Environmental Systems Research Institute, Inc., 380 New York St., Redlands, CA 92373-8100 USA • TEL 909-793-2853 • FAX 909-307-3014

ESRI Systems Integration Technical Brief

ArcSDE High-Availability Overview

Brian Price
Senior Systems Consultant
ESRI, Redlands, CA

September 23, 2004

Rev. 2

Introduction

An ArcSDE database is usually one of the most critical elements in a GIS implementation. This has become increasingly true as organizations have migrated from distributed file-based coverages and shapefiles to centralized geodatabases. Hundreds of ArcGIS users may be connected to a single database, along with ArcIMS and ArcGIS servers that are dependant on it for serving spatial and attribute data. A failure in a production ArcSDE server or process can dramatically affect the ability of the connected users to get their work done. A 911 system, for instance, which routes emergency teams based on the location of the incident and the available response teams, would be crippled if its data server is unavailable. For this reason, many organizations have a contingency plan for these outages by designing ArcSDE systems which are fault tolerant to some degree.

There are a number of issues to consider in designing an ArcGIS/ArcSDE environment to meet high-availability (HA) requirements. In this paper, we will examine the components of an ArcSDE implementation, the elements that are required for high-availability, and the options for implementing them. We will also discuss some of the implications in choosing the various options so that the reader can make informed decisions regarding these configurations. As with many enterprise-level systems such as this, the cost increases significantly with each decrease in amount of required down time. This is not meant to be a complete 'how-to' document, but it should serve as an introduction to the concepts and general implementation issues related to high-availability systems.

Since HA ArcSDE implementations are so dependant on the functions available in the database management software, database vendors are a good starting point for the most current information about the high-availability options for their products. We strongly suggest contacting them for more detailed implementation planning resources. Their Web sites are excellent sources and we've listed several of them at the conclusion of this document. It should be noted, however, that an ArcSDE geodatabase is more than just a database and has its own set of constraints and design issues which must be considered to successfully implement a high-availability configuration. Our discussion will highlight these issues.

This paper will cover the following topics:

- Definition of Terms
- High-Availability Design Strategies
- ArcSDE/DB HA Design Components
    - Data Storage
    - Database Management Software
    - ArcSDE Middleware
    - ArcSDE/DBMS Network Availability
- Application Server Considerations
    - ArcIMS
    - ArcGIS Server
    - Windows Terminal Server/Citrix
- Summary
- Reference Materials/Resources

Definition of Terms

We will begin our discussion with a definition of terms related to high availability as provided in www.webopedia.com (and other sources).

Failover – A backup operation that automatically switches to a standby database, server or network if the primary system fails or is temporarily shut down for servicing. Failover is an important fault-tolerance function of mission-critical systems that rely on constant accessibility. Failover can automatically, and transparently to the user, redirect requests from the failed or down system to the backup system that mimics the operations of the primary system.

Failback – The ability of a back-up operation to automatically switch back from the standby database, server or network once the primary system is restored to full operation.

Fault Tolerance – The ability of a system to respond gracefully to an unexpected hardware or software failure. There are many levels of fault tolerance, the lowest being the ability to continue operation in the event of a power failure. Many fault-tolerant computer systems mirror all operations; that is, every operation is performed on two or more duplicate systems, so if one fails, the other can take over.

Clustering – Connecting two or more computers together in such a way that they behave like a single computer. Clustering is used for parallel processing, load balancing and fault tolerance. Clustering is a popular strategy for implementing parallel processing applications because it enables companies to leverage the investment already made in servers and workstations. In addition, it's relatively easy to add additional capacity by adding a new server to the cluster in those environments that permit load-balancing.

Active-Passive Clustering – Active-passive (or active-standby) is a method of providing database system redundancy where the secondary systems are in hot standby mode and is only activated for full database transaction processing if the primary system fails. This is a common method for providing high availability in the geodatabase context and is implemented through clustering software such as Microsoft Clustering Service (MSCS, Veritas Cluster Software and Sun Cluster Software. In some implementations (i.e., Informix IDS) the hot standby can also be used for read-only processing, although the functionality of this with ArcSDE has not been tested and is not widely used.

Active-Active Clustering – Active-active clustering utilizes system redundancy as well; however, the secondary instance(s) are also in use providing database (or other) services to non-GIS applications. There are two different flavors of this. With the most commonly used products, the secondary platforms are normally in use serving non-GIS data to (i.e., ERP, other DB applications). The secondary platforms cannot access the primary server database while the primary is active, but it can be used to support other databases or applications. The second type is a newer active-active technology, Oracle RAC, which can be used in some environments to allow multiple instances to access the same database concurrently. These alternatives will be discussed in greater detail later in the document.

HA Design Strategies

High-availability capabilities can be implemented in hardware, in operating system software, in special clustering software, and in applications. In mission-critical systems, redundant elements with fail-over functions will be incorporated into many if not all of these system components. In less critical applications, they may only be implemented in some specialized hardware or in clustering software as mentioned above, or may not be included at all. The extent to which these high-availability functions are included in system design will vary with the risks associated with a system failure or planned outage and the cost to mitigate those risks. Prior to settling on high-availability (HA) architecture, we suggest that you examine the impact of geodatabase downtime on your organization and then perform a cost/benefit analysis on the target HA solution.

The goal in designing a highly available database solution is to eliminate as many potential single points of failure as possible within a reasonable cost constraint (typically a function of the cost of downtime associated with the solution). In the section below, we will outline the technologies available, starting with the least complex and expensive and then moving into the more complex solutions. Availability and costs increase with each option, and each option typically builds on the previous technologies.

Redundant Server Components - The first line of defense against ArcSDE outages are the server platforms. If you can build fault-tolerant DB servers with redundant hardware components (such as dual power supplies, storage controllers, network cards), multiple CPUs and RAID disk storage, many potential single points of failure are eliminated. These configurations allow the servers to continue operating even if a single component fails, and provides a great deal of benefit for little cost. This is highly recommended for an ArcSDE/DB environment.

Redundant ArcSDE/DB Servers - The next line of defense includes adding a second DB platform to serve as a backup to the primary server in case it fails. This platform could be a test or development server with the DBMS and ArcSDE installed, sized to handle the critical (but reduced) user load during a primary system failure. A more robust implementation would include a second server which mirrors the primary in its hardware configuration. It could handle the entire geodatabase production load if the primary system failed. The recovery scenario for a system failure includes providing the secondary platform access to the data storage subsystem on which the production database resides, starting the DBMS, moving the primary system IP address to the secondary system, testing the system to make sure it is operating properly, and then releasing the system for production use. This process can take several minutes to several hours depending on how well prepared the hardware, software and system staff is to accommodate these tasks. This option can double the cost of server hardware but it does provide the ability to recover from a server failure within several hours or less. Without a redundant server significant downtime will accrue while waiting for server parts, repair and testing.

Shared-none Clustered ArcSDE/DB Servers - The next higher level of availability creates a shared-none cluster with the primary and secondary platform(s). Clustering does not prevent the production geodatabase server from failure, but it does provide the ability to automatically recognize a production platform failure and move the production workload to the standby secondary platform. This ensures that ArcSDE/DB is quickly available to the users. The recovery scenario for clustering is very much like the one listed above, except all the fail-over tasks are performed automatically. This process typically takes 20 to 90 seconds. This can either be an active-active or active-passive shared-none configuration and generally does not add significant software or hardware expense once the redundant hardware is in place.

Shared-disk Clustered ArcSDE/DB Servers - In addition to shared-none clustering (active-active or active-passive) another level of availability can be achieved in some environments. In the shared-disk cluster each node has equal access to all of the information in the data store. Only the processing work is divided amongst them, and not the data itself. If one or more servers fails, all of the application data remains available to the other nodes which can continue operating. If multiple servers are updating the database, they all must synchronize their in-memory data structures, system tables and temporary data structures to avoid database corruption. In a GIS environment this is typically not feasible and for that reason (and the added expense) shared-disk clustering has not been successfully implemented. Oracle RAC is one example of this technology.

Before proceeding with designing failover cluster systems for ArcSDE, we should review the system components that make up an ArcSDE configuration.

- ArcSDE Service
  - GIOMGR
  - GSRVR
- DBMS (third party – i.e., Oracle, Informix, DB2, MS SQL Server)
- Operating Systems (MS Windows 2003 Enterprise Edition, Linux, HP-UX, etc.)
- Clustering Software (Veritas Cluster Server, Sun Clustering Software, Oracle TAF and RAC; MS Clustering Service, Linux Clustering, etc.)
- Server Platforms
- Network Connections
- Storage Sub-systems
- Connection to Storage Sub-systems
- Client Workstations
- ArcIMS Servers, Processes and DB connections
- ArcGIS Servers, Processes and DB connections
- ESRI License Manager

As you can see, there are quite a few components involved in serving data through ArcSDE to users, and each of these must be considered when designing high-availability systems. We will not attempt to cover every possible combination of hardware and software in this analysis; that is beyond the scope of this paper and, instead, readers should consult with hardware and system software vendors on the specifics of their implementations. In this paper we will focus primarily on failover clustering while examining some of the new load-balancing and 'scale-out' technologies and their suitability for a geodatabase environment.

ArcSDE/DBMS HA Component Design

The major components to protect in an ArcSDE configuration are:

1. The data (which means the data storage subsystems and the data which resides on them).
2. The executing database management software and the server, operating systems, etc., on which it depends.
3. ArcSDE services and connections.
4. ArcSDE/DBMS access to the network.

In the sections below, we discuss each of these areas, focusing primarily on the DBMS and ArcSDE services.

It should also be noted that the ArcGIS desktop and ESRI ArcSDE software components are not 'cluster aware' nor do they include any failover capabilities. This places the responsibility for these functions, if implemented, on other system components.

This discussion does not directly address disaster planning/recovery. A system designed for high-availability, however, will reduce some of the risks associated with catastrophic events and can be used to support disaster recovery. Shared-none clusters where backup nodes are placed offsite can be an effective way of implementing this and is in use in the GIS community, in particular in large Internet sites. In this scenario, data is replicated between sites and the cluster operation is supported through routers located at the Internet Server Provider (ISP).

1) The Most Critical System Component—The Data

Losing GIS data through disk crashes or other storage subsystem failures can be disastrous. Fortunately there are several strategies for preventing this kind of catastrophic loss. Section 7.6.4 in the *Systems Design Strategies* white paper (www.esri.com/systemsint/kbase/strategies.html) discusses various options for providing disk storage protection but, in general, a combination of RAID 5 and RAID 1/0 storage will provide ample storage loss protection while maximizing system performance and minimizing costs. RAID storage has become 'best practice' for database management systems.

Database or disk replication can also be used to safeguard data. As noted in the disaster recovery discussion above, data replication can allow organizations to maintain several copies of a database, either locally or at a distance. Replication can also support data publishing, data warehousing, data distribution and a variety of other business functions in addition to providing data protection. Hardware-level disk mirroring can also be used to maintain a second copy of a database. Hardware-level mirroring also avoids several potential problems inherent in replicating geodatabases.

In addition to RAID technology, best practices for data storage hardware configurations usually include redundant pathways/connections between the storage subsystem and each DBMS/file server, and the transparent failover capabilities for each of them. Actual storage protection recommendations are based on system availability requirements, storage architecture, and will vary depending on the overall GIS infrastructure and application architecture. Storage protection must also be considered in any disaster recovery planning.

2) Protecting the Database Management System

In this section we will discuss:

- DBMS Platform Configuration
- Transparent Failover
- Failover Scenarios
- ESRI Software Functionality
- ESRI License Manager

DBMS Platform Configuration - After the data itself, the database management system is the most critical component. Re-starting (or rebuilding) a server, the database, and re-establishing all the database connections is a complex operation and can take quite a bit of time. Best practice to protect a DBMS instance is system clustering, where several separate server platforms are configured either in active-active or active-passive mode as defined previously. This usually requires two or more servers to be configured appropriately. It also requires that each server have the DBMS software installed and have access to the storage subsystem on which the database tables reside. This can be a Storage Area Network (SAN), or direct attached storage (DAS). Network attached storage is not supported in clustering environments. Generally a SAN will provide the most flexibility in a failover configuration where multiple platforms must have access to the same physical storage.

Microsoft Cluster Service (MSCS), which is included with Windows 2003 Enterprise Edition, W2003 Datacenter, Windows 2000 Advanced Server and W2000 Datacenter works with the Windows operating system to configure multiple Windows systems into clusters (active-active or active-passive). Veritas Cluster Server provides similar capabilities in the Unix world and Sun Cluster Software operates in the Solaris environment. These products provide a basic cluster environment (known as a shared-nothing or shared-none), in which disks, IP addresses, and other cluster resources can be owned and accessed through only one cluster node at a time. When a database or application fails over from one cluster node to another, ownership of the disks, IP addresses, and other cluster resources associated with the database or application is quickly and automatically transferred to the new node. Two cluster nodes cannot write to the same disk at the same time (because all access is exclusively through the single node that currently owns the disk), and a given application workload cannot scale across multiple cluster nodes. This means that the database tables cannot be accessed by both the primary and secondary platforms at the same time. This kind of clustering is in widespread use in the GIS community and is the most common method for providing failover capability.

There are two different flavors of this. With the most commonly used products, the secondary platforms are normally in use serving non-GIS data or applications (i.e., ERP, other DB applications). The secondary platforms cannot access the primary server geodatabase while the primary is active, but it can be used to support other databases or applications. One newer active-active technology, Oracle RAC, can be used in some environments to allow multiple DB instances to access the same database concurrently. In theory, this could provide database scalability and load-balancing functionality in addition to failover capabilities by allowing servers to be added to the configuration as performance requirements increased. However, this technology has not yet been successfully implemented in the geodatabase environment and it is unclear as yet whether it can add value to an ArcSDE configuration. The major concern is performance. Each server must maintain an in-memory copy of the work happening on all servers. This could add considerable overhead to the geodatabase environment in particular. For lighter-weight transactional applications this might work well, but the GIS environment does not qualify in that regard. One additional consideration with any active-active solution is the additional software licenses for the DBMS and other software that may not be required for the active-passive method.

Other database systems (such as IBM Informix) have several different methods of providing failover and cluster capabilities. For instance, the current IBM Informix IDS active-active offering utilizes disk mirroring, log shipping, and read-only capabilities to synchronize the secondary platforms. Load balancing and horizontal scalability are not available in this configuration. This has also not been widely used in GIS implementations and has not been tested by ESRI and would require customer testing to validate its feasibility and functionality.

Transparent Failover (specifically, Oracle TAF) is a form of active-active clustering which allows users to continue working after a database failure with minimal perceivable impact to the user. It accomplishes this by quickly and automatically moving the user's database services to a surviving active database instance. The degree of 'transparency' varies with the type of service in progress when the failure occurs. If a database transaction was in progress, the transaction would be rolled back and the user would have to redo their updates. A read-only user would receive an error message and would have to redo a request for data. A read-only user with no database transactions in progress at the time of failure might never know that anything happened. In-progress server-side database queries theoretically would be moved over to a surviving database instance and restarted from the beginning.

This technology works best in a light-transaction environment where work arriving at the database is in small, discrete units which are quickly processed. The ArcGIS desktop software architecture utilizes complex, long-running transactions. In addition, editors and viewers establish persistent sessions and database connections which are not easily moved or recreated if the underlying database fails. As mentioned earlier, there are other technical issues that could prevent it from providing adequate performance with multiple active instances. For this reason, the RAC/TAF solution has not been implemented in the ESRI GIS environment and is not

supported by ESRI. For additional information view the technical article on this subject on the ESRI Knowledge Base at:
http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=27653

As mentioned elsewhere, these vendor technologies can change over time, so we recommend that customers check with the hardware and software vendors before making architecture decisions and validate vendor marketing materials with functional testing in the GIS environment prior to committing to a solution.

Failover Scenarios - With the most commonly used clustering solutions, such as Microsoft Cluster Service, once the cluster detects a failure in the primary database, an automated script (user created) is initiated which moves ownership of the database on the storage subsystem and the primary database IP address to the failover server. It then restarts the database, rolls back any incomplete transactions and opens for business. During the failure, the ArcGIS desktop software attempts to reconnect to the database. At the point that the database and other resources became available, the desktop software normally automatically re-establishes the connection to the database. Typical failover time for an ArcSDE database node ranges from 20 to 90 seconds. Customers are responsible for creating and testing the automated failover scripts in their environment.

Users editing information in the geodatabase at the time of failure lose their edit session and also lose any edits that had not been committed to the database at the time of failure. To recover this work, they would have to restart their edit session and recreate these changes. After a database failure, view-only ArcMap users would also lose the connection to their MXD and would receive an error message the next time they attempted to perform work. They would have to reload their MXD after the failover instance became available.

All of this information pertains to Oracle RAC as well, although the failover time for the database may be reduced from 20 to 60 seconds to 2 to 5 seconds.

ESRI Software Functionality - As noted earlier, ArcGIS desktop and ESRI ArcSDE software components are neither 'cluster aware' nor do they have failover or high-availability functions built in. Therefore, the monitoring of application functionality by the cluster can only occur at the operating system service level or through special application monitoring processes. It is also possible for a database to become non-functional yet appear to be operational to cluster monitoring software. This can happen if the determination of database health is simply the absence or presence of the database application processes. Clustering and other fault-tolerant configurations for the DBMS are dependant on operating system and/or database-specific clustering implementations provided by those vendors.

License Manager - Any high-availability solution must also take into consideration the ESRI License Manager. For ArcGIS 8.3 and below, the License Manager file must be available on a server for users to connect to the database, so generally, clients install one copy on an active server, and maintain a second inactive copy on a secondary or failover server. ESRI generally does not require additional license fees for the second copy of ArcSDE as long as customers agree to have only one active copy at a time. Check with your ESRI regional office or account manager for the details on this. Clustering software can be used to manage the License Manager failover, although this should be verified once a configuration has been selected. For failover clusters, the FlexLM service is included in the failover script and transferred to the failover platform.

With ArcGIS 9.0, the license information is stored in the database. In a clustered environment, the license manger information would be available on the secondary platform after failover.

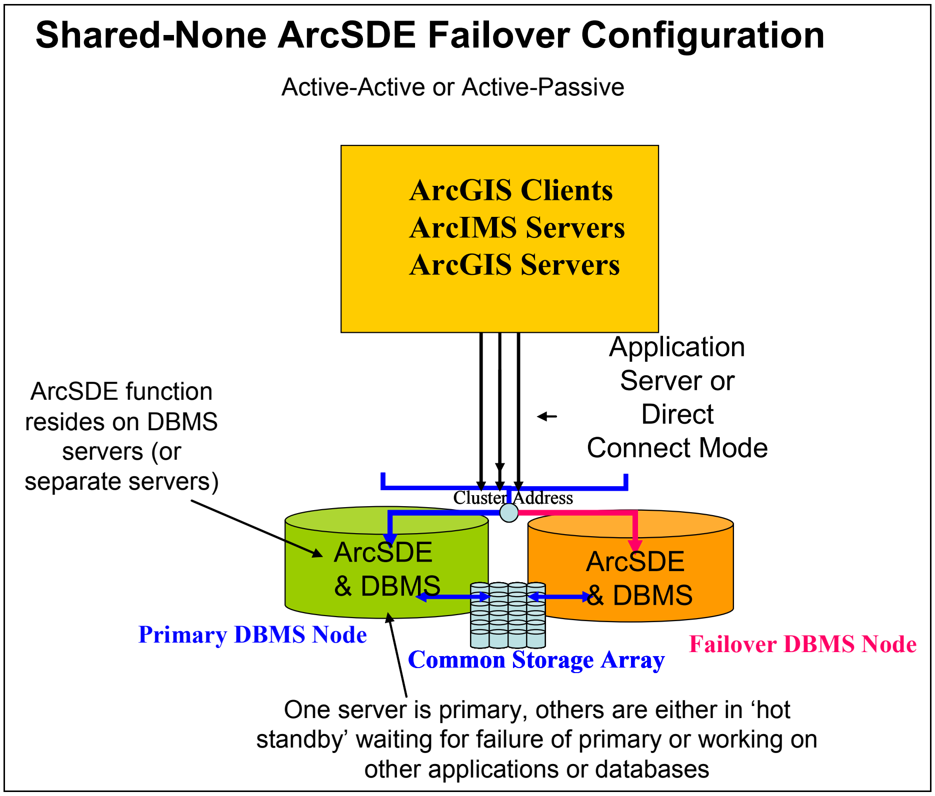
3) ArcSDE Services and Connections

ArcSDE can run as middle-ware processes on the same server that the DBMS runs on, on a separate server (both methods are referred to as Application Connect mode), or on each client workstation (Direct Connect mode) and application server (ArcIMS, ArcGIS Server, Windows Terminal Server). Configurations where ArcSDE is supported on a separate middleware platform are more difficult to maintain, add complexity to high-availability solutions and create another point of failure in the configuration. Review the Technical Brief on the Systems Integration website for additional information (www.esri.com/systemsint/kbase/docs/tieredhardware.pdf).

With shared-none clusters, either mode can be utilized to connect to the database and the end-user experience during failover will generally be identical. The client software will automatically re-establish the connection once the failover database is successfully restarted and ready for use.

If an Oracle 9i/10g database participating in an Oracle RAC is utilized, the client workstations must utilize the ArcSDE Direct Connect mode (utilizing ArcSDE functionality included in ArcGIS desktop software) for establishing connections with the ArcSDE database. Figure 1 below shows an example of a typical ‘shared-none’ configuration.

Figure 1



A Direct Connect configuration will also remove about 20 to 40 percent of the CPU load on the DBMS server compared to a combined ArcSDE/DBMS server configuration. It does place an additional load on the client workstation (or application server) and might increase overall client response time somewhat; however, most current workstations can readily support the added work without adversely affecting performance. In a site with an overburdened database server combined with relatively fast client workstations, the Direct Connect mode can be useful in improving overall response time for the user. The pros and cons of this Direct Connect mode are listed below as taken from an update to the ArcSDE Multi-tiered Configurations document mentioned above. Note that with ArcGIS 8.X this will increase network traffic and requires software to be installed on each client workstation. With ArcGIS 9.0, the network traffic penalty is significantly reduced.

Pros:

- Reduces database server CPU load by 20 to 40 percent due to off-loading of GSRVR processes, which provides an increase in database server scalability.
- Reduces memory requirements on the database server.
- Eliminates the requirement to install application middleware (ArcSDE) on the database server.

Cons:

- Increases network traffic relative to application server connect due to the ArcSDE server/client communications (reduced for ArcGIS 9.0).
- Requires database client software installation and configuration on each client PC.
- With ArcGIS 8.X, client response time will increase by roughly 10 to 30 percent, depending upon the database and data type used. With ArcGIS 9.0, this penalty is reduced to 10 percent or less.

4) ArcSDE/DBMS Access to the Network

High availability for the ArcSDE servers requires redundant network paths over shared connections (backbones, WAN, etc.). Capabilities also exist (as mentioned earlier) to configure cluster failure sites through network components. For database availability, the most critical network resources are the connections between the ArcSDE servers and the other GIS-related servers (ArcIMS, ArcGIS Server, Windows Terminal Server), and between ArcSDE and the enterprise local area network (LAN). Best practice usually involves installing two network interface cards (NICs) on each server with failover capabilities implemented through NIC card teaming. One NIC card is often used in a cluster configuration to provide the cluster a 'heart-beat' from each node, which, if missing, will trigger the failover process.

Best practice is for very high-bandwidth network connectivity for ArcSDE servers such as those provided by 100-Mbps or 1-Gbps NICs.

Application Server Considerations

ArcIMS. ArcIMS can be setup to use either the Direct Connect or Application Server modes of connection to the ArcSDE database. In either case, ArcIMS will automatically re-establish database connections once the failover database server is operational. ArcIMS itself can also be setup in a clustering environment with Microsoft Cluster Service for failover capabilities and load balancing for the application and spatial server components. Refer to Chapter 3 of the *System Design Strategies* white paper at http://www.esri.com/systemsint/kbase/strategies.html, for high-availability configurations for ArcIMS. ArcIMS protection can be implemented with network load balancing and does not require special clustering software. Review http://www.esri.com/systemsint/kbase/archive.html for a technical briefing on ArcIMS network load balancing and clustering.

ArcGIS Server. The introduction of server-based applications through such facilities as ArcGIS server requires configuration planning to accommodate high availability. ArcGIS server establishes sessions between the database and the server platform on behalf of the clients using ArcGIS server services. Performance through a cluster failover would be similar to ArcIMS for transaction-based services, and similar to ArcGIS desktop connections for ArcGIS server user workflow sessions ArcGIS servers can also be configured to provide high availability and application load balancing. Refer to Chapter 3 of the *System Design Strategies* white paper at http://www.esri.com/systemsint/kbase/strategies.html, for high-availability configurations for ArcGIS server.

Windows Terminal Server/Citrix. Windows Terminal Servers (WTS) running ArcGIS desktop software can be configured to utilize either Direct Connect or Application Server modes. Since WTS servers are usually co-located with the database servers on high-bandwidth LANs, the network penalty for Direct Connect is of little consequence. Using Direct Connect will add some additional load to these servers, but the users will likely experience little performance degradation. For additional information on configuring high-availability and load-balancing configurations for WTS and Citrix refer to the vendor documentation.

Summary

The choice of technology to provide failover capabilities will vary depending on the specific hardware, operating system and database system environment, and on the type of failover required, if any. Careful research in each of these areas should be undertaken prior to committing to a particular solution. The best solutions (lowest risk) are those that are already proven in the ESRI geodatabase environment. New technology should be thoroughly tested prior to the completion of a system design to insure that the desired features will work in an ArcSDE environment. The shared-none offerings that have been on the market for some time currently provide the least amount of risk in implementing high-availability solutions. This could change over time as more customers get experience in using the newer offerings and as vendors and ESRI work to improve the functionality in a geodatabase setting.

High-availability ArcSDE configurations include the following elements:

- The DBMS servers should incorporate fault-tolerant features such as redundant power supplies, NICS, RAID storage.
- The DBMS server instances should be part of a server cluster with at least two servers.
- The DBMS servers should be connected to high-speed and redundant network paths.
- Each cluster server must have the same DBMS software installed.
- A common storage subsystem for database table storage must be available to each node of the cluster.
- Each cluster server should be connected to the storage subsystem with redundant high-speed paths (i.e., dual-host bus adapters).
- When clustering, users must provide failover scripts that automate the transfer of resources to and re-starting of processes on the secondary platform.

Additional Resources for High-Availability Planning

- www.microsoft.com/ntserver/productinfo/enterprise/ - For information on Microsoft Cluster Service (MSCS), MS SQL Server, network load balancing and other Windows-based high-availability technologies.
- h18000.www1.hp.com/solutions/enterprise/highavailability/oracle/rac-overview.html – For a good discussion of HP/Oracle RAC architecture and issues.

- http://otn.oracle.com/deploy/availability/pdf/MAAoverview.pdf – for Oracle availability information.
- www-306.ibm.com/software/data/informix/pubs/whitepapers/HA_f.pdf – A discussion on IBM Informix IDS high-availability options.
- www-106.ibm.com/developerworks/db2/library/techarticle/0304wright/0304wright.html – An overview of IBM DB2 Availability and Disaster Recovery.
- ESRI Test Report – MS SQL Server 2000 MSCS Configuration
- www.esri.com/systemsint/kbase/docs/SQL_Server_2000_MSCS_Configuration.pdf