the Availability Digest

Towards Zero Downtime: High Availability Blueprints

August 2007

Vishal Rupani focuses on Microsoft Clustering in his very readable book, *Towards Zero Downtime: High Availability Blueprints*.[1] He first provides a high-level overview of many topics pertinent to high availability. He then devotes most of his book to Microsoft Clustering and to the proper installation of Microsoft's Cluster Server and several of Microsoft's cluster-aware applications. He follows this with a brief discussion of disaster-tolerant architectures, and concludes with a case study.

High-Availability Topics

Rupani begins with a definition of several high-availability concepts, such as planned and unplanned outages, single points of failure, mean time between failures and interruptions, fault tolerance, and redundancy. He then provides an overview of several important high-availability technologies.

RAID

RAID technology is the most common method to provide data resiliency. With RAID (Redundant Arrays of Independent Disks), data is redundantly stored on multiple disks such that, should one disk fail, the data can still be reconstructed. RAID arrays typically provide a hot-swappable capability so that a failed disk can be replaced without taking down the array.

There are several configurations of RAID. Most stripe data across multiple disks to achieve improved performance. Mirrored disks, used by fault-tolerant systems and designated as RAID 1, provide an entire copy of the database on a backup disk.

The most common form of RAID in use today is RAID 5, which provides one additional disk and which stripes data and parity across disks. Since there is one extra disk, the data can be reconstructed should any one disk fail.

There are several other forms of RAID designated as RAID 0, 2, 3, and 4, and combinations of these. Rupani describes each of these configurations.

High-Availability Architectures

Rupani divides system architectures into several levels of availability:

- *Unmanaged Tiers* are implemented generally without availability in mind and typically achieve availabilities in the order of 90%.

[1] Vishal Rupani, *Towards Zero Downtime: High Availability Blueprints*, 1st Books; 2004.

- *Managed Tiers* use low-end servers, perhaps with RAID storage arrays and uninterrupted power supplies (UPS). They have many single points of failure and exhibit availabilities in the order of 99%.

- *Well-Managed Tiers* use higher-end servers with hot-swappable RAID arrays and with perhaps dual processors, dual power supplies, and other components to eliminate some single points of failure. Routers may be used to direct transactions to multiple servers to achieve load balancing or a degree of fault tolerance. These systems strive to achieve 99.9% availability.

- *Fault-Resilient Tiers* incorporate redundant components to eliminate single points of failure within the servers and storage systems, though there may be single points of failure at network connection points. A cold backup site may be provided for disaster recovery. These architectures can achieve four 9s of availability (99.99%).

- *High-Availability Tiers* are clustered systems with automatic failover capabilities. All network connections are redundant, and the systems are powered through a UPS. System monitoring is provided, and the system undergoes periodic tests to ensure proper failover. These systems can achieve five 9s of availability (99.999%).[2]

Discovery

The design of a high-availability solution should begin with a Discovery process, in which an organization's needs and candidate solutions are determined. The organization's needs include:

- its central purpose.
- its business units.
- previous attempts at achieving high availability.
- existence of incident or problem management.
- description of the business critical applications, their acceptable service levels, their cost of downtime, and their acceptable downtime.

Candidate solutions cover:

- networks
- storage
- servers
- applications
- security
- monitoring

Storage Interconnect

The author describes the primary technologies today for interconnecting high-speed storage with servers.

- *SCSI* (Small Computer Systems Interface) is a parallel I/O bus which has been in use for over two decades and is still the predominant interconnect today. Speeds in the tens or hundreds of megabytes per second can be achieved. However, distance limitations are measured in tens of meters; and only a handful of devices can be connected to a SCSI bus.

[2] We would add active/active systems to this list. Active/active systems achieve six 9s availability and beyond.

- *Fiber Channel* (FC) offers full duplex gigabit/second speeds over distances measured in kilometers. Its addressing capability is virtually unlimited. It can be used in several topologies such as point-to-point, arbitrated loop (similar to a token ring network), and switched fabric. It is rapidly becoming the dominant storage interconnect technology.

- *Infiniband* is an emerging technology supported by many of the major server vendors. It provides multi-gigabyte speeds over distances exceeding that of fiber channel. It is intended to replace the PCI bus found in contemporary servers.

Storage Technologies

Towards Zero Downtime continues with descriptions of direct attached storage, network attached storage (NAS), storage area networks (SAN), and storage virtualization.

With direct attached storage, storage units are directly connected to the server, usually via SCSI or fiber channel. The server must provide all file server tasks while at the same time handling its business tasks. Though the simplest of all of the storage technologies, direct attached storage is not very scalable; and its availability is limited to that of the direct attached RAID arrays or whatever other storage mechanism is used.

Network attached storage (NAS) is similar to direct attached storage except that the storage units are connected to the network and are accessed through the network by the servers. They are stand-alone appliances that provide all file and database services, thus offloading these tasks from the servers. NAS storage is highly scalable and significantly minimizes administration costs.

Storage area networks (SANs) provide pooled storage for a network of servers. Storage is divided into logical units (LUNs), which are allocated to servers as needed. SANs can include tape libraries that can be used for backup independently of any of the servers in the network. Storage administration can be centralized for all storage requirements of the enterprise.

Storage virtualization is a convergence of SAN and NAS technologies over fiber channel. With storage virtualization, a storage controller sits between a pool of storage devices and the file servers. The virtualization manager allocates storage to applications either as LUNs or as files.

Clustering Technologies

Clustering is a mature technology that allows two or more independent systems to work together as a single system. The primary benefit of clusters is that single points of failure are eliminated, thus providing significantly improved availability over single systems.

There are several cluster models:

- *Active/passive*, a shared-nothing architecture in which the resources required by a particular application can be active on only one node at a time in the cluster. Should a node fail, the application and its resources are failed over to another node.

- *Active/active,*[3] a shared device architecture in which multiple processors in a cluster can share a common resource such as a database. Should a node fail, users need only reconnect to another node that is currently processing that application. Active/active architectures require distributed lock management to prevent data corruption due to

[3] Rupani's definition of the term "active/active" is somewhat different from that of others in the cluster community. The more common definition of an active/active cluster is that different applications are being run on each node, but no application runs on multiple nodes. The system he describes is sometimes called a "multi-instance" cluster. His use is also different from our use of active/active. In a cluster, there is only one copy of the database. Therefore, the cluster cannot be separated geographically for disaster tolerance as can an active/active system that employs replicated database copies, the context in which we use the term.

multiple servers trying to update the same data. Therefore, they must use specialized database management systems.

Clusters are useful for high-availability systems, for load balancing between servers, and for high-performance computing using parallel processing. Clusters can reduce planned downtime since they can be upgraded online with rolling upgrades. Using this technique, the applications running on one node in the cluster are failed over to another node; and that node is taken down and upgraded. The upgraded node then rejoins the cluster, and the upgrades are rolled to other nodes using the same procedure.

Typically, an application should be cluster-aware to run in a cluster. Such applications can interact with the cluster's services and can take advantage of them via an application programming interface (API).

A cluster-unaware application can run in a cluster provided it meets certain criteria. For instance:

- It must not share temporary files on shared disk.
- It must not delete registry keys.
- It must not change data structures.

Clustering products are available on UNIX from the top UNIX vendors (HP, Sun, IBM), on HP's OpenVMS, from Microsoft, and from others.

Microsoft Clustering

The primary focus of this book is on Microsoft clustering. The material presented in the first part of the book lays the groundwork for understanding Microsoft clustering. The bulk of the book is then devoted to this topic and especially to the details of installation, verification, and testing for Microsoft clusters and several Microsoft cluster-ready applications, including:

- SQL Server
- Internet Information Server (IIS) Cluster
- File Share Cluster
- NLB Cluster

Other Microsoft cluster-aware applications include Exchange Server, DCHP Server, and Print Server.

The installation and test procedures are extensively documented with screen shots of the Microsoft procedures.

Microsoft Cluster Server

The Microsoft Cluster Server initially ran on Windows NT and now runs on Windows 2000 and Windows 2003, Enterprise Edition. Rupani uses Windows 2000 as the example platform.

The installation procedures for Microsoft Cluster Server involve installing the operating system on the cluster nodes, creating shared volumes, configuring the networks, installing and configuring the Cluster Server, and verifying the proper installation.

The author then walks through the procedures for joining any node in the cluster and for testing proper failover. Failover is checked by using the Initiate Failover command provided by the Cluster Administrator, by powering down each node one at a time, by disconnecting the public network from each node one at a time, and by disconnecting the shared storage from each node one at a time.

Rupani provides several tips for cluster installation, including:

- Place the paging (swap) file on a local drive, not on a shared volume.
- Schedule manual failovers periodically to test failover.
- Ensure write-back caching has been disabled.
- Be aware that some virus checking software can cause blue screens of death.

SQL Server

SQL Server installation is described in a similar way. SQL Server failover tests are detailed, and tips are given. Tips include:

- Use SQL Server in an active/passive configuration.
- Backing up SQL Server data should be a priority activity.
- For optimum data protection, use log shipping to move data from the production cluster to a backup site.

Internet Information Server (IIS)

IIS is used to provide services to Web applications. It is often used with Network Load Balancing (NLB) rather than with Cluster Services unless fault tolerance is required.

Decisions must be made whether to keep related data local to each server or to put it on a shared volume. If data changes infrequently (such as a Web page), it may be best to keep copies of the data on local storage. If it changes frequently, it may be better to have it reside on shared storage.

The procedures for installation, verification, and testing are detailed with a full set of screen shots. The author notes that Microsoft's FrontPage Server Extensions are not supported on clustered Web sites.

File-Share Cluster

Providing a file-share cluster can significantly improve system availability by providing redundancy for the file servers. A file-share cluster first requires a Network Name resource. This resource should not be dependent upon the Cluster Name resource.

Once a Network Name resource has been created, Rupani describes the sequence of steps required to create a file-share resource.

Network Load Balancing (NLB)

A Microsoft Network Load Balancing cluster distributes incoming client requests among the servers in the cluster in an attempt to balance the workload. To a client, the NLB cluster appears as a single server which is highly scalable (up to 32 servers) and is fault tolerant.

NLB does not have a single server that receives and routes all requests. Rather, all servers listen on the same IP address and discard packets that are not for them.

The request load can be equally distributed to all servers in the cluster, or each can be configured to handle a specified portion of the load.

The author details the installation and validation procedures for an NLB cluster.

Disaster Tolerance

Even if a data center is fully redundant, the data center itself represents a single point of failure should it be put out of commission by some disaster. To protect itself from such an occurrence, a company must provide alternate data processing facilities at another site.

Disaster recovery is the ability to return IT services to an acceptable level of operation following a site outage. It is a subset of a Business Continuity Plan. Disaster tolerance, on the other hand, is the ability to maintain ongoing IT services even in the face of such a catastrophe.

In a truly disaster-tolerant configuration, a remote site is kept in near-real-time synchronization with the primary site via data replication. The combination of clustering, data mirroring within the cluster, and data replication to a remote site provides a truly disaster-tolerant configuration.

Case Study

Rupani concludes with an actual in-depth case study conducted for a company running SAP applications with Oracle databases on Windows 2000 systems. Following an intensive discovery process, several recommendations are made. The study concludes with recommended changes in procedures and configurations and their associated costs.

Summary

Vishal Rupani focuses on the use of Windows clustering techniques and products. He introduces the topic by covering a broad range of availability issues such as storage and processing redundancy. He highlights the need for an extensive discovery process to understand the client's current systems and future needs.

He then details the installation, validation, and test procedures for Microsoft clustering and several Microsoft cluster-aware applications. He follows this with a brief discussion of geographically-distributed fault-tolerant architectures and concludes with an in-depth case study that applies the concepts covered in his book.