“Achieving Century Uptimes”
An Informational Series on Enterprise Computing

As Seen in *The Connection*, An ITUG Publication
December 2006 – Present

About the Authors:

Dr. Bill Highleyman, Paul J. Holenstein, and Dr. Bruce Holenstein, have a combined experience of over 90 years in the implementation of fault-tolerant, highly available computing systems. This experience ranges from the early days of custom redundant systems to today’s fault-tolerant offerings from HP (NonStop) and Stratus.

***Gravic, Inc*.**
Shadowbase Products Group
17 General Warren Blvd.
Malvern, PA 19355
610-647-6250
www.ShadowbaseSoftware.com

Achieving Century Uptimes
Part 6: Active/Active versus Clusters

September/October 2007

Dr. Bill Highleyman
Dr. Bruce Holenstein
Paul J. Holenstein

The classic approach to achieving high availability is the use of clusters to provide processing redundancy. Cluster technology has been with us for decades.

In this series, we have focused on the use of active/active systems to achieve high availability. Are these two technologies not substantially the same? If they are not, what distinguishes one from the other?

The bottom line is that clusters are a five 9s technology. Active/active is a six 9s and beyond technology. Active/active systems can provide reliabilities that are greater by an order of magnitude or more than those provided by clusters. Active/active systems provide scalable support for applications, and they are naturally disaster-tolerant.

However, active/active technology is relatively new. Clusters are a mature technology with hundreds of thousands of successful installations. Production active/active systems are just now coming online in significant numbers.

We have described the active/active architecture extensively in our first article of this series.[1] This article assumes that the reader is familiar with that material. We therefore first review clustering technology; and we then compare clustering to active/active, leading to the above conclusions.

Clusters

A cluster is a grouping of two or more processors and associated storage configured to provide a single-system image to the users. An application can run on any operable processor in the cluster. Should its processor fail, the application will fail over to a surviving processor. The user is unaware of which processor is currently running his application. He sees the cluster as a single service provider that can survive any single failure.[2]

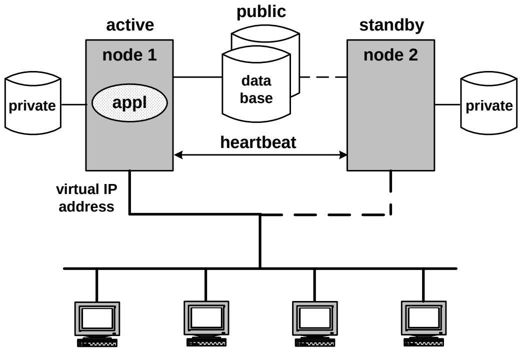
[1] Achieving Century Uptimes – Part 1: Survivable Systems for Enterprise Computing, The Connection; November/December, 2006.

[2] By this definition, a NonStop system is basically a cluster. However, its use of synchronized process pairs for instant failover has no parallel in contemporary cluster technology.

Cluster Architecture

There may be two or more processors (or nodes) in a cluster. These processors must generally be identically configured and be running the same version of the operating system.

There is also a redundant public database – typically RAID 5 or mirrored disk pairs (RAID 1) – that holds all of the application data. Each processor has a physical connection to the public database. However, a fundamental rule is that a particular database volume can only be mounted on one processor at any one time to prevent data corruption, which would otherwise be caused by multiple processors writing to the same record or row. (There are exceptions to this rule, which will be mentioned later.)

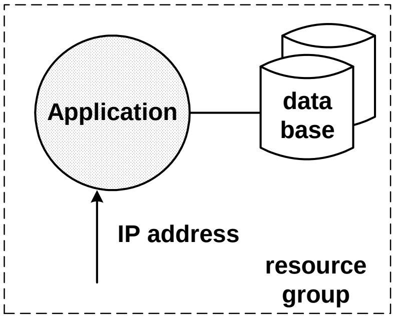


Each processor also has associated with it a private disk which is used for system software. Application software can either be resident on each private disk, or it may be resident on shared public storage. The considerations for this choice are discussed later.

The processors are interconnected with redundant heartbeat channels so that they can monitor the health of each other. They are also all connected to the user population by a LAN so that any user can get to any application no matter where it is currently residing.

Cluster Resource Group

When a processor fails, it is not the processor contents that are failed over. Rather, it is one or more *cluster resource groups*. A cluster resource group represents an application. It contains the application executable, the database volumes used by the application, and an IP address that is used to access the application.



Users generally know the application by its IP address. When an application fails over to another processor, its IP address is mapped to the physical address of the new processor upon which it is running. Therefore, any requests made to an application are always routed to the application no matter where it is running. This is the mechanism that allows a cluster to present a single system image to the user community.

Since only one processor in the cluster can access the data volumes used by a resource group, this means that there can only be one instance of an application running in a cluster. This rule governs much of our comparisons to active/active systems presented later.

Heartbeats

Heartbeats are exchanged between processors in the cluster so that the health of each processor can be monitored. Should a processor cease sending a heartbeat, it is declared out of service. The resource groups running on that processor are failed over to surviving processors so that service to the users can be continued.

It is imperative that the heartbeat network be highly reliable. Should the heartbeat network fail, the cluster becomes split into two groups separated by the failed network. If no provision for this failure is made, each group may consider that the other group has failed and will attempt to take over. The result, called “split-brain” mode, is that two processors will be writing to the same volume, causing severe corruption of the database.

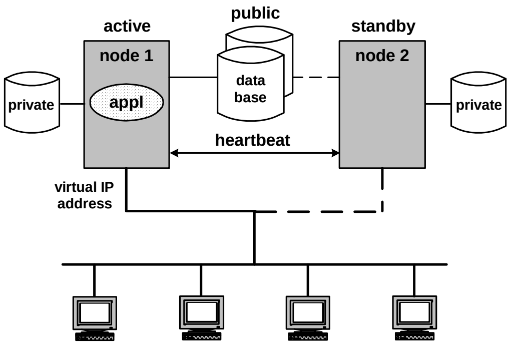
Heartbeat networks should therefore be redundant. Furthermore, if the network should fail anyway, adjudication facilities must be provided to determine which subgroup of processors should continue on and which subgroup should be failed to force its resource groups to fail over to surviving processors. This is often done via a cluster lock, which must be seized by the winning group, or by a separate and independent quorum server which monitors processor heartbeats and determines reconfigurations.

Failover services are provided by today’s Failover Management Software products such as HP’s ServiceGuard, Microsoft Cluster Services, and Sun Clusters.

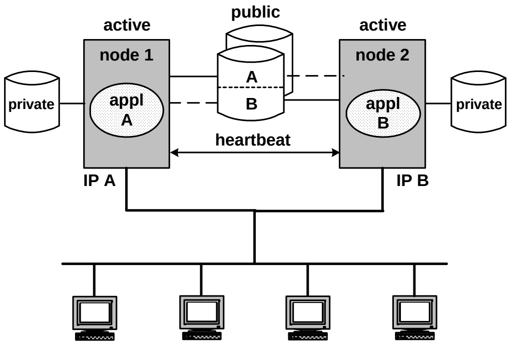
Cluster Configurations

There are two primary cluster architectures – active/standby and “active/active.” (We put “active/active” in quotes because this architecture does not conform to our definition of active/active architectures.)

In an active/standby cluster, all resource groups run in one set of processors, and one or more other processors are inactive, standing by to take over the resource groups of a failed processor.



Active/Standby Cluster



"Active/Active" Cluster

In an “active/active” cluster, each processor is running different resource groups. There can only be one instance of each resource group running in a cluster since only one processor can open the database used by an application. Multiple instances of an application cannot be running in a cluster.

There is one exception to this rule. Specialized Failover Management Software products can use a distributed database such as Oracle’s RAC (Real Application Clusters). RAC provides distributed lock management and distributed cache management so that multiple instances of an application can be run with each instance accessing common volumes.

Cluster Failover

When a resource group fails over to another processor, there are several tasks which must be accomplished. Once the loss of the heartbeat is detected:

- the volumes used by the resource group must be mounted by the new processor,
- the database must be checked for consistency (typically with *scandisk* or *fsck*),
- the application must be restarted in its new processor, and
- any required application recovery sequence must be initiated (such as rolling back incomplete transactions).

These tasks must be completed for each resource group to be failed over. Typically, failover takes several minutes or more.

Cluster Disaster Tolerance

Because of the requirement that all processors in a cluster must have physical connections to all database volumes, the cluster components must be collocated. Therefore, clustered processors cannot be geographically distributed to achieve disaster tolerance.

Disaster tolerance is achieved by creating one or more additional clusters that are geographically distributed. The databases of these clusters are kept in synchronism via data replication, just as are those in an active/active network. However, only one instance of any given application may be running in the entire cluster network at any one time.

Comparison of Clusters to Active/Active

Cluster architectures are very similar to active/active architectures. However, subtle differences lead to performance characteristics that are markedly different.

Availability – the Differentiator

One of the primary differences between cluster and active/active technologies is in the availability offered by clusters as opposed to active/active systems. This difference is due to the disparity in failover times. Clusters fail over in minutes, whereas active/active systems can have failover times measured in seconds. In addition, all users are affected by a cluster failure, whereas only the users on the failed node are affected by an active/active node failure.

This may not be terribly important in some applications. But in others, hourly downtime costs can be measured in six digits. If the hourly cost of downtime for an application is $600,000, then five minutes of downtime can cost the company in the order of $50,000. These are the companies that are seeking six 9s of availability (30 seconds per year of downtime) or better.

Let us compare a cluster with a five-minute (300 second) failover time to an equivalent active/active system with a failover time of three seconds. The cluster will be down 100 times as long as the active/active system, thus chopping off two 9s from the cluster's availability. A cluster providing four 9s of availability can be improved to six 9s by moving to a true active/active architecture.

A more detailed analysis using typical single-system availabilities of three 9s for industry standard servers (ISS) and four 9s for NonStop servers yields the following availabilities:

• Single ISS system	three 9s
• Single NonStop system	four 9s
• ISS Cluster	more than four 9s
• ISS Active/Active	almost six 9s
• NonStop Active/Active	almost eight 9s

HP has declared a corporate goal of achieving "5nines:5minutes" availability in its systems.[1] Clusters almost achieve this. Active/active technology goes far beyond this goal.

Disaster Tolerance

Disaster tolerance comes for free in an active/active architecture since the nodes in an active/active system can be geographically distributed.

A cluster requires that all processors be collocated. The only way to achieve disaster tolerance with cluster technology is to create additional clusters that are geographically distributed.

[1] P. S. Weygant, page 7, *Clusters for High Availability*, Prentice-Hall; 2001.

Heterogeneity

The nodes in an active/active system do not need to be the same. Not only can they be of different sizes, but they also can be using different versions of the application software, database software, or operating system or can even comprise processors from different vendors.

In a cluster, all processors typically must be identical, down to the version of the operating system that they are running.

Application Scaling

Active/active systems are inherently scalable at the application level. If an application needs more capacity, additional nodes can be added to the application network.

The only way to increase the capacity of an application in a cluster is to buy a bigger processor. Since the processors in a cluster must generally be homogeneous, this means that every processor in the cluster must be upgraded.

Zero Downtime Upgrades

In an active/active system, nodes can be upgraded without denying service to any user by simply moving users off of a node to other nodes, upgrading that node, and then returning its users. An upgrade can be rolled through an application network in this manner one node at a time.

A cluster can be similarly upgraded provided that the application and system code is resident on the private disks of each processor. However, this configuration brings with it the problem of version control. Since each processor in a cluster must generally be homogeneous, patching or upgrading software requires that all processors must be upgraded and coordinated. This versioning problem can be avoided by having all application and system executables be resident on the publicly shared disks. However, rolling upgrades cannot now be supported. The entire cluster must be brought down to upgrade software.

Maturity

Clusters have been around for a long time. The granddaddy of commercial clusters is the VAX Cluster (now the OpenVMS Cluster) that was introduced in 1984. Cluster technology has been in use for over two decades, and many good products are available to provide cluster management.

Active/active technology is relatively new. However, good data replication products to support active/active are now available; and active/active systems providing extraordinary availability are being put into production.

Summary

Active/active is the new guy on the block. Cluster technology has been around for decades. Product support for active/active is growing rapidly but is fairly new.

However, an active/active application network can provide significant advantages over a cluster. It can provide availabilities that are greater by an order of magnitude or more. Disaster tolerance comes for free. The scalability of applications is virtually unlimited. Nodes in an active/active network can be heterogeneous, and rolling upgrades are much more feasible. And, when a failure does occur, only a portion of users are affected in an active/active system versus all users in a cluster.

Clusters provide *high* availability. Active/active systems provide *continuous* availability. Watch for the rapid growth of active/active technology in fields such as financial, communications, and healthcare that require extreme availabilities.