

Architecting and implementing a fault-tolerant Microsoft Windows environment.



Executive summary	2
Achieving highly-reliable, highly-available computing: two dominant approaches.....	3
Failover	3
Fault tolerance	4
The Marathon Endurance Server approach to fault tolerance	4
Architectural view.....	4
Typical Endurance Server usage scenarios.....	6

Executive summary

The goal of highly-reliable, highly-available computing is nothing new to the enterprise and scientific IT professional. Many have tried several approaches, from decentralizing to recentralizing, each time achieving certain levels of success in the face of technological and budgetary constraints, though not necessarily in that order. Nevertheless, through trial-and-error and the proof-points of time, two dominant approaches for achieving highly-reliable, highly-available computing have emerged: failover and fault tolerance.

This document compares the two approaches with the intent of providing a conceptual framework for understanding the relative merits and applications of fault tolerance achieved through the use of Marathon Endurance FTvirtual Server on HP ProLiant servers. However, the weight of this document will be to describe to the technology implementer the how-to steps for creating a fault-tolerant Microsoft® Windows® 2003 environment using HP and Marathon products.

Achieving highly-reliable, highly-available computing: two dominant approaches

Failover and fault tolerance have varied meanings, with the variances typically stemming from the environments in which the IT professionals using the term have their roots. To establish a common frame of reference, hopefully without introducing yet another meaning of either of the two terms, the following sections will describe failover and fault tolerance in simplified language. The goal is not to be all-inclusive in describing the multiple ways each is achieved, but rather to provide a basis for comparison and contrast, and to frame the applicability of the Marathon FTvirtual Server product, hereafter referred to as Endurance Server, in enterprise, scientific, and small/medium business (SMB) environments.

Failover

As the name implies, failover is the approach of moving computing tasks from one compute node over to another, in an automatic or manual fashion, should the original node fail. In the most basic of designs, one compute node is actively carrying out tasks while another is passively waiting for the first to fail. This active-passive approach is typically achieved through clustering or data replication technologies. Using data replication technologies, data from the active node are periodically transferred to the

passive node. Should the active node fail, the passive node can be started and pick up the computing tasks from the point of the most recent data transfer. Obviously, in-process transactions will be lost and – if the time between the last data replication, the failure of the active node, and the pick-up by the once passive, now active, node is more than a moment – there can be substantial data loss.

Clustering technologies address, and to varying degrees, reduce the issues of data loss. In a clustered environment, the active and the passive compute nodes access a shared data drive (or drives). Should the active node fail, the passive node can be started and pick up the compute tasks with the last data written to the shared drive. The time to move the passive node to active status can be seconds or minutes, depending on the application, but the time required is routinely less than that required in a data replication environment; therefore, fewer data are lost. In addition, clustering eliminates the data lost between the last data transfer from active node to passive node and the time required for the passive node to become the active node. Nevertheless, some compute environments cannot tolerate even a second's loss of data or lack of compute capacity, and, in such cases, fault tolerance is the dictated solution.

Fault tolerance

Simply defined, fault tolerance is the ability of a compute system to provide uninterrupted compute functions with no loss of process or data. In more complete terms, fault tolerance is the ability of two or more compute nodes to provide uninterrupted and expected services despite faults that occur *internally* or *externally* to the control of a system or application.

Faults that occur *internally* to the control of a system might be compute node component failures (e.g., memory, CPU), disk drive errors, or operating system/application crashes. Faults that occur *externally* to the control of a system might be the building-wide loss of electricity, the physical severing of network connections, or the destruction of the system or data center by fire or natural disaster. While the *external* faults are typically aggregated and defined within the context of disaster recovery, a fault-tolerant environment must comprehend and, to some degree, support the requirements of disaster recovery.

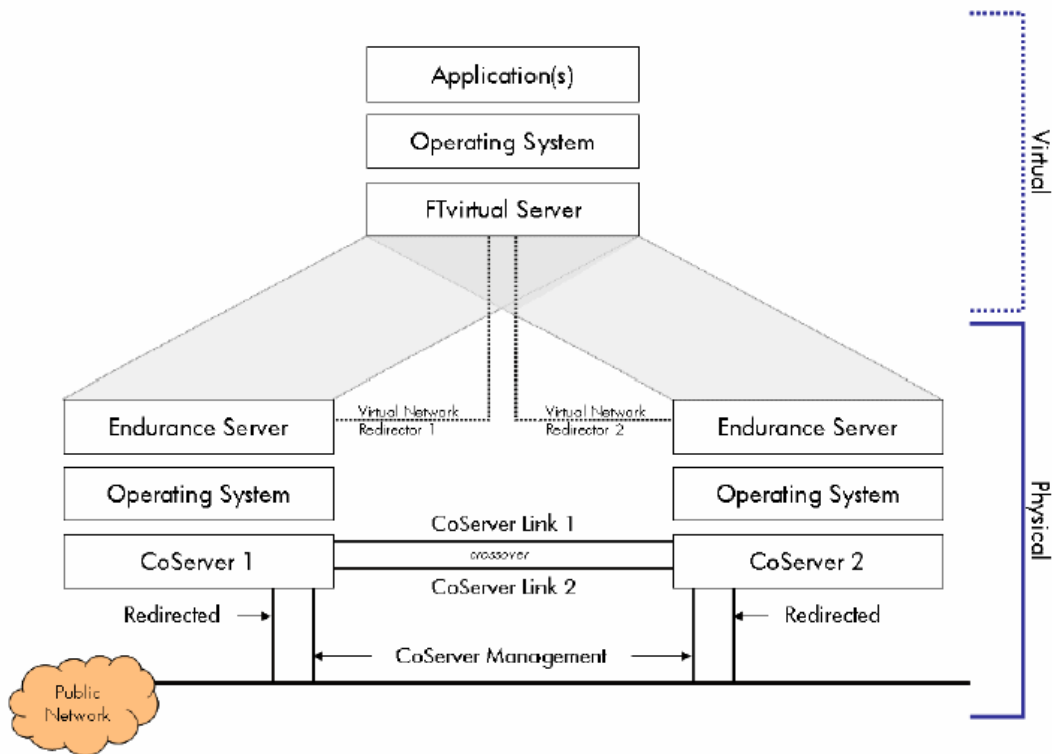
The Marathon Endurance Server approach to fault tolerance

Unlike the clustering approaches, Marathon Endurance Server eliminates the fail-over process and the subsequent time lags in application availability and data recovery by synchronizing the application(s) and data of two physical servers and two virtual servers. Consequently, should a physical server and/or a virtual server fail, the end-user will experience neither a lack of application availability or data loss.

Architectural view

Figure 1 below provides a simplified overview of the architecture of a typical Endurance Server environment. The two physical servers, called CoServer1 and CoServer2 in the Endurance Server nomenclature, operate on Microsoft Windows 2000 Server or higher and have Endurance Server installed. It is during the process of installing Endurance Server that virtual servers are created on each CoServer. The virtual servers, hosted by the CoServers' operating systems, are neither redundant nor fault tolerant until the CoServers are fully booted and the synchronization of the virtual environment (application and data) has taken place; only then do the two virtual servers become one *Fault-Tolerant* virtual Server (FTvirtual Server).

Figure 1. Simplified Architectural Overview



It is via the two dedicated and redundant Gigabit Ethernet (GbE) links (CoServer Links 1 and 2) that the CoServers and virtual servers communicate and synchronize. Of the remaining Ethernet links (10/100/1000) depicted in Figure 1, two are dedicated as CoServer Management Links for administrative access to the CoServer operating system environments; the others – the Redirected links – are dedicated for client access to the application(s) running on the FTvirtual Server.

It is important to note that although there are two virtual servers operating concurrently, it is because they are synchronized to create the FTvirtual Server that the end user, the network, the application, and the operating system perceive only one Windows-based, Intel®-based architecture server. In addition, the Windows application running on the FTvirtual Server needs no modification to operate as a fully fault-tolerant application.

Though not depicted in Figure 1, the last architectural item to note is the disposition of devices such as hard disks, CD-ROMs, NICs, etc. The FTvirtual Server has no hard disks or other devices attached directly; rather, during the Endurance Server install process, devices are identified and *redirected* for use by the FTvirtual Server.

Typical Endurance Server usage scenarios

The following usage scenarios are real-world examples of companies that have implemented Endurance Server on HP ProLiant servers. For more details of these implementations, the full stories are downloadable from the Marathon website (www.marathontechnologies.com)

Always On E-Mail. A billion-dollar hedge fund depends heavily on instant communication; a delay of even a few seconds could spell disaster. In order to ensure reliable, available communication, this global finance company created a fault-tolerant Microsoft Exchange environment using Endurance Server and HP ProLiant DL360 servers. According to the company, unplanned downtime of the Exchange environment has been all but eliminated and data protection is virtually 100%.

Real-Time Communications and Data Access. Live television and radio broadcasting can come to a screeching halt if on-air personalities lose contact with one another or their audiences. On top of that, immediate access to stored information or breaking news keeps the broadcasts lively and the audiences engaged. To ensure zero 'dead-air' time, this global sports media company implemented a fault-tolerant suite of Windows-based applications that handles e-mail, database, and broadcast scheduling using Endurance Server on HP ProLiant DL380 and DL580 servers. The net result is a fault tolerant environment that consistently exceeds all of the company's requirements for uptime and availability.

Gaming Meets Gamer Around-the-Clock. Casino gaming is a prime examples of a 24/7 industry, and this gaming franchise operates 24/7 in eight locations simultaneously. Even a moment of downtime can mean tens of thousands of dollars in lost revenues and hundreds of dissatisfied patrons. To keep the more than 1,800 geographically-dispersed slot machines turning, and the IT staff working normal daytime hours, this company implemented their slot operations, including the accounting and bonusing software, using Endurance Server on HP ProLiant DL380 servers. As a result, the slots are turning 24/7, after-hours emergency calls to IT staff have been eliminated, and the combination has directly impacted the bottom line.

For more information

http://www.cwlsystems.co.uk/data_availability.html