Server Clustering

At one point in time only a single processor was needed to power a server and all its applications. Then came multiprocessing, in which two or more processors shared a pool of memory and could handle more and larger applications. Then there was a network of servers, each dedicated to a different set of applications. Today, there are server clusters, where two or more servers act like a single server, providing higher availability and performance than ever thought possible. Applications can move from one server to another or run on several servers at once - all transparent to the users.

Granted, clusters are not really brand new, but until recently they have been proprietary in both hardware and software. Information Systems managers are looking at clusters more seriously now as they become more accessible using mass-produced, standards-based hardware like RAID, symmetric multiprocessing systems, network and I/O adapters and peripherals. And while clusters will surely gain more technical advances in the future, a growing number of cluster options are available today, even while real standards for clustering are still being developed.

What is Clustering?

A simple definition of clustering is two or more computers, or nodes, in a group that provide higher availability and scalability than would be possible if the computers worked separately. Each node in the cluster typically has its own resources (processors, I/O, memory, OS, storage) and is responsible for its own set of users.

Availability is provided by failover functionality: when one node fails, its resources can "fail over" to one or more other nodes in the cluster. Once the original node is restored to full operation again, its resources can be manually or automatically switched back.

Scalability is provided with the ability to add processing power or disk capacity without interruption of service. In effect, upgrades can be performed by proactively "failing over" the functions of a server to others in the cluster, bringing that server down to add components, then bringing it back into the cluster and switching back its functions from the other servers. Additional scalability is delivered with distributed message passing (DMP), an intracluster communications technique that allows applications to scale beyond the single symmetric multiprocessing (SMP) system in a way that is transparent to end users.

Each node in the cluster must run cluster software that provides services such as failure detection, recovery and the ability to manage the servers as a single system. The nodes within a cluster must be connected in a way that each node is aware of the state of all the other nodes. This is normally accomplished with a communication path separate from the LAN path, using dedicated network interface cards to guarantee clear communications among the nodes. This communication path relays a "heartbeat" between systems so that if a resource fails, thereby failing to send a heartbeat, the failover process will start. In fact, the most reliable configurations employ redundant heartbeats using different communication connections (LAN, SCSI or RS232) to ensure that a communication failure does not invoke false failover.

Levels of Clustering

Luckily for the cluster shopper today there are many levels and variations of clustering that can provide a broad spectrum of availability. As expected, higher levels of availability come at greater expense and complexity of management.

Shared Storage

A shared disk subsystem is frequently the basis for clustering, using shared SCSI or fibre channel. While each node uses its local disks to store operating system swap space and system files, the application data is stored on the shared disks, and each node can read the data written by other nodes. A distributed lock manager (DLM) is required for concurrent disk access among the applications, and the distance between the shared disk subsystem and its clustered nodes is limited by the choice of medium (SCSI vs. fibre channel, etc.).

Server Mirroring (Mirrored disks)

Environments that need data redundancy without the expense of external disk subsystems have the option of mirroring data between servers. In addition to a lower cost, another benefit of server mirroring is that the connection between primary and secondary server can be LAN-based, which can remove the SCSI distance limitation. When data is written to the primary server, it is also written to the secondary server; data integrity is maintained by locking the secondary server data. Some server mirroring products can also switch workload from primary to secondary server.

Shared Nothing

Today, some cluster products use a "shared nothing" architecture, where the nodes neither share centralized disks nor mirror data among nodes. In a failure situation, the shared-nothing cluster has software that can transfer ownership of a disk from one node to another, without using a distributed lock manager (DLM).

How Does Failover Work?

There are a number of ways a cluster can be configured for failover. The first is the N-way configuration, where each node in the cluster is normally active, with its own set of users and workload. A failed node's resources can failover to other nodes, which can cause some degradation of the remaining servers' performance as they take on the additional workload.

On the other hand, an N+1 configuration includes a hot standby system that is not normally active, but waits in idle mode for a primary system failure. While a node failure in the N+1 configuration avoids performance degradation in the other nodes, the cost of overhead is higher since the standby is not normally an active service provider.

In any configuration, the cluster software should be able to provide local recovery as a first option when a problem occurs. Local recovery is the ability to automatically restart applications or services on a local node if they fail. Local recovery is logically the preferred action for failures that are not fatal to the node since it causes less disruption to the users than failover to another node.

As for failover variations, some clustering products can perform parallel recovery, wherein multiple resources are recovered in parallel instead of sequentially. There is also the capability for wide area failover, wherein resources can failover to a geographically remote node. This is appropriate for disaster-tolerant requirements. In addition, to account for multiple node failures, some clustering products can perform cascading failover, which works in a domino fashion: node one fails over to node two, which then experiences its own problems causing it to failover to node three, and so on.

Failover Example

Here's an example of a failover scenario in a two-node cluster, where both nodes have their own users and active applications.

1. Node 1 experiences a memory problem that causes an application failure. Users experience error messages and their application dies. The cluster management software notifies the system administrator of the problem.

2. Node 1 performs local recovery, restarting the failed application. Users are able to restart their application.

3. When the application fails again, the cluster software initiates failover to node 2. Users experience what appears to be a hang condition for approximately one minute while the failover takes place. (Actual time can range from seconds to minutes.) Some applications may be able to detect the failover process and display a message to users that the application is being transferred to another server.

4. The application and client communications are switched over to node 2. Once the application is restarted on node 2, users are able to continue their work.

5. Node 1 is diagnosed and repaired. After it is brought back up, the failback (switchback) process is initiated so that the application and associated resources are brought back to node 1. This failback can be run manually or automatically. For instance, it may be configured to failback during non-peak activity.

Scalability with Clusters

Beyond increased availability, performance scalability is a major benefit of clusters. Performance in general can be improved through load balancing across the cluster. Essentially, load balancing means moving related applications and resources from a busy node to a less busy node.

But the real scalability comes in other areas. The first area is incremental scalability, meaning the ability to add more and more servers, disk storage, etc. without throwing out previous systems. In effect, clusters provide a pay-as-you-go environment as your computing requirements grow. The second type of scalability will be seen when true "cluster aware" applications are developed that can automatically spread their workload over multiple nodes in the cluster. Beyond that, it will be possible to split applications so that different "threads" of an application can run on different nodes, thus enhancing scalability even more.

What About Applications?

The next logical question is "How do applications handle failover?" The answer is "It depends upon the application and the clustering product being used." Some cluster products provide recovery or switchover kits for specific applications such as databases or communications protocols. These kits provide the ability to detect when the application has failed, and restart the application on a new server.

The way applications handle failover can vary between cluster products. As mentioned earlier, there are no common standards for cluster software today, though various vendors are working to develop them.

While applications today must be modified to handle failover, the ultimate goal is for applications to remain ignorant of the underlying hardware. One solution is a set of programs and APIs (Application Programming Interfaces) running in conjunction with operating systems that allow application vendors to create libraries that perform these restore functions. These APIs are used to make applications "cluster aware.” Many vendors of current cluster products are working to ensure they will be compliant with these various operating system APIs.

Virtual Interface Architecture (VIA)

The Virtual Interface Architecture (VIA) initiative, led by Intel, Compaq, HP, Microsoft, Novell, SCO and Tandem is working to define standards for developing clustering hardware and software that will be vendor-independent, allowing users more choice in where they buy their technology.

Key Points to Remember

- True clustering can be thought of as the next step in the multiprocessing evolution - applications that once ran across multiple processors in one system can now be spread across multiple processors in several systems.
- Clusters provide two major benefits: high availability (through failover functionality) and scalability (through incremental growth and load balancing across processors).
- Failover occurs when a node experiences a hardware or software problem, and its applications, along with communication connections, are switched over to another server. The cluster management product can be used to specify which applications should failover, as well as which failure conditions trigger the process.
- Many clustering variations and configurations are available to provide users the exact level of availability they require. Shared disks, server mirroring, and shared nothing are a few of these configurations.