Real-world Implementation and Design Issues for Microsoft Windows 2000 Datacenter Server

See how Hewlett-Packard combines MS Datacenter technology with 30 years of mission critical experience to create highly available IT Infrastructures.

Ken Van Voorhees

Introduction

IT system administrators now face the challenge of making Microsoft Windows 2000 available more than 99.9% of the time. Microsoft partnered with several top-tier OEMs to deliver and support "High Availability" (HA) Microsoft Windows 2000 Datacenter Server and called it the Datacenter Program. The "Program" provides a framework for HA Windows. But what is HA and do you really need it? Here are some best practices HP engineers and consultants share with new Datacenter customers all over the world. See if Datacenter makes sense for you and see what you must do to create your own HA infrastructure.

I'll cover two keys to implementing HA Windows. First, I'll help you gain a clear understanding of what "High Availability" means. By understanding your own availability requirement, you will see which HA technologies are most relevant to you. Second, I'll provide general guidelines and detailed steps to help you design an infrastructure that blends HA technologies to meet your availability requirement.

Ultimately, High Availability is not a product you can buy. It's a goal. The result of carefully blended HA technologies, support services and human processes. These are applied using disciplined, repeatable methodology. As you will see, technologies address only small part of HA. Of greater importance are choices in systems design, outsourcing, support and process. HA is about attention to detail and the discipline to manage every aspect of your environment.

Before we talk about solutions, we need a common understanding of what HA is and how to measure it.

1. Understanding HA and the Availability Requirement.

Availability

By any practical measure, "Availability" is a ratio of the amount of time the system (or IT Infrastructure) was available over the amount of time it should have been available. Service Level Agreements (SLAs) often include availability as a performance metric. Note that "Availability" often applies to the whole infrastructure (although some would apply it only to a particular resource). Be clear on the distinction between "fault resilience" and "fault tolerance". Resilient systems are made up of clusters that achieve high availability through fail-over. Cluster nodes have independent system images and fail-over can take several minutes. Data recovery is left to the application by way of checkpoint files on shared storage. Fault tolerant systems have tighter coupling of resources and a single system image. No failure of this image is tolerated. When constituent components fail, redundancies take over without failure of the system image. Most HA computing uses resilient systems, although fault tolerant systems have a role and can help achieve 99.999% planned availability in certain HA designs.

By the Numbers

"When you measure what you are speaking about and express it in numbers, you know something about it, but when you cannot express it in numbers your knowledge about is of a meagre and unsatisfactory kind."[i]

Lord Kelvin maintained that measurement is central to understanding. This is no less true today. When discussing availability, it's easy to assume everyone is on the same page. This is not always the case. To clarify, you want a number that unambiguously describes levels of availability. By convention, this is expressed as a percentage or some number of "nines". If 100% is perfect availability, less than that describes real systems. I prefer the simple calculation[ii]

A = MTBF/(MTBF+MTTR)

"A" is availability, "MTBF" is mean time between failures and "MTTR" is mean time to repair (or recover). Three nines implies A=99.9%. Four nines A=99.99% and so on. Substitute 20 minutes for MTTR (Microsoft claims this is the average time to restore an NT-based system) and .999 for A. Solving for MTBF gives you approximately 14 days (actually13.875 days). Not coincidentally, 14 days is the duration of the Microsoft stress test for Datacenter hardware and kernel-mode drivers. The primary HA design goal is to increase A by increasing MTBF and/or decreasing MTTR.

Here is a table of availability by nines.

Number Nine, Number Nine …

Nines	% Downtime	Downtime/Yr	Downtime/Wk	Incremental HA Technology
One (98%)	2%	7.3 days	3 hr, 22 min	Tape Backups
Two (99%)	1%	3.65 days	1 hr, 41 min	RAID/Online Backups
Three (99.9%)	0.1%	8 hr, 45 min	10 min, 5 sec	Clustering/SAN Storage
Four (99.99%)	0.01%	52.5 min	1 min	Custom Support/Fibre Fabric
Five (99.999%)	0.001	5.25 min	6 sec	Fault Tolerance/Predictive Maint.

These numbers result from "unplanned downtime". In today's world of HA, techniques such as online backup and rolling upgrades for system maintenance or hardware updates keep "planned downtime" close to zero.

How much availability do you need?

Do you need three nines? Costs increase five to ten times for each additional nine.[iii] Take a critical look at your business. What are *your* costs for downtime? Here are samples from various industries. Justification starts with costs of being unavailable.

Business Category	Hourly Cost of Downtime[iv]
Brokerage	$6,500,000
Telephone Ticket Sales	$69,900
Home Shopping (TV)	$199,500
Credit Card Sales	$2,600,000
Catalog Sales	$90,000
Airline Reservation	$89,500

"Why is the system down?"

There have been many studies on downtime and its causes. The following table shows causes of downtime as evenly divided among planned outages, software and physical factors (people, hardware, environment).

Causes of Downtime[v]

Root Cause			HA or Datacenter Approach to Solve
Planned Downtime		30%	Rolling upgrades can make this nearly zero
People		15%	ITSM[vi] and MOF procedures (Defined below)
Hardware		10%	Datacenter hardware certification
Environment		5%	Datacenter availability assessment
Software			
Server Software	30%		Datacenter release management
Client Software	5%		VeriTest (User Mode) software testing
Network Software	5%	40%	WHQL 14-day stress test (Kernel Mode s/w)
Total		100%	

It's hard to look at these data and not notice the importance of people and process to achieving High Availability. In a recent white paper, Microsoft refers to industry studies that show 80% of system failures come from human errors or flawed process.[vii]

2. How to Design an "Always On" Infrastructure.

At its simplest HA means increasing the time between failures and decreasing the time to recover.

General Design Guidelines

HA starts with reliable components. Even though you configure a system as a cluster node, configure it to minimize the likelihood of failure. Consider duplex mirroring the boot partition, redundant, hot swappable power supplies, hot swappable SCSI disks, error-correcting memory, redundant fans and so forth. Each redundant component should be highly reliable on its own.

Avoid Single Points of Failure (SPF). Eliminate SPFs by redundant components and automatic fail over. When redundancy is not possible (as in human error), create processes that minimize both the chance of failure and the time to recover.

Diminish chances for human error. Minimize unstructured human contact with the system. Use scripts to automate routine tasks. Use system management tools to identify trends, conduct root cause analysis and trigger automatic responses to error conditions and events. Record the entire production infrastructure in a configuration management database.

Perform comprehensive testing for new hardware or software. Maintain separate environments for development and testing. Proposed changes to the production environment should be documented and justified.

From the guidelines, here are specific steps for your own design.

Determine your availability requirement.

This comes from understanding HA and your costs of downtime (discussed above).

Characterize current availability. Document all Single Points of Failure.

Determine how available is your current infrastructure. Record the sources of unplanned downtime. Document your SPFs.

SPFs are sometimes obvious. The power supply, connection to the Internet, disk drive, network component come to mind when designing for HA. Sometimes SPFs are very unexpected. For example, recently it was reported that Microsoft.com, msn.com, expedia.co.uk and msnbc.com were all unavailable[viii] for a period of several hours. There is speculation that the root cause was a configuration change to a router at the edge of Microsoft's DNS network. The questions are interesting and important. Was it human error or an error in the process for configuring routers? Did this change undergo appropriate levels of review and justification? Which processes should be changed to insure this doesn't happen again?

Evaluate applications. Determine changes to be made and plan development accordingly.

This assessment will depend on your current applications architecture. Some architectures will be better than others for Datacenter. Today, the best Datacenter candidates are n-tiered, component-based "Line of Business" (LOB) applications designed to store and retrieve data from a back-end data service using MS SQL2000. These are "best" for Datacenter because they leverage so much of the development, scalability and HA technologies developed by Microsoft. Soon, we may add Oracle and Microsoft Exchange 2000 to the list (only after Microsoft Exchange 2000 SP1). Further testing will determine whether Oracle or Exchange will be good applications for Datacenter. For now Microsoft SQL 2000 is the only application to take full advantage of the large memory space and four-node clustering in Datacenter.

Designing a HA infrastructure for.NET is straightforward. If your application can be logically divided into Presentation, Business Logic and Data Services tiers, clustering options are clear. Microsoft provides several clustering technologies that map well to n-tiers. For Presentation, use IIS 5.0 (with reliable restart) on Network Load Balance (NLB) clusters. NLB is available starting with Windows 2000 Advanced Server and supports up to 32 nodes per cluster. NLB provides stateless clustering with stateful connections to client browsers[ix]. Alternatively, load balance IP traffic using Cisco LocalDirector (CLD-a hardware solution[x]). Some prefer CLD because of its support for network switches (NLB requires hub's in conjunction with switches due to NLB-induced switch flooding[xi]).

The Business Logic tier is supported using Component Load Balancing (CLB) clusters on any Windows 2000 server. CLB requires Microsoft Application Center 2000 and dynamically load balances COM+ components. CLB clustering is stateless and supports up to 16 nodes. Although CLB runs on all Windows 2000 servers, there are good reasons to consider Datacenter for the Business Logic tier. First if the application is stateful, you need the Datacenter implementation of Microsoft Cluster Service (MSCS) to scale out beyond two nodes. You also need Datacenter to scale up above the eight-processor SMP and eight GB limit of Windows 2000 Advanced Server. Second, even for applications not written to take advantage of Datacenter's large memory space (i.e. not written with Address Windowing Extensions-AWE), there could be substantial benefit to running on a large memory Datacenter System. Large system cache and reduced physical paging could have major performance

benefits. Finally, there may be performance benefits using WinSock Direct (WSD) for communications with Data Services.

The last piece of standard .NET architecture is Data Services. Data Services (in this case Microsoft SQL 2000) benefit from scale up, i.e. expanding the memory or number of SMP processors on a single system. Databases are typically stateful and use Microsoft Cluster Services to scale out or add redundancy. Windows 2000 Advanced Server supports two-node stateful MSCS clusters. Advanced Server also supports scale up to eight SMP processors and eight GB of RAM. Windows 2000 Advanced Server is a very stable platform with many scalability and HA capabilities built in. Reasons for choosing Datacenter over Advanced Server include; a scale up requirement greater than 8 GB of memory, a stateful clustering requirement for three or four nodes , a network performance requirement for WinSock Direct, need for a GUI-based Process Control tool or a support requirement for the extended testing and Joint Support Queue available only with the Datacenter Program. The Datacenter Program also provides change management by stress testing, signing and certifying kernel-mode drivers, certifying hardware changes and certifying applications software.

Applications not constructed according to DNA or .NET can still benefit from Datacenter. To take best advantage of four-node clustering, applications should be modified to use the MSCS cluster API. Best use of large memory requires that Address Window Extensions (AWE) be added to your scale-up application. Unmodified applications can still benefit from MSCS and PAE, just not as much. Applications using MSCS should, however, provide for their own data recovery using checkpoint files. Checkpoint files are important for recovering transactions in memory during a fail over. For large database applications, these decisions will need substantial justification. For applications not using COM+ or using stateful COM+, the programming may be easier. COM+ components are typically small and easy to modify. Stateful COM+ components could be modified to become stateless or to store state information in Data Services.

Select HA components and redundancies consistent with availability requirements.

Once you've established availability requirements, characterized current availability and analyzed applications, you're ready to select HA components to meet your needs. Scaling, cluster, memory and SMP are all variables in the mix. The correct configuration will likely result from benchmarking the application in a test environment.

Even though you need higher levels of availability, is Datacenter the best answer? Windows 2000 Advanced Server provides many excellent and affordable availability improvements including two-node MSCS clusters, 32-node NLB clusters, 16-node CLB clusters and reliable restart in IIS 5. Combined with disciplined backup, recovery and change management, it's entirely possible to reach two or even three nines with Advanced Server. Answers to the following should help you decide if you need Datacenter. Do you need to scale up beyond eight GB of memory and eight processors? Do you need the extra redundancy provided by three or four-node MSCS clustering (over two-node available with Advanced Server)? Is your application "cluster aware" or are you willing/able to modify the application to make it so? Is there some level of Data Redundancy (RAID 1, 5 or 10) that would satisfy your availability needs? Here are good candidates for Datacenter today.

Line of Business Application based on COM+ and using Microsoft SQL Server 2000.

Mission critical application needs stateful clustering beyond two nodes

Mission critical application needs scale up expandability beyond eight GB of memory and/or eight processor SMP

Large Microsoft SQL2000 databases (either for OLTP or Data warehousing)

Large File servers

Select system administration tools.

BODY COPY Not part of the formal Datacenter Program, system administration tools are still an important part of HA. Find tools, which help you manage all components of the infrastructure. These typically use SNMP and agents to monitor the condition of your site, trap errors, generate alerts, carry out preprogrammed responses to various conditions, identify dependencies between components and perform root cause analysis of dangerous trends. Hewlett-Packard's OpenView and Computer Associates' Unicenter are examples. Large servers often ship with management utilities such as HP Toptools or Compaq Insight Manager that allow detailed health investigation of hardware while still fully online. If the server is remote, remote control cards can recycle power, monitor a boot sequence and provide remote keyboard/video/mouse capabilities.

Develop, document and train on procedures to follow on failure. Regularly review and rehearse procedures. Revise as needed.

The Datacenter Program goes a long way toward formalizing support for HA infrastructure. In the Datacenter Program, Microsoft, the OEM partners and Certified Application developers, do extensive testing and "change management". OEMs must offer SLAs on time-to-repair for support. The Joint Support Queue, staffed by both Microsoft and OEM personnel provides a well-defined escalation path. Some OEMs offer consulting and support beyond the Program's minimum requirements. In addition to an HA design, make sure support (both internal and from third parties) is aligned with your availability requirements. Establish and enforce SLAs for external and internal support teams

If "Mission Critical", or "High Availability" are new to you or your organization, learn about IT Service Management and the IT Infrastructure Library (ITIL). Standards groups in the UK started codifying best practices for HA into the IT Infrastructure Library in the late 1980s. Most successful HA sites today use some or all of these practices. There are a number of training programs and publications to introduce you and your staff to the world of HA computing and IT Service Management (ITSM).

The Microsoft Operations Framework (MOF) builds upon the ITIL and, according to Microsoft, is better suited to the rapidly changing needs of Windows environments. The MOF emphasizes iterative processes for risk assessment, configuration management and adoption.

For large multi-vendor, multi-OS environments, HA is possible but achieving it requires more involvement by the IT Department. There's a "make or buy" decision. To "buy", turn facilities design and management over to a third party. Focus your efforts on integration and SLA enforcement. Identify, document and resolve support gaps. To "make", be prepared to assume the tasks described here. Change management, testing, documentation, help desk support and more should all be handled. ITSM or the MOF can provide a good starting point to determine what procedures are appropriate for your environment.

Document system state. Plan for change; define change management roles and processes and build scope into the HA design for redundancy that can facilitate and expedite change management and recovery.

Leverage best practices from ITSM and/or MOF and use system management tools to document the state of your HA infrastructure. Create a Configuration Management Database (CMDB) with information on Configuration Items (CI's) and dependencies between them. This record should be comprehensive and contain information on every component involved in keeping your infrastructure available. Configuration settings, firmware versions, build or service pack numbers and more should be included. A record should be made during the initial installation and updated after any change. This will be of help in troubleshooting and getting a failed system back into service (decreasing MTTR). There are tools from Microsoft and third parties to help with this. At installation, run the Datacenter

config comparison utility (cfgcmp.exe) and Microsoft PSS utility to get baseline data on hardware, software and detailed system configuration. On stateful clusters (MSCS with two, three or four nodes) create an initial cluster log. Run network diagnostics (netdiag.exe) to confirm and document that there are no network problems. Finally, save your event logs to document that there are no errors or warnings on boot.

Define processes whereby a Request for Change (RFC) is created and submitted. Define roles and responsibilities for a Change Advisory Board (CAB) to determine how any RFC will impact Availability, Capacity and Service Level Agreements (SLAs). Once an RFC has been authorized, have formal procedures to implement the change and record new or changed CI's in the CMDB. Make sure the CMDB is available for root cause analysis of any failures. As appropriate, create new RFC's to address design changes after a failure.

Troubleshooting is often on the critical path to recovery. Faster recovery means lower MTTR and higher availability. Expedite recovery by designing for fast troubleshooting. Put parallel installations of Windows 2000 on all servers to provide redundant "safe boot" on failure. Create, document and practice procedures for handling a stop screen, Dr. Watson message, hung server and hung process[xii]. Make sure support personnel are familiar with these procedures.

Summary - Call to Action

The challenge for IT infrastructures is clear. The stop screen, or "Blue Screen of Death", must not happen (or if it does, fail over systems must quickly restore lost functionality). Run away applications with memory, handle or semaphore leaks must not be allowed to affect the rest of the system or other users' applications. Software that touches the kernel must be thoroughly tested. Applications must not be allowed to replace or modify Windows system files. All hardware must be stress tested. All problems that have been, or could possibly be encountered must be avoided or accommodated for systems to be "Highly Available". The entire organization must understand and support the need for new, more structured approaches to enterprise computing.

The challenge for network administrators is also clear. Take full advantage of the experiences, tools and support that's available. If HA is your goal, use the Datacenter Program to help you achieve it.

---

[i] Originally attributed to Lord Kelvin (William Thompson) quoted in D MacHale, *Comic Sections* (Dublin 1993)

[ii] Marcus and Stern, **Blueprints for High Availability**. Wiley. 2000

[iii] Marcus and Stern, ibid.

[iv] Source: Contingency Planning Research. http://www.hwcs.com/html/cicsbatch_cost_of_downtime.html

[v] **IEEE Computer** April 1995 with additional annotations by author.

[vi] HP relies heavily on ITSM in designing HA infrastructures and supporting Datacenter customers. The lynchpin of ITSM is the combination of "Change Management" and "Configuration Management". These interact with each other as well as other ITSM processes. HP groups these remaining processes as "Business – IT Alignment", "Operations Bridge", "Service Design & Management" and "Service Development and Deployment".

[vii] "Increasing System Reliability and Availability with Windows 2000" on the Microsoft web site, http://www.microsoft.com. The paper was written in November 2000, prior to the Datacenter release. It's a nice review of the reliability improvements in the base OS.

[viii] John Leyden. **The Register**. Web-based newspaper. Posted 1/25/2001 at 11:43 GMT. He quoted a Microsoft source, "In a statement, Microsoft admitted: 'At 6:30 p.m. Tuesday (PST), a Microsoft technician made a configuration change to the routers on the edge of Microsoft's Domain Name Server network. The DNS servers are used to connect domain names with numeric IP addresses (eg. 207.46.230.219) of the various servers and networks that make up Microsoft's Web presence... The mistaken configuration change limited communication between DNS servers on the Internet and Microsoft's DNS servers. This limited communication caused many of Microsoft's sites to be unreachable (although they were actually still operational) to a large number of customers.'"

[ix] "Stateful" and "stateless" refer to the component or application's need (or lack thereof) to retain state information about ongoing client sessions. Databases are typically "stateful", retaining information about the state of the database before, during and after every transaction. It's the "stateful" nature of databases that requires log files to recover the database after a failure. An example of "stateless" applications is an HTML page. Clients connect to the web server, retrieve the page, and disconnect. The HTML doesn't need to change as subsequent clients make requests. Even though the HTML page is stateless, the client's connection could be stateful (to allow secure transactions, logon, etc.). That's why NLB supports stateful web connections. If state information is required, it's often stored at the client as "cookies" or in a table in the back-end data service. COM+ components are often designed to be "stateless". MSCS is the clustering solution best suited to stateful applications such as Microsoft SQL 2000. NLB and CLB are intended for stateless applications such as serving HTML pages and sharing COM+ components.

[x] For more on the Cisco LocalDirector, see the web site:
http://www.cisco.com/warp/public/cc/pd/cxsr/400/index.shtml

[xi] For more on use of network switches with Network Load Balancing, see the Windows 2000 Resource Kit.

[xii] For more on debugging stop screens and the OEM diagnostic tools available from Microsoft, see the Knowledgebase article Q253066.