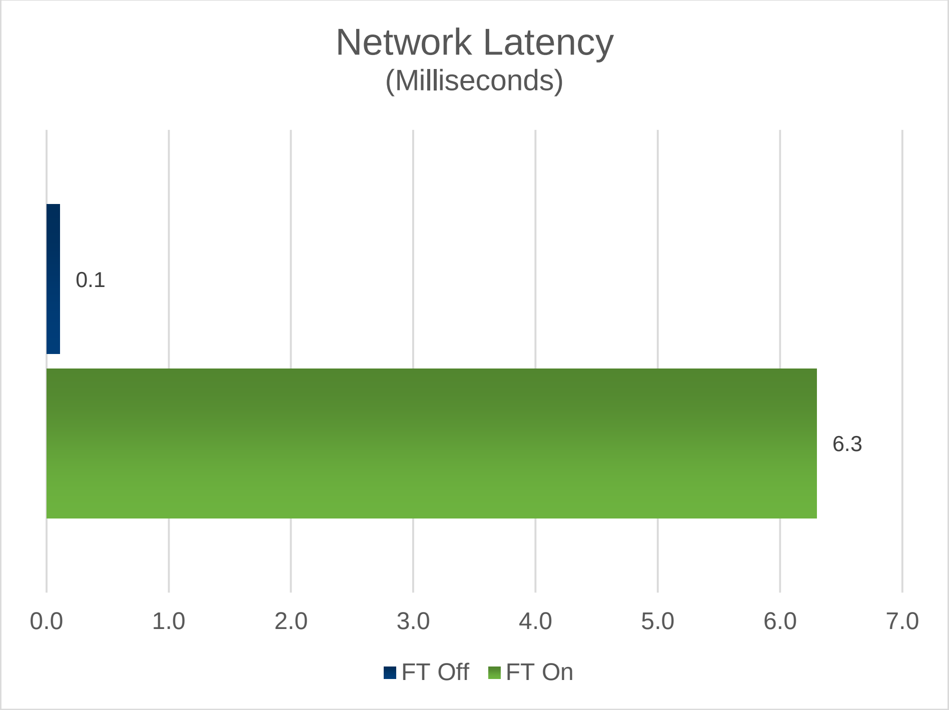
This netperf experiment measures TCP/IP latency under FT. Fault tolerance introduces some delay to network output (measureable in milliseconds, as seen in Figure 5). The latency occurs because an FT-protected primary withholds network transmissions until the secondary acknowledges to the primary that it has reached an identical runtime state.

In this experiment, netperf was run with the TCP_RR configuration (single stream, no parallel streams) and the round-trip latency is reported here (it is the inverse of the round-trip transaction rate). TCP_RR is a pure latency benchmark: the sender transmits a 1-byte message and blocks waiting for a 1-byte response, the benchmark counts the number of serial transactions completed in a unit time, and there is no parallelization.

In a pure latency benchmark, all latency increases drop throughput (for example, latency increases 57 times and throughput drops identically). In real-world applications, however, workload performance is generally robust against network latency increases.

This is because normal server applications are not pure latency benchmarks. They handle multiple connections at a time, and each connection will transmit several packets worth of data before pausing to hear a response. The result is that real-world applications tolerate network latencies without dropping throughput. The netperf throughput experiment is an example of this, and the client/server workloads examined in this paper demonstrate the same thing. In all these experiments, while low-level network latency may increase, it typically results in slight visible effects on perceived UI interactivity, latency for database queries, and overall application throughput.

One aspect not measured by netperf is jitter and latency fluctuation. FT-protected virtual machines can vary widely in latency depending on the workload, and over time within a given workload. This can cause significant jitter. Highly latency-sensitive applications, such as high frequency trading (HFT), or some voice-over-IP (VoIP) applications may experience high overhead with FT. However, some voice applications, where the bulk data is carried by separate processes where only call management traffic is FT-protected, would perform well.

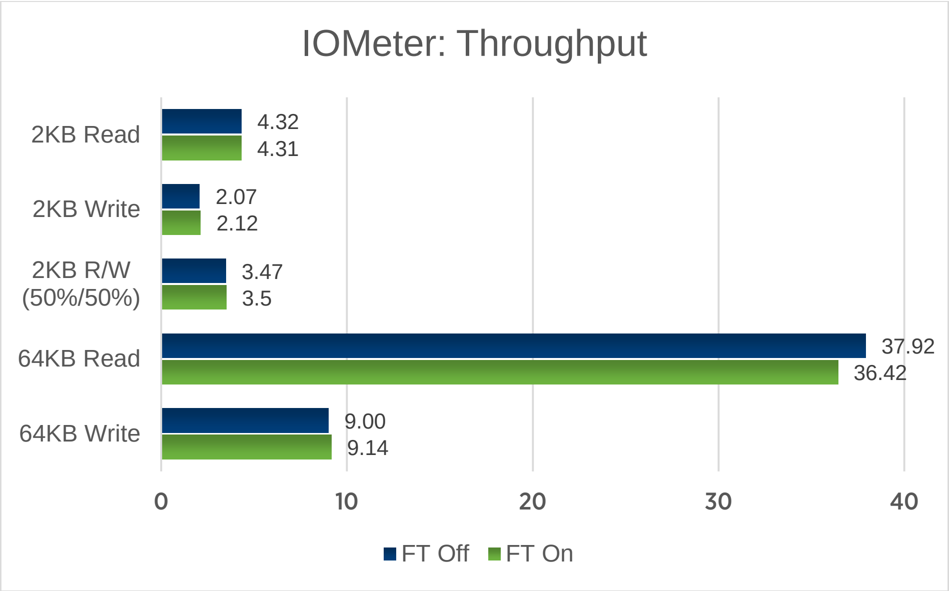


FT traffic:
273.9 Mbps

Figure 5. Network latency for VMs with FT off and on (lower is better)

IOMeter

IOMeter is an I/O subsystem measurement and characterization tool for Microsoft Windows. It is designed to produce a mix of operations to stress the disk. This benchmark ran random I/Os of various types. Figure 6 shows the FT-protected virtual machine achieves nearly as much throughput as the non-protected virtual machine.



FT Traffic:

2KB reads:
167 Mbps

2KB writes:
62 Mbps

2KB reads/writes:
79 Mbps

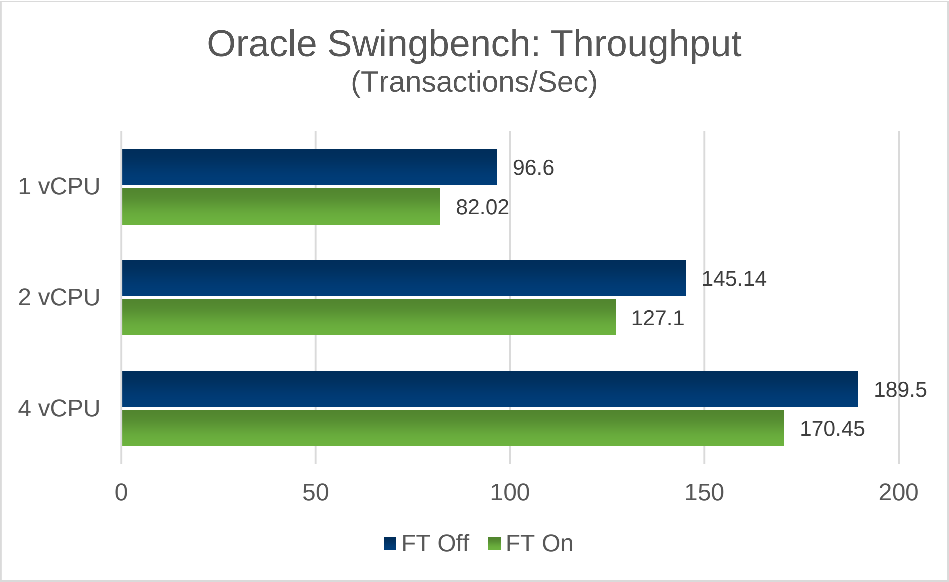
64KB reads:
208 Mbps

64KB writes:
45 Mbps

Figure 6. IOMeter throughput results for 2KB and 64KB message sizes (higher is better)

Swingbench with Oracle 11g

In this experiment, an Oracle 11g database was driven using the Swingbench 2.2 [3] order entry online transaction processing (OLTP) workload. This workload has a mixture of CPU, memory, disk, and network resource requirements. The FT-protected virtual machine is able to achieve nearly as much throughput as the non-FT virtual machine (Figure 7). The latency of basic operations has increased under FT protection (Figure 8).



FT Traffic:

1 vCPU: 0.9 Gbps
2 vCPU: 1.3 Gbps
4 vCPU: 1.6 Gbps

Figure 7. Swingbench OLTP workload run against an Oracle 11g database VM with vs. without FT (higher is better)

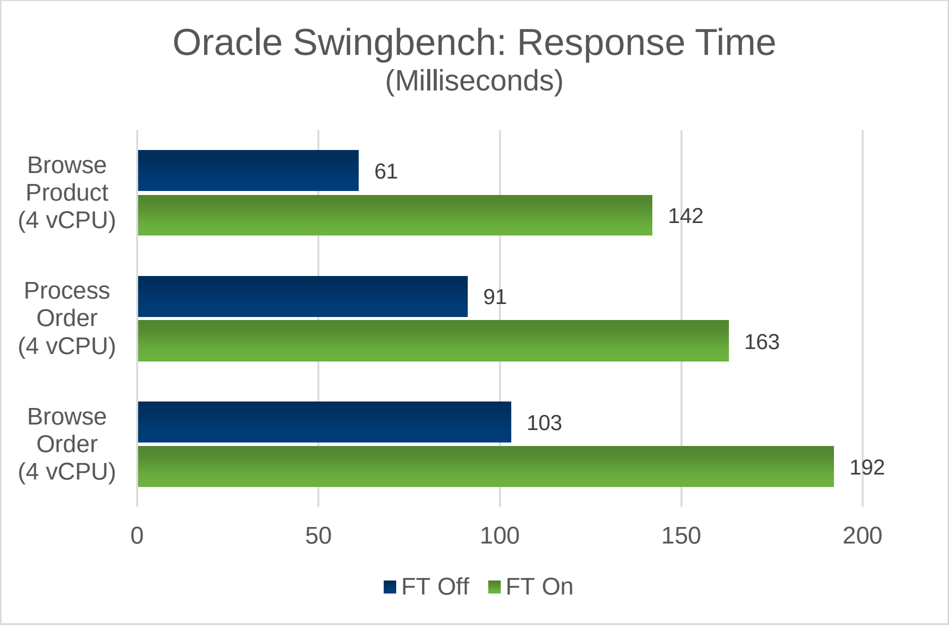


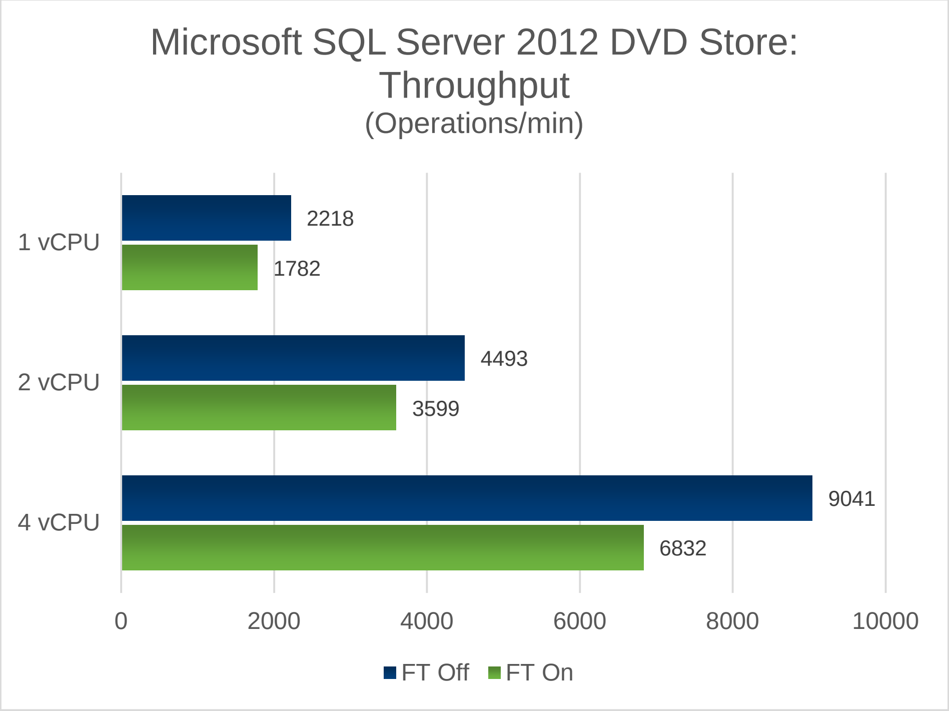
Figure 8. Benchmark response time for three common actions run against the Oracle 11g database VM with vs. without FT (lower is better)

DVD Store with Microsoft SQL Server 2012

In this experiment, the DVD Store 2 [3] benchmark was used to drive the Microsoft SQL Server® 2012 database. This benchmark simulates online transaction processing of a DVD store. 64 simultaneous user sessions were used to drive the workload. As with the Swingbench benchmark, this workload has a mixture of CPU, memory, disk, and networking resource requirements.

FT protection incurs throughput overhead for this workload (Figure 9), and latency for basic operations has increased (Figure 10).

Disclaimer: The throughput here is not meant to indicate the absolute performance of Microsoft SQL Server® 2012, nor to compare its performance to another DBMS. SQL Server was used to place a DBMS workload on ESXi, and observe the performance of ESXi with and without FT.



FT Traffic:

1 vCPU: 0.8 Gbps
2 vCPU: 1.1 Gbps
4 vCPU: 1.6 Gbps

Figure 9. DVD Store throughput for basic operations of a Microsoft Server 2012 VM with vs. without FT enabled (higher is better)

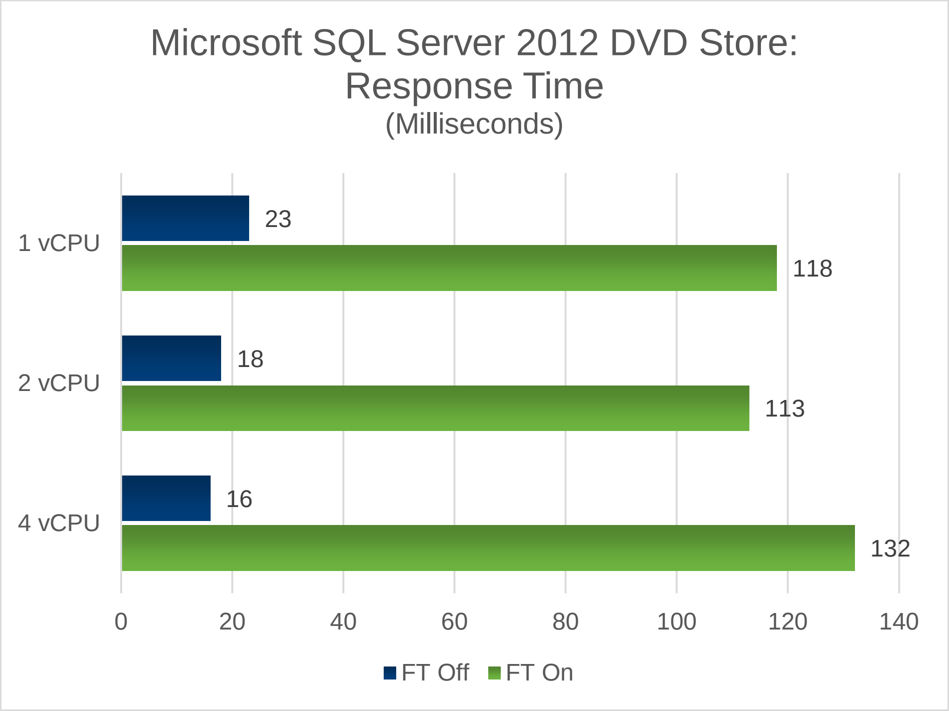
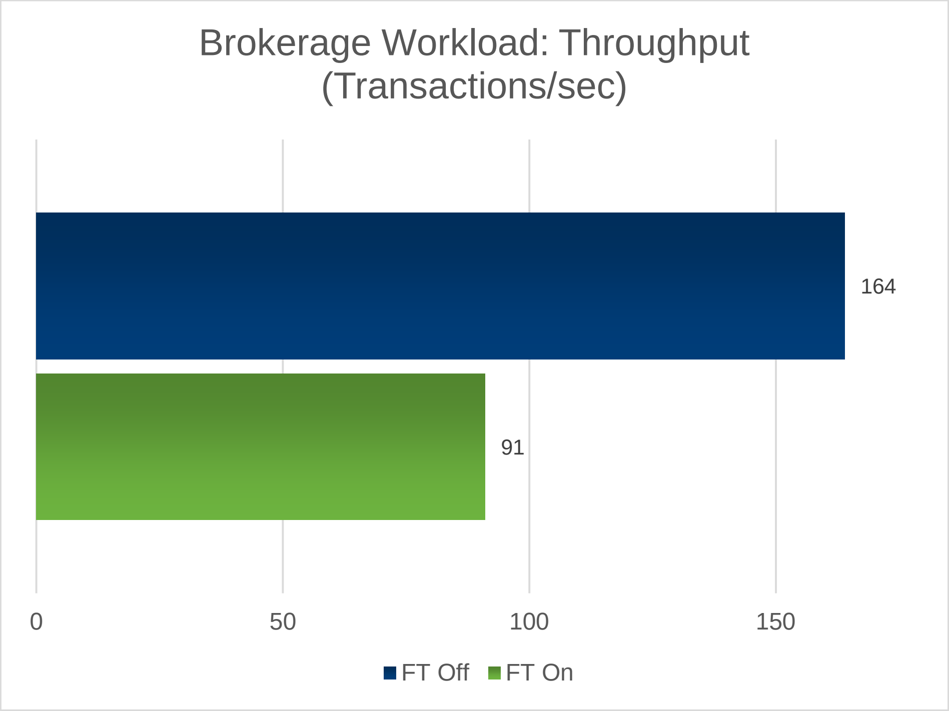


Figure 10. DVD Store response time for basic transactions against a Microsoft SQL Server 2012 VM with vs. without FT enabled (lower is better)

Brokerage Workload Derived from TPC-E

This experiment uses a workload modeled on the TPC-E [5] benchmark, which simulates the online transaction processing workload (OLTP) of a brokerage firm in which customers trade stock, query accounts, and research markets. This workload has a mixture of CPU, memory, disk, and networking resource requirements. The workload is particularly stressful and has a large working set.

Disclaimer: The brokerage workload is a non-comparable implementation of the TPC-E business model.



FT traffic: 4.4 Gbps

Figure 11. Throughput of the brokerage workload with FT off and on (higher is better)

As shown in Figure 11, there is a reduction in user transaction throughput when FT is enabled. This is due to the very intensive nature of this brokerage workload.

vCenter Server

This experiment shows the difference in performance of VMware vCenter Server™ when using a vCenter Server appliance protected versus unprotected by Fault Tolerance. The baseline setup consisted of 32 hosts and 1000 virtual machines managed by a vCenter Server appliance that ran version 6.0 with an embedded vPostgres database and was configured with 4 vCPUs and 16GB of RAM. Similar results were observed when running vCenter Server on Windows with an external SQL Server database.

Workloads

Three workloads were tested to highlight the impact of VMware Fault Tolerance on vCenter Server:

1. Management operations. In these workloads, a number of vCenter Server management operations were issued to the hosts and virtual machines; for example, powering on virtual machines, cloning them, or reconfiguring them. This test generated a load of many such operations per minute and measured the resulting throughput with FT. Two load profiles were measured. Typical customer setups have operational rates that are less than both profiles, so both loads represented higher load scenarios. “Load A” generated tens of operations per minute, while “Load B” generated hundreds of operations per minute.
2. Browsing with the vSphere Web Client. The latency of browsing operations (for example, clicking on summary pages or virtual machine list pages) was measured when FT was enabled.
3. Restart. The latency to restart the vCenter Server process was measured with FT enabled. This determined the timeframe it takes for vCenter Server to recover from a software crash.

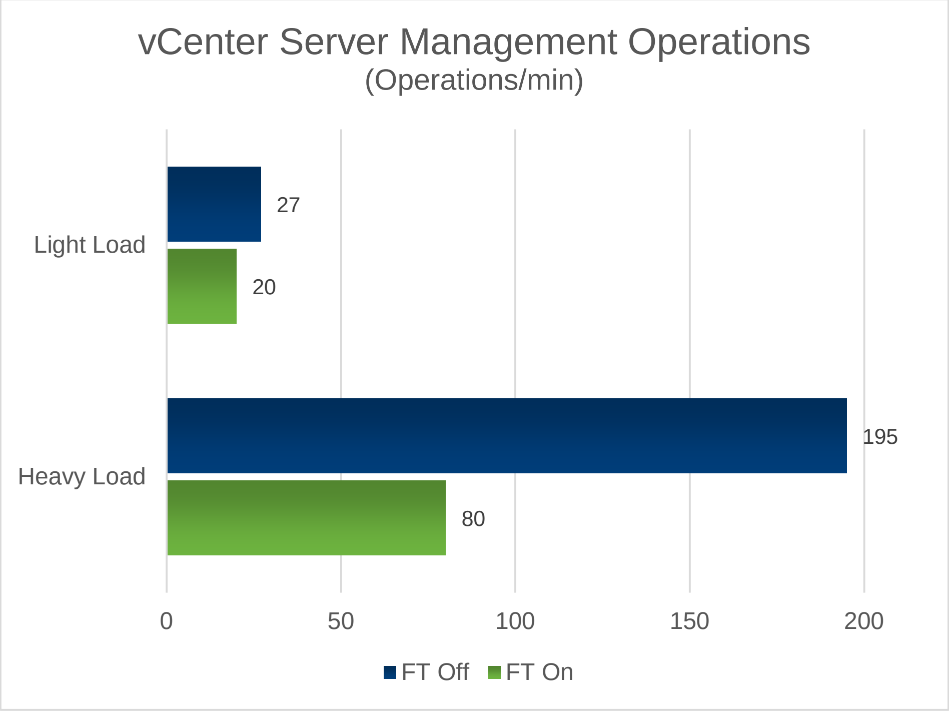
These workloads have a mixture of memory churn, disk I/O, and network I/O. The management operations induce network traffic between vCenter Server and the vSphere hosts, disk traffic to the embedded database, and memory churn within the vCenter Server Appliance for tracking the creation, deletion, and migration of various objects like virtual machines. The Web browsing workload generates network bandwidth from the vCenter Server Appliance to the Web browser and induces memory churn within the appliance to handle session management, object caching, and database query formulation. Finally, restarting vCenter Server causes network traffic to retrieve data from vSphere hosts, disk I/O to persist data to the database, and memory churn in order to fill internal caches in vCenter Server.

Results

The results are shown in Figure 12 (vCenter Server Management Operations), Figure 13 (Web Client Browsing), and Figure 14 (Restarting vCenter Server).

Management Operations

When running a light load (approximately 27 operations per minute), a vCenter Server using FT can support about 74% of the throughput of vCenter without FT. Similarly, a vCenter server under heavy load (nearly 200 operations per minute), is able to support about 40% of the throughput of vCenter without FT, as shown in Figure 12.



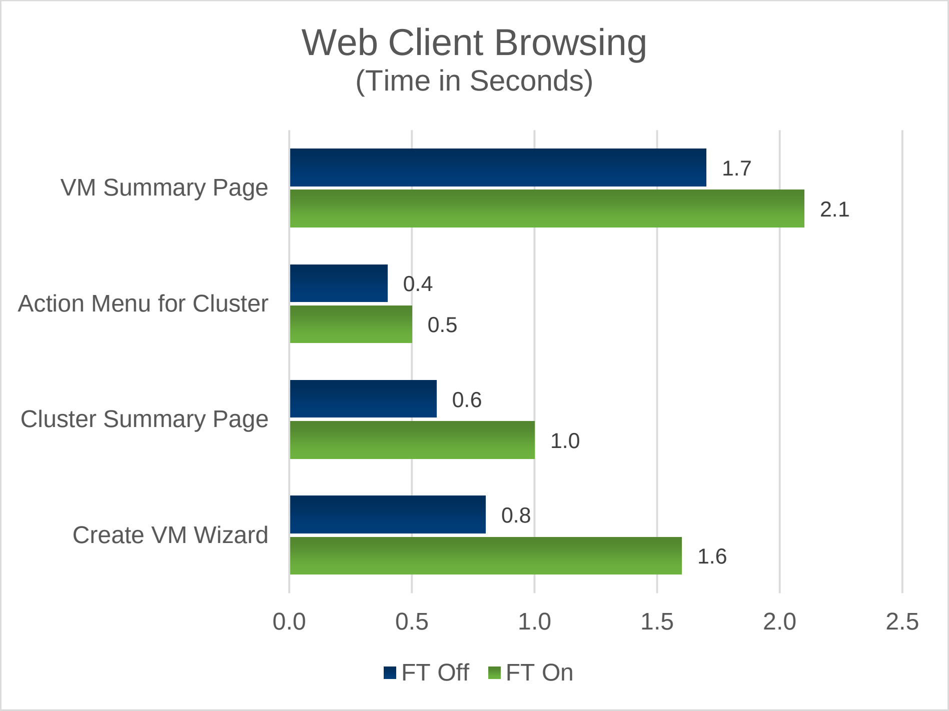
FT Traffic:

Light: 2.7 Gbps
Heavy: 4.2 Gbps

Figure 12. Operations per minute achieved by vCenter Server Load A (higher is better)

Browsing with the vSphere Web Client

The latency for various browsing operations increases under FT from an average of 1.1s without FT to 1.5s with FT. The effect this has on UI browsing inside the Web Client, while noticeable, is such that the Web Client remains comfortably usable.



1.67

Figure 13. The time it takes in seconds to perform common browsing activities with the vSphere Web Client (lower is better)

Restarting vCenter Server

Finally, the restart time of vCenter Server with FT is approximately 1.3 times longer than without FT. The restart time includes the amount of time it takes for the vCenter Server to come back online with the vSphere Web Client available.

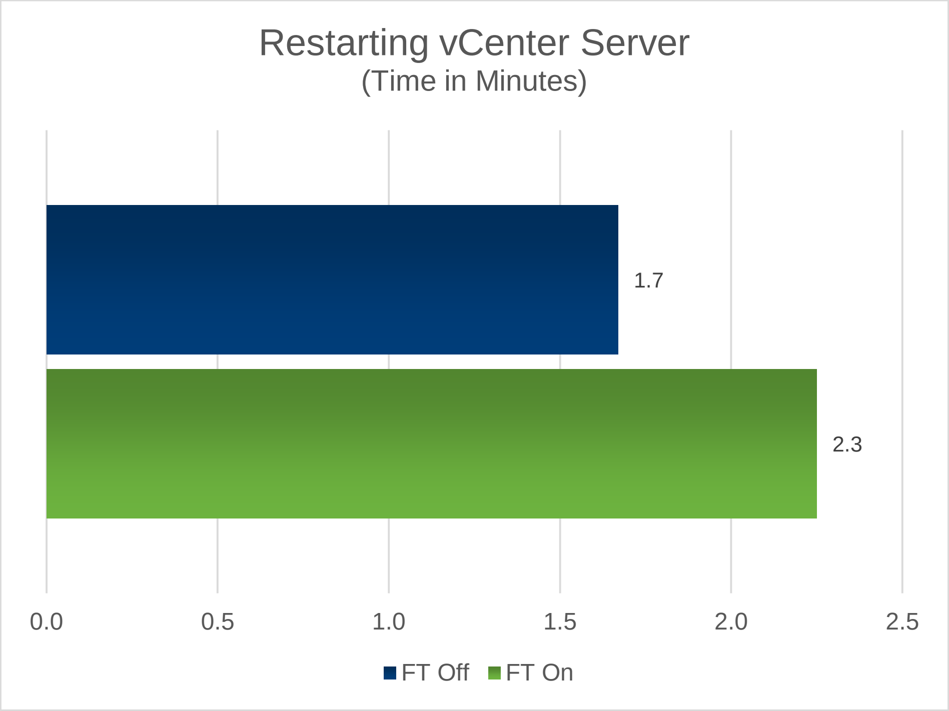


Figure 14. Restarting vCenter Server to simulate failover due to application crash (lower is better)

For more information about protecting vCenter Server with FT, see the *VMware vCenter Server Availability Guide* on page 8 [6].

Fault Tolerance Performance Summary

Based on performance testing described in the previous section, a wide range of virtual machine workloads work well under FT protection. Among these are CPU-bound workloads, I/O-bound workloads, servers, and complex database workloads. One of the most common performance observations of virtual machines under FT protection is a variable increase in the network latency of the virtual machine (increasing both latency and jitter). But as the performance testing in the previous section demonstrates, good application performance is typically possible under FT protection; network latency does not dictate overall application throughput for a wide variety of applications. On the other hand, applications that are highly sensitive to network latency (such as high frequency trading or real-time workloads) have a higher performance cost under FT protection.

VMware vSphere Fault Tolerance is a revolutionary new technology. It universally applies fault-tolerant technology to any multi-vCPU workload in a uniquely simple-to-use way. The vSphere FT solution is able to achieve good throughput for a wide variety of applications.

Appendix A: Benchmark Setup

Storage Array

- System: EMC VNX 5300
- FLARE OS: 05.31.000.5.720
- LUNs: RAID 5 LUNs (6 disks), RAID 0 LUNS (6 disks)

Primary and Secondary Hosts

- System: HP ProLiant DL380 Gen9
- CPU: Intel® Xeon® Processor E5-2667 v3 @ 3.20GHz
- Number of cores/socket: 8; Number of sockets: 2; Hyper-Threading: enabled
- Memory: 128 GB

Client Machine

- System: Dell PowerEdge R720
- CPU: Intel® Xeon® Processor E5-2650 @ 2.00GHz
- Number of cores/socket: 8; Number of sockets: 2; Hyper-Threading: enabled
- Memory: 64 GB
- OS: Windows Server 2012 R2 Datacenter Edition, 64-bit

Appendix B: Workload Details

Kernel Compile

- Virtual machine configuration: 1 vCPU / 2 vCPUs / 4 vCPUs, 2 GB RAM, LSI Logic virtual SCSI adapter
- OS version: RHEL6
- Kernel version: 2.16.32-71.el16
- Benchmark details: Time taken to compile (make bzImage) Linux kernel 2.6.20 was measured. The experiment was repeated 5 times and the average run time was reported.

Netperf

- Virtual machine configuration: 4 vCPUs, 8 GB RAM, VMXNET3 virtual NIC, LSI Logic Virtual SCSI adapter
- OS version for both client/server: Windows Server 2012 R2 Datacenter Edition, 64-bit
- Netperf configuration: 8 clients started in parallel, remote and local message size: 64K, remote and local socket size: 4MB
- The recommended configuration found in KB articles KB 2039495, KB 1010071, and KB 2111976 were applied

Iometer

- Virtual machine configuration: 4 vCPUs, 8 GB RAM, VMXNET3 virtual NIC, LSI Logic Virtual SCSI adapter
- OS version: Windows Server 2012 R2 Datacenter Edition, 64-bit

- Benchmark parameters:
  - 2k 100% read, 100% write, 50-50% read-write
  - 64k 100% read, 100% write

Swingbench with Oracle 11g

- Virtual machine configuration: 1 vCPU / 2 vCPUs / 4 vCPUs, 8 GB RAM, VMXNET3 virtual NIC, LSI Logic virtual SCSI adapter
- OS version: RHEL5
- Kernel version: 2.16.18-164.el15
- Oracle Version: 11.1.0.6.0
- Database Details: max number of processes: 150, SGA buffer size: 1535MB, data file size: 23 GB (index, redo, and database files on the same location)
- Swingbench configuration:
  - Swingbench version: 2.2, Calling Circle Database
  - Number of orders: 23,550,492
  - Number of customers: 864,967
  - Runtime: 30 mins
  - ODBC driver: ojdbc6.jar
  - Driver type: Thin
  - Number of users: 160
  - Pooled: 1
  - LogonDelay: 0
  - Transaction MinDelay: 50
  - Transaction MaxDelay: 250
  - QueryTimeout: 60
  - Workload Weightage: NewCustomerProcess – 20, BrowseProducts – 50, ProcessOrders – 10, BrowseAndUpdateOrders – 50

Note: Database was restored from backup before every run.

DVD Store with Microsoft SQL Server 2012

- Virtual machine configuration: 1 vCPU / 2 vCPUs / 4 vCPUs, 8 GB RAM, VMXNET3 virtual NIC, LSI Logic virtual SCSI adapter
- OS version: Windows Server 2012 R2 Datacenter Edition, 64 bit
- Microsoft SQL Server version: SQL server 2012
- Database size: 100 GB
- Database row count: 200 million customers, 10 million orders per month, 1 million products
- DVD Store benchmark version: 2007/12/03
- Benchmark parameters
  - n_threads: 64
  - ramp_rate: 2
  - run_time: 30 mins
  - warmup_time: 4 mins
  - think_time: 0.40 secs

  - pct_newcsutomers: 40
  - n_searches: 5
  - search_batch_size: 8
  - n_line_items: 10
  - db_size_str: L

Note: The database was restored from backup after every run.

Brokerage Workload Derived from TPC-E

- Virtual machine configuration: 4 vCPUs, 42 GB RAM, VMXNET3 virtual NIC, PVSCSI virtual SCSI adapter
- OS version: Windows Server 2012, Datacenter Edition, 64-bit
- Microsoft SQL Server version: SQL Server 2012
- Database scale: 115,000 customers
- Number of user connections: 80
- SQL Server buffer max memory size: 40 GB

vCenter Server Workload

- Virtual infrastructure configuration:
  - VMware vCenter Server 6.0 run in a VM with 4 vCPU and 16 GB memory
  - 32 ESXi hosts each containing 125 VMs (4,000 VMs total)
- Operations performed: See Table 1, below.

Operation	Description
Add Port Group	Create a new port group on a virtual switch.
Remove Port Group	Remove a port group from a virtual switch.
Clone VM	Create a clone of a virtual machine.[1]
Create Folder	Create a new folder in the vCenter inventory hierarchy.
Delete Folder	Delete a folder from the vCenter inventory hierarchy.
Create Snapshot	Create a snapshot of a virtual machine.
Delete Snapshot	Delete a snapshot of a virtual machine.
Group Power-On VMs	Power on several virtual machines in a single operation.
vMotion VM	Move a powered-on virtual machine to a different host.
Power On VM	Power on a single virtual machine in a DRS cluster.
Power Off VM	Power off a single virtual machine.
Reconfigure VM	Edit a virtual machine's configuration settings.[2]
Register VM	Add a .vmx file from a datastore to the vCenter inventory.
Unregister VM	Remove a virtual machine from the vCenter inventory without deleting its files from its datastore.

[1] The performance of a clone operation depends on the disk size of the VM being cloned.

[2] This benchmark's Reconfigure operation uses VM Memory Shares as a representative configuration setting.

Relocate VM	Move a powered-off virtual machine to a different host and datastore.
Remove VM	Delete a virtual machine from the vCenter inventory, and delete its files from its datastore.
Reset VM	Reset a single virtual machine.
Suspend VM	Suspend a single virtual machine.
Resume VM	Resume a single virtual machine.

Table 1. Operations performed in vCenter Server workload

For more information about the vCenter Server workload, see “vCenter Server 6.0 Performance and Best Practices [6].”

References

[1] VMware, Inc. (2015, June) Performance Best Practices for VMware vSphere 6.0. [Online]. http://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf

[2] Dominic Giles. Swingbench. [Online]. http://dominicgiles.com/swingbench.html

[3] Todd Muirhead and Dave Jaffe. (2007, December) linux.dell.com. [Online]. http://linux.dell.com/dvdstore/2007-12-03/

[4] Transaction Processing Performance Council. (2015) TPC-E. [Online]. http://www.tpc.org/tpce/

[5] Mohan Potheri, G. Blair Fritz, and Puneet Gupta. (2015, May) VMware vCenter Server 6.0 Availability Guide. [Online]. http://www.vmware.com/files/pdf/techpaper/VMware-vCenter-Server6-Availability-Guide.pdf

[6] Mike Stunes and Ravi Soundararajan. (2015, August) vCenter Server 6.0 Performance and Best Practices. [Online]. https://communities.vmware.com/docs/DOC-29203

[7] VMware, Inc. (2015) vSphere Availability. [Online]. http://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-availability-guide.pdf

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Date: 4-Jan-16 Comments on this document: https://communities.vmware.com/docs/DOC-30054