

WHITE PAPER

Protecting Mission-Critical Workloads with VMware Fault Tolerance



Table of Contents

Fault Tolerance and Virtualization 3
Fault Tolerance in the Physical World 3
VMware Fault Tolerance 4
vLockstep Technology 5
Transparent Failover 5
Component Failures 6
VMware HA and FT Compared 7

Protecting Mission-Critical Workloads with VMware® Fault Tolerance

Fault Tolerance and Virtualization

As server virtualization becomes commonplace in the enterprise, IT organizations are growing increasingly reliant on the associated virtualization benefits: higher server consolidation ratios, better resource utilization, lower power consumption, greater workload mobility via technologies such as VMotion, and the general ease of system management.

Three key trends in the IT industry are driving consolidation ratios even higher in the coming decade:

- Desire to improve overall resource utilization and power consumption.
- Emergence of multicore technologies in computer architecture.
- Innovative solutions such as VMware Distributed Resource Scheduler (DRS) and VMware Distributed Power Management (VMware DPM). DRS and VMware DPM complement the hardware evolution trends by applying dynamic resource allocation to lower the capital and operating costs in a data-center.

However, improved resource utilization and higher consolidation ratios also introduce new availability requirements to virtualized environments. As more business-critical workloads are deployed in virtual machines, a catastrophic failure of a single physical server might lead to an interruption of a large number of services. VMware solutions address many of these requirements and bring additional unique advantages in the business continuity area. These advantages mainly derive from disassociating the virtual machine state including all business logic from the underlying hardware and applying data protection, disaster recovery, and high availability services to virtual machines in a hardware-independent fashion.

Examples of such solutions and services include VMware Consolidated Backup (VCB), VMware High Availability (HA), and VMware Site Recovery Manager (SRM). VCB provides a centralized backup solution for virtual machines. VMware HA delivers clustering, monitoring, and restart services that are operating system and application agnostic. SRM provides disaster recovery services. Virtualization-aware high availability products and services will continue to gain wider acceptance in the future. According to the IDC predictions for 2009, "High availability

of application and data will become a key driver for the next phase of virtualization software adoption."¹

For virtual machines that can tolerate brief interruptions of service and data loss for in-progress transactions, existing solutions such as VMware HA supply adequate protection. However, for the most business-critical and mission-critical workloads even a brief interruption of service or loss of state is unacceptable. In the world of physical servers such workloads traditionally enjoyed special treatment as outlined in the next section.

Fault Tolerance in the Physical World

All fault tolerance solutions rely on redundancy. For example, many early fault tolerant systems were based on redundant hardware, hardware failure detection, and failing over from compromised to properly operating hardware components. More broadly, traditional high availability solutions fall into two categories: fault tolerant servers based on proprietary hardware and software clustering.

Fault Tolerant Servers

Fault tolerant servers have been in use for several decades but are generally adopted for only the most mission-critical applications. Such servers generally rely on proprietary hardware, leading to much higher initial cost. Some fault tolerant servers also employ proprietary operating systems, limiting or eliminating the use of unmodified commodity applications. Fault tolerant servers traditionally come with 24x7 support contracts and require highly trained IT staff. Fault tolerant servers provide CPU and component redundancy within a single enclosure, but they cannot protect against larger-scale outages such as campuswide power failures, campuswide connectivity issues, and loss of network or storage connectivity. In addition, although failover is seamless, re-establishing fault tolerance after an incident might be a lengthy process potentially involving on-site vendor visits and purchasing custom replacement components. For physical systems, fault tolerant servers provide the highest SLAs at the highest cost.

Software Clustering

Software clustering generally requires a standby server with a configuration identical to that of the active server. The standby must have a second copy of all system and application software, potentially doubling licensing costs. A failure causes a short interruption of service that disrupts ongoing transactions while control is transferred to the standby. Application software must be made aware of clustering to limit the interruption

1. IDC, "Worldwide System Infrastructure Software 2009." Top 10 Predictions. December 2008, IDC #215390, Volume 1.

of service. However, the potential for data loss or corruption during a crash is not fully eliminated. IT staff requires special training to build and operate such complex applications, and this requirement limits the number of deployable services. Configuring the clustering software can also be quite complex and can require special quorum devices. An example of such a system is an application built around Microsoft Cluster Service (MSCS).

Both solutions may be applied to virtualized environments. A single fault tolerant physical server could accommodate a number of virtual machines (if, for example, an x86-based fault-tolerant server vendor supports ESX running on custom hardware). Similarly, clustering software could be installed inside virtual machines (for example, MSCS is supported in guest operating systems running on ESX). Although they enable the virtualization of mission-critical applications, both strategies merely bring to virtualization the same solutions that exist in the physical world with all the associated downsides: the high initial cost and high operating expenses of dealing with custom hardware in the case of fault tolerant servers or the configuration complexity, software development complexity, licensing overhead, and ongoing operating expenses of software clustering.

VMware Fault Tolerance

VMware Fault Tolerance (FT) leverages the well known encapsulation properties of virtualization by building high availability directly into the x86 hypervisor in order to deliver hardware style fault tolerance to virtual machines. It requires neither custom hardware nor custom software. Guest operating systems and applications do not require modifications or reconfiguration. In fact, they remain unaware of the protection transparently delivered by the ESX hypervisor at the x86 architecture level.

FT relies on VMware vLockstep technology to establish and maintain an active secondary virtual machine that runs in virtual lockstep with the primary virtual machine. The secondary virtual machine resides on a different host and executes exactly the same sequence of virtual (guest) instructions as the primary virtual machine. The secondary observes the same inputs as the primary and is ready to take over at any time without any data loss or interruption of service should the primary fail. Both virtual machines are managed as a single unit but run on different physical hosts (and even in different buildings, if you choose to configure your installation in this way). Because the solution is built directly into the virtualization stack, you can engage virtual machine protection with just a few clicks, minimizing deployment, configuration, licensing, and operating costs. Guest operating systems and applications remain unmodified.

Key Feature of VMware Fault Tolerance

FT provides the following key features:

- Runs on standard x86 based servers, vendor neutral.
- Supports standard unmodified guest operating systems and applications.
- Protects dozens of guest operating systems already supported by ESX, including 32- and 64-bit Windows, Linux, Solaris, and many other legacy guests.
- x86 hypervisor-based solution; integration with virtual machine technology, operating system neutral.
- Support for all emerging applications frameworks that have not yet evolved their own clustering solutions.
- Support for existing virtual machines.
- Single image management: virtual machine is installed and managed in the usual way as a single image; no need for additional operating system and software licenses.
- vLockstep guarantees: the primary and secondary execute exactly the same x86 instruction sequences.
- Transparent failover with no data or state loss in the virtual machine; all state, including storage, memory, and networking is preserved even in the face of catastrophic hardware failures.
- Potential different physical locations for primary and secondary to guard against campuswide or buildingwide failures.
- Automatic re-establishment of fault tolerance after hardware failures.
- Integration with HA and DRS that are responsible for selecting a new secondary host after a failure; no manual steps during failover or after recovery.
- Failing systems can be returned to the HA cluster after repairs without any additional FT reconfiguration.
- Component failover when combined with network teaming and storage multipathing.
- No additional installation; FT is a built-in feature of VMware ESX.
- Mixing FT and non-FT virtual machines in the same environment for higher utilization.

FT delivers continuous availability in the presence of even the most severe failures such as unexpected host shutdowns and loss of network or power in the entire rack of servers. It preserves ongoing transactions without any state loss by providing architectural guarantees for CPU, memory, and I/O activity. The two key technologies behind FT are vLockstep and Transparent Failover.

vLockstep Technology

vLockstep technology was developed to deliver architectural guarantees that the states of the primary and secondary virtual machines are identical at any point in the execution of instructions running in the virtual machine. vLockstep accomplishes this by having the primary and the secondary execute identical sequences of x86 instructions. The primary captures all nondeterminism from within the processor as well as from virtual I/O devices.

Examples of nondeterminism include events received from virtual network interface cards, network packets destined for the primary virtual machine, user inputs, and timer events. The captured nondeterminism is sent across a logging network to the secondary. The secondary virtual machine uses the logs received over the logging network to replay the nondeterminism in a manner identical to the actions of the primary. The secondary thus executes the same series of instructions as the primary. See Figure 1: vLockstep architecture.

Because both the primary and secondary virtual machines execute the same instruction sequence, both initiate I/O operations. The main difference between them is the treatment of the outputs. The output of the primary always takes effect: disk writes are committed to disk and network packets are transmitted, for example. All output of the secondary is suppressed by the hypervisor. The external world cannot detect the existence of the secondary and, at all times, treats a fault tolerant virtual machine as single unit executing the workload.

vLockstep technology provides full system guarantees: at each guest instruction boundary the primary and the secondary are identical, including all guest operating system and guest application state as well as all virtual hardware state. Because the secondary needs to see only nondeterministic inputs, the logging network can use conventional 1Gbps NICs and switches. Because both the primary and secondary are active

and execute the same instruction stream at similar speeds, the overall performance impact is minimal.

vLockstep technology requires physical processor extensions and was developed in collaboration with Intel and AMD. All currently shipping x86 server processors are vLockstep capable. vLockstep technology is fully integrated into the ESX hypervisor.

Transparent Failover

When you are using vLockstep technology, the existence of the primary and secondary virtual machines is hidden from the outside world, which observes only a single virtual machine image executing a workload. VMware Fault Tolerance must be able to detect hardware failures rapidly when they occur on the physical machine running either the primary or the secondary virtual machine and respond appropriately. The hypervisors on the two physical machines establish a system of heartbeat signals and mutual monitoring when you initiate vLockstep. From that point on, a failure of either physical machine is noticed by the other in a timely fashion. Should a failure happen on either physical machine, the other physical machine can take over and continue running the protected virtual machine seamlessly via transparent failover.

As an example, consider a failure of the physical machine running the primary virtual machine, as shown in step 1 of Figure 2 on page 6. The hypervisor on the secondary physical machine immediately notices the failure. The secondary hypervisor then disengages vLockstep. The secondary hypervisor has full information on pending I/O operations from the failed primary virtual machine, and it commits all pending I/O. It then performs a “go live” operation and becomes the new primary, as shown in step 2 of Figure 2. This terminates all previous

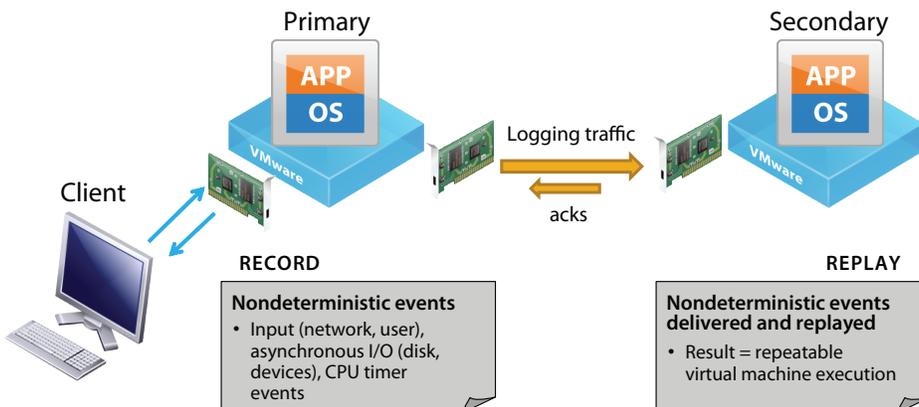


Figure 1: vLockstep architecture

dependencies on the old (and now failed) primary. For example, after going live, the new primary (the virtual machine that was initially the secondary) starts accepting network input directly from physical NICs and starts committing disk writes. There is zero state loss and no disruption of service, and the failover is automatic.

After the initial failover, VMware HA automatically selects a new host that is operating properly and has available resources, and it starts a new secondary virtual machine instance on that host, as shown in step 3 of Figure 2. The new primary hypervisor establishes vLockstep with the new secondary, thus re-enabling redundancy. From this point onward, the virtual machine is protected once more against future failures.

The entire process is transparent (zero state loss and no disruption of service) and fully automated (no IT staff intervention required). In addition, the entire process relies entirely on the capabilities of VMware HA and does not require any contact with vCenter Server, enabling continuous operation in the presence of wider-scale failures potentially affecting multiple hosts or a management network.

Most traditional fault tolerant solutions are tied to two specific hosts, which have specialized hardware or have local copies of the workload data. If one host goes down and fails to restart in a reasonable time, redundancy cannot easily be re-established. In contrast, with VMware Fault Tolerance, redundancy is automatically re-established by restarting the secondary virtual machine on any other compatible host in the local cluster.

FT deals similarly with the failure of the host executing the secondary virtual machine. The primary hypervisor notices the failure of the secondary and disengages vLockstep. The services provided by the primary virtual machine continue uninterrupted. As in the previous example, VMware HA automatically selects a new host and starts a secondary virtual machine. The primary hypervisor re-engages vLockstep, thus re-establishing

redundancy. The entire process is transparent, requires no human intervention, and does not involve any contact with vCenter Server.

VMware Fault Tolerance combines the benefits of fault tolerant servers with those of software clustering: it delivers hardware-style fault tolerance while surviving full node failures. In addition, it also automatically re-establishes fault tolerance as a part of the standard fault recovery process.

Traditionally, one of the hard technical problems to solve in providing fault tolerant solutions is avoiding “split-brain” situations. In a virtualized world, the split-brain problem can lead to two (active) copies of a virtual machine after recovery from a failure. VMware Fault Tolerance avoids split-brain under all circumstances. The transparent failover technology uses file locking on the shared storage to coordinate failover and thereby guarantees that only one side continues running as the primary virtual machine.

Component Failures

ESX already protects against component failures, such as the failure of an individual SCSI host bus adapter (HBA) or network interface card. The storage multipathing functionality in ESX automatically fails over to a new HBA if the currently active HBA stops working. Similarly, the NIC teaming functionality in ESX automatically uses another NIC if the active NIC fails. These component-level failovers happen automatically, and no FT failover is required. On the other hand, if all HBAs fail on the primary host but HBAs are still functioning on the secondary host, the VMware Fault Tolerance causes a full failover to the secondary virtual machine, so the workload can continue.

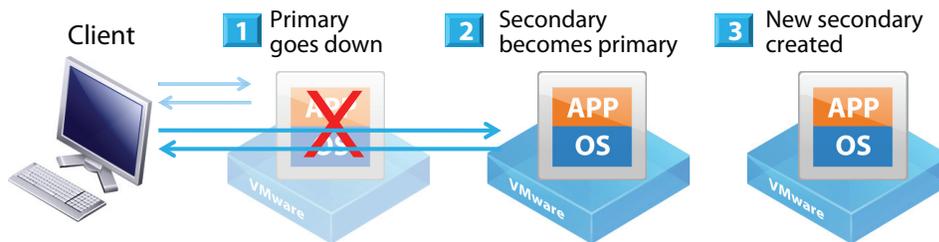


Figure 2: Transparent failover

VMware HA and FT Compared

VMware HA is a viable virtualization solution for environments that can tolerate brief interruptions of service and potential loss of transactions serviced at the time of failure. VMware HA strives to minimize downtime and deliver service continuity by restarting a virtual machine on a different host should the initial host fail.

FT targets the most mission-critical applications that cannot tolerate any interruption of service or data loss. You should also consider FT for complex multitier environments in which a sudden reboot of one server might leave you with a level of service different from what existed before the failure — for example, because of external dependencies or constraints related to the order in which services are initialized.

All VMware HA services remain available for virtual machines protected by FT. In fact, to enable FT for a virtual machine, the virtual machine must first join an HA cluster. FT also relies on

VMware HA to re-establish a new secondary automatically as a part of transparent failover. The two services are used in concert to provide the highest level of protection.

Unlike VMware HA, FT maintains an active secondary with levels of resource consumption similar to those of the primary. VMware HA does not actively use additional resources before a failure, although you might need to reserve those resources to guarantee that the virtual machine can restart successfully.

In general, to decide on the appropriate level of protection, the two factors you should consider are:

- Different levels of availability demanded by the application
- Constraints on available resources

Revision: 20090507 Item: WP-084-PRD-01-02



VMware, Inc. 3401 Hillview Ave. Palo Alto CA 94304 USA Tel 650-475-5000 Fax 650-475-5001 www.vmware.com
© 2009 VMware, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,961,941, 6,961,806, 6,944,699, 7,069,413; 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149,843, 7,155,558, 7,222,221, 7,260,815, 7,260,820, 7,269,683, 7,275,136, 7,277,998, 7,277,999, 7,278,030, 7,281,102, 7,356,679, 7,409,487, 7,412,492, 7,412,702, 7,424,710, 7,428,636, 7,433,951, and 7,434,002; patents pending. VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

