ExaStore architecture

WHITE PAPER

July 2005

EXANET

STORAGE ACCESS - NO LIMITS

1. Introduction

ExaStore innovative grid NAS solution delivers unlimited scalability, high availability, record-breaking performance, and superior automation.

The ExaStore solution is a technology independent approach, consisting of Exanet software and standard off-the-shelf hardware. The ExaStore system solves business and technology problems at all levels of the organization while utilizing best of breed software and hardware. Highlights include:

- Single global file name space greatly simplifies information sharing and system administrations.
- Fully distributed file system enables flexibility and agility.
- Massive scalability supports growth in both capacity and bandwidth independently, enabling high computing environments and performance demanding applications.
- Fully clustered solutions allow for high availability, reliability, and reduces downtime thus keeping up with the needs of business.
- Innovative architecture provides ultimate performance under all access patterns.
- Industry-standard hardware results in lower acquisition and maintenance costs in addition to investment protection for the future.
- Multi protocol compatibility allows administrators to consolidate many individual file servers into a single entity, which is accessible via all major network-file protocols, such as NFS, CIFS and AFP.

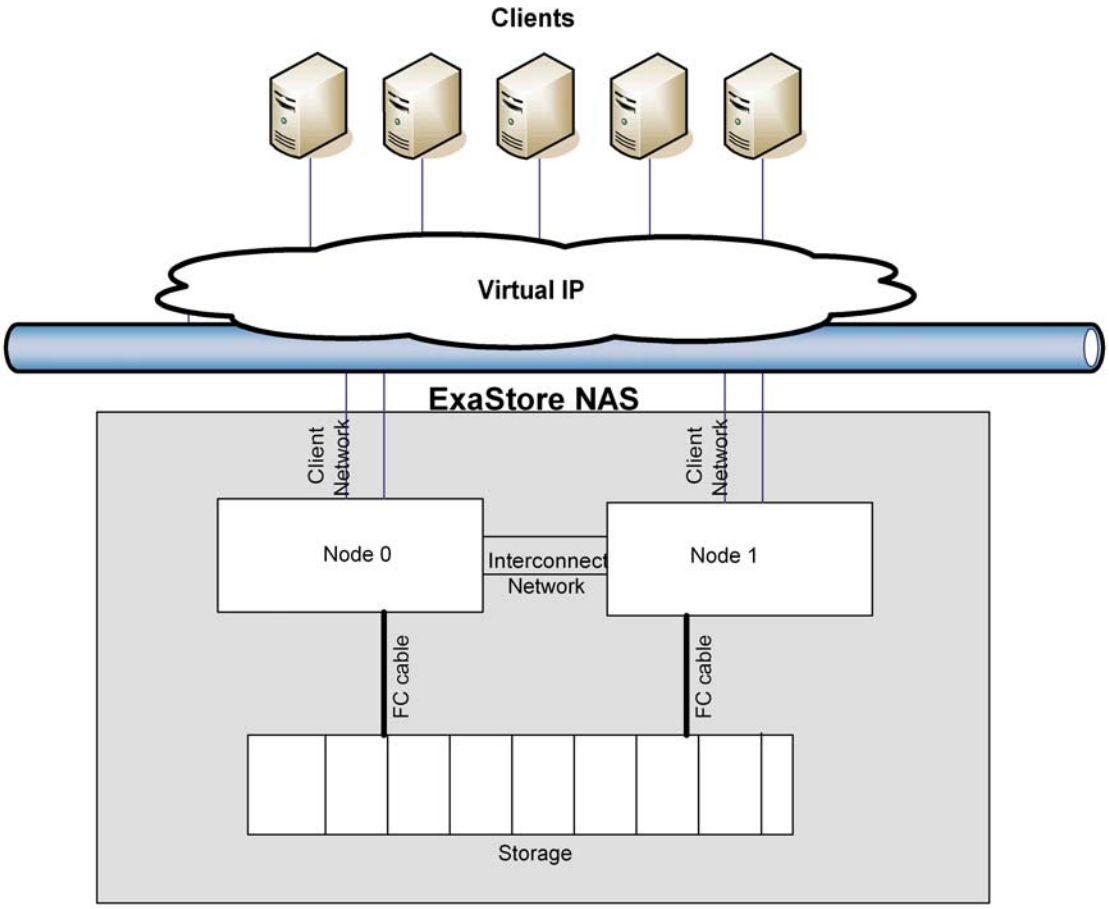
The Exanet solution is future proof, providing for business needs and the ever changing technology landscape.

To understand how the ExaStore solution helps administrators optimize their storage environment, one must understand ExaStore's architecture and how it delivers unlimited scalability and high performance. This document is a technical overview of ExaStore's architecture and components, distributed file system, and clustering technology. This document is intended for a technical audience, such as IT and storage administrators, who want to understand ExaStore architecture and components in greater detail.

2. ExaStore Architecture

The system architecture picture illustrates an ExaStore two node configuration.

Figure 1: ExaStore Architecture



2.0.1 The Client's View

- From the client's point of view, the ExaStore NAS solution is presented as a single server with a single file system, IP address, and name, regardless of the number of nodes and storage subsystems. Unlike other scale-limited NAS and SAN products, the ExaStore global file system serves all users concurrently, without performance constraints. End users connect to the ExaStore storage system utilizing their respective operating systems NAS protocols:
  UNIX users access ExaStore via the NFS protocol.
- Windows users connect to ExaStore as any standard Windows server using CIFS protocol.
- Apple users mount files through the AFP protocol - AppleShare IP method.

Once the user establishes the preliminary connection, ExaStore acts as a regular storage subsystem, accessed normally by end users utilizing any application.

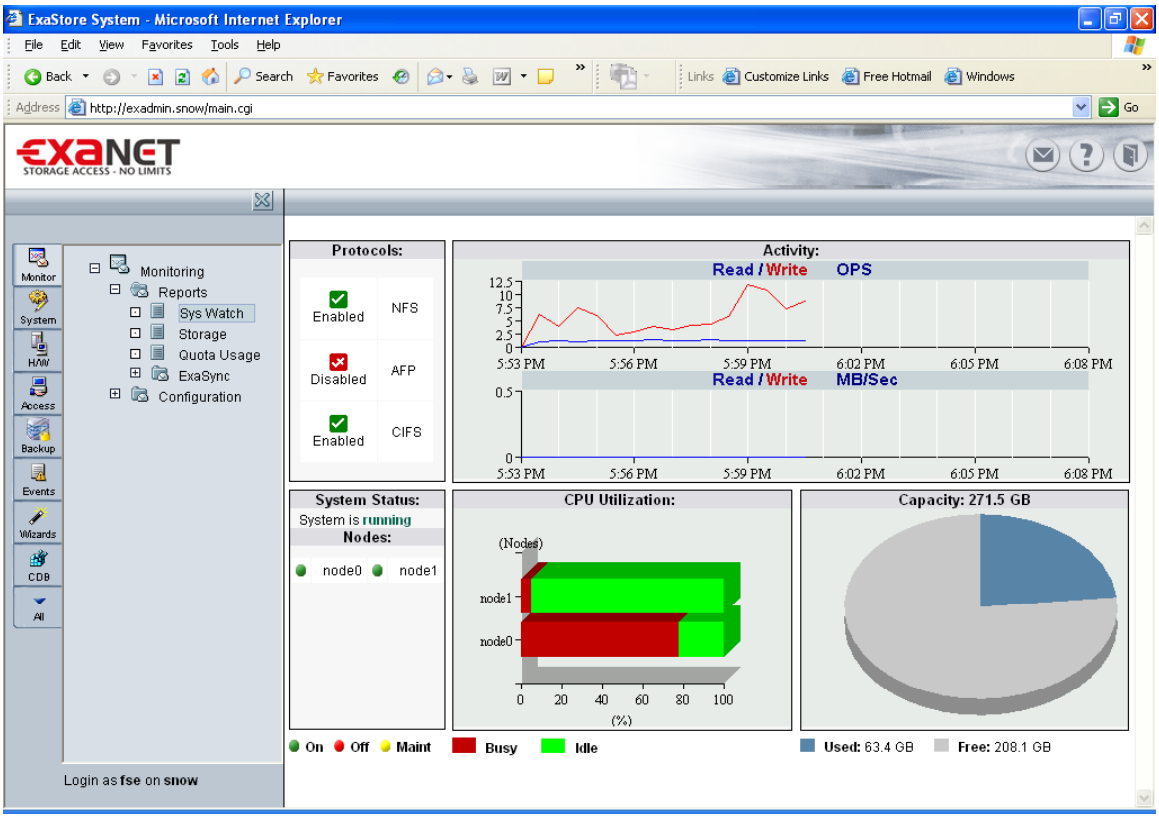
2.0.2 Administrator's View

Administrators can use either the command line interface (CLI) or the Web interface (WUI) to configure ExaStore services.

- CLI is a simple command line interface that covers all management functions
- WUI provides access to system functionality via all standard browsers.
- SNMP Interface supplies any Network Management System (NMS), such as HP Open View, Tivoli and CA Unicenter, with access to the system using MIB II and the ExaStore MIB.

Administrators change system settings, such as configuring protocols, adding users, and setting permissions, through the WUI or CLI. Any Network Operation Center (NOC), which supports the SNMP interface, can be used to provide system management and monitoring. Most maintenance procedures are performed automatically.

Figure 2 – ExaStore web interface

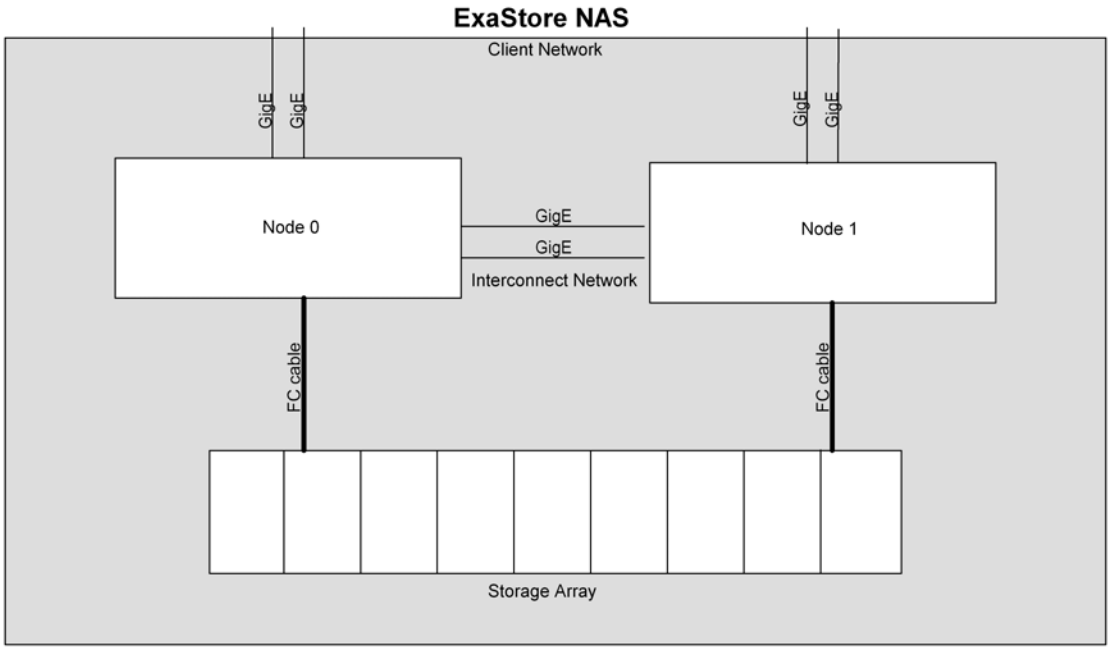


2.1 ExaStore Solution Components

ExaStore aggregates and effectively utilizes all components in the systems to provide a complete grid NAS solution with maximum performance, scalability, and high availability.

The picture below illustrates ExaStore's two node system configuration, where both nodes are attached to shared storage through fiber channel (FC) connections. A different UPS backs up each of the nodes. Two back-to-back GigE connections comprise the private interconnect network. The public network consists of two GigE interfaces per node, which consolidate into one virtual IP.

Figure 3: ExaStore solution - two node configuration



2.1.1 Nodes

Description:

Simply, nodes are servers in the ExaStore cluster. Nodes handle all read and write disk operations, perform caching, and interface with servers and workstations. The smallest configuration consists of two nodes in a cluster. This ensures that there will not be a single point of failure. ExaStore's flexible architecture enables system expansion by node pairs, as and when needed.

Function:

ExaStore software is installed on each of the nodes. The software is a complete package, consisting of ExaStore operating system, volume management, distributed file system, and clustering technology. A virtual IP consolidates all the system's client network interface cards (NICs). ExaStore's built-in traffic load balancing mechanism redirects clients request to the least loaded server, maintaining an even balance between all nodes in the system.

Operations are handled through mirrored non-volatile RAM (NVRAM). Each pair of nodes performs cache mirroring. This assures a quick response to clients' requests while maintaining complete data integrity. Data from the cache to permanent storage is transferred asynchronously, according to a variety of optimized data-placement schemes.

2.1.2 Cache

Description:

Each server is equipped with a minimum of 4GB of RAM. It is possible to increase this, depending on customer requirements and server technology. Most of the RAM is used for storage caching. Since a UPS backs up each node, the cache is used by the file system as NVRAM and boosts performance levels.

Function:

The file system uses the cache efficiently to provide fast and reliable writes and reads.

Writing or modifying files occurs first in the cache. Then data is mirrored to its peer node cache. This ensures that all transactions are duplicated and secured.

After the cache is mirrored, the system acknowledges the action to the clients, who continue with their work. The cache combines all new and recent writes and then distributes and writes the data evenly across all available LUNs in the system.

When requested to read a file, ExaStore retrieves the selected data from cache. ExaStore only accesses the disk if the file is unavailable in cache. The file loads onto the cache the first time it is requested. All subsequent read requests for the same file are serviced from the cache.

2.1.3 UPS

Description:

The Uninterruptible Power Supply, or UPS, provides continuous power to the nodes.

Function:

Each node is connected to a UPS. The UPS enables ExaStore to use the cache as NVRAM. This backs up the system in case of a node power down.

The management interface monitors the UPS in order to facilitate an orderly system shutdown in the case of imminent power loss.

2.1.4 Storage

Description:

Each node connects to a storage array which is a Redundant Array of Independent Disks (RAID) subsystem.

The physical disks in the RAID subsystem are divided and configured as LUNs (Logical Unit Number). A LUN is a group of physical disks configured in any RAID technology and is the smallest logical disk unit ExaStore handles.

RAID storage subsystems are designed with no single point of failure. Each active component in the storage subsystem is redundant and hot swappable. ExaStore supports the typical RAID configurations of striping (RAID 0), mirroring (RAID 1), striping and mirroring (RAID 0+1), and striping with parity (RAID 5). RAID subsystem can support FC or SATA disk technologies.

Function:

Each node controls one or more storage LUNs. In a normal situation, when both nodes are available, each of the nodes controls half of the total number of LUNs. If one of the nodes fails, its pair node takes ownership of the LUNs. This ensures that all data is available to clients continuously.

ExaStore's unique distributed file system automatically aggregates all LUNs regardless of physical location, to a single file system and name space which appears as one large disk.

2.1.5 Fibre Channel (FC) connections

Description:

The FC connection provides the nodes access to the storage subsystem either directly or through FC switches. Storage arrays provide dual access; ExaStore software utilizes this for redundancy.

Function:

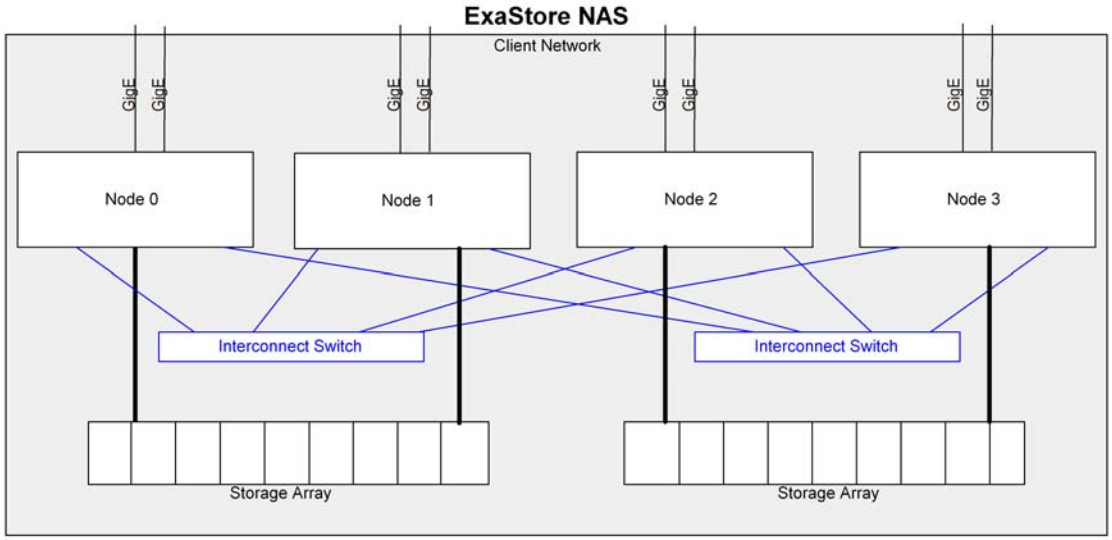
With ExaStore cluster architecture, connecting nodes and arrays through FC switches is unnecessary, as is use of more than one connection per node. ExaStore software via interconnect switches regulates access to other LUNs in the system, controlled by other nodes. If one of the nodes fails, the other node takes control of the LUNs, using its own FC path. This second node then presents the data to the file system.

2.1.6 Interconnect Network

Description:

ExaStore's interconnect network is comprised of two independent GigE networks. The interconnect network acts as ExaStore's heartbeat mechanism and an internal data transfer between nodes. In a two node system, no switches are used. In configurations with more than two nodes, the interconnect network includes two GigE switches. All ExaStore nodes are connected to both interconnect switches. These employ the dual links for redundancy and load balancing.

Figure 4 – ExaStore Interconnect network in four node configuration



Function:

In order to achieve complete data distribution and to maintain high availability, each of the nodes in the ExaStore cluster system must have access to all other nodes in the system. The interconnect network achieves this goal.

The interconnect network is the infrastructure for ExaStore clustering, including the heartbeat monitor, transferring data, mirroring information between the nodes' caches, and distributing data evenly across all LUNs in the system.

2.1.7 Client Network

Description:

Public network is comprised of 2x GigE NICs from each node, which connect to public network switches. ExaStore supports either fiber or copper switches, according to client request. Exanet recommends that switches and client machines support jumbo frames for better performance.

Function:

All client NICs aggregate as a single Virtual IP (VIP) for clients' traffic. Additionally, ExaStore supports a single VIP per protocol (NFS, CIFS, and AFP), user interface VIP, and Backup VIP. This provides flexibility to customer networking requirements and needs.

2.1.8 Management Network

Description:

The ExaStore management network connects the nodes, UPS, and storage array through switches to a private management network.

Function:

The management network in ExaStore achieves several important functions:

- Communication notification between nodes and UPS:
  If a power failure occurs, the UPS notifies the node in order to properly dump the cache to disk and shut down.

- Array management:
  This is designed for arrays that support out-of-band management. ExaStore integrates with various array functions, including LUN expansions and additions. This involves monitoring and reporting via ExaStore's user interface.
- Automatic reboot and shutdown of nodes:
  In case of a suspected split brain situation, where the nodes in a cluster lose communication with each other, the management network is used to automatically reset suspected nodes. This prevents a split brain situation while ensuring data integrity.
- System boot:
  In configurations comprised of four or more nodes, only two of the nodes hold the system image on local disks. The rest of the nodes boot through the management network.

2.2 Scaling in ExaStore

ExaStore is built to supply “on demand” growth. Whenever the need arises, ExaStore can expand in both storage capacity and bandwidth independently.

ExaStore scales easily; storage capacity increases through the addition of new storage arrays or LUNs. ExaStore incorporates these devices while the system remains online and operational. Scaling bandwidth is achieved through the addition of nodes.

ExaStore discovers new storage subsystems or nodes automatically. Once the capacity in a system increases, ExaStore’s load balancing mechanism distributes the system’s content among the new and old storage subsystems. By doing this, ExaStore ensures that data remains balanced across multiple disks, and therefore eliminates write bottlenecks. This operation is performed during system idle time and can be tuned by the administrator.

When the customer needs more bandwidth, the administrator can add additional nodes to the system. ExaStore auto-configures the new nodes as the system runs; this operation does not require user downtime. Once the nodes are added, ExaStore automatically expands its bandwidth and increases the number of new node interfaces to the same client's VIP.

3. ExaFS: ExaStore Distributed File System Overview

ExaStore core software is a unique distributed file system. From a client's perspective, ExaStore acts as a single large NAS solution, exporting one file system with one root directory and tree.

Each node in ExaStore contains four sub file system daemons, or FSDs. Clients use FSDs to open sessions and interact with ExaStore's file system. Each FSD consumes one fourth of the node's CPU and RAM. It uses these resources to provide services to the FSD's attached clients.

A FSD carries its own metadata file only. This information points to the physical location of the files, directories, and metadata. A FSD accomplishes this while fully sharing storage with all other FSDs.

Data and content reside on any available LUNs in the system, regardless of their physical location and owner node.

As the client creates and modifies files, free blocks from available storage can be allocated to any FSD. As a client deletes data, the system frees the storage blocks, returning them to ExaStore's overall storage capacity.

Each of the FSDs sees and accesses the entire file system. Through the FSDs, the whole system becomes one very large disk. Thus the customer is able to access and utilize all the used and available storage efficiently.

3.0.1 File Access Scenario

The following scenario describes how ExaStore handles an I/O request:

A client connects to the system. ExaStore uses its client load balancing mechanism to assign the client's connection to the appropriate node and FSD.

The client requests an operation. The FSD on the connected node is responsible for servicing the request until the client receives a response. Typical scenarios are:

- **Read Request:**
  When a client wants to read a file, the FSD checks its own metadata file first to identify the file's location. If the FSD possesses the requested file, the FSD fetches the needed data from the cache. If data is not available in cache, the FSD retrieves the file from disks instead.

  If the FSD processing the read request is not the file's owner, the FSD communicates with the appropriate FSD, through the interconnect switch. The Owner FSD fetches the file from the correct cache or disk and transfers the information to the FSD requesting the file read.

- **Write Request:**
  The client creates a session with one of the FSDs. The client's new FSD becomes the owner of all of the files and directories that the client generates in that session. This means that this particular FSD contains the metadata for these objects.

  The client writes data to FSD's cache. ExaStore mirrors this information across to FSD's peer cache. Then the client receives acknowledgement for the write and continues working. In the background, the cache collects data and writes the information to disks while ensuring even data distribution to all available LUNs. This automated capacity-balancing feature promises equal disk utilization and reduces the risk of disks bottlenecking in the system.

3.0.2 File System Operations in Degraded Mode

When one of the major components fails, the system operates in degraded mode. The system allows full access to all information while maintaining data safety and integrity. If one node fails, a peer node takes over all its FSD domains. The active node then holds eight FSDs and continues providing services to clients.

Data in an uncommitted cache is written to disk. In degraded mode, ExaStore changes from a cache-mirror method to a journaling approach. This ensures that all write transactions are recorded to stable disk before the system gives the client acknowledgment. When a peer node becomes available, normal mode operation resumes.

Through a UPS notification, the node senses power failure. Immediately, the node dumps all of the write-cache data and metadata to a local disk. This operation is internal and does not depend on the availability of any external components. When power returns, the node restarts. ExaStore restores the uncommitted data and mirrors in the peer node. This reinstates normal operation.

4. ExaStore Clustering Technology Overview

ExaStore architecture is inherently highly available. ExaStore ensures seamless access to data regardless of requests for particular data or demand on the system overall. Redundant pairs of servers with transparent failover protect file access, and RAID technology safeguards the disk's data.

ExaStore creates multiple network paths to each server, which shields against network failures. The cache is duplicated to prevent data loss. ExaStore's Active/Active architecture utilizes otherwise idle resources to improve performance.

ExaStore is a highly available system, designed to remain running 99.999% of the time. ExaStore creates a high availability system by using the following features:

- **No Single Point of Failure**: All critical system components, both hardware and software, are redundant.
- **Automatic Recovery**: When either a hardware or software component fails, ExaStore recovers automatically. This eliminates the need for immediate human intervention to restore service.
- **Maintainability**: All maintenance procedures permit system upkeep and upgrade while the system provides service. In most cases, end users do not experience service degradation during maintenance operations.
- **Self Healing Mechanism:** ExaStore cluster possesses a self healing mechanism which enables each node to monitor its peer. If a node recognizes a service failure, it tries to restart the node before initiating failover.

ExaStore's basic configuration consists of two nodes to enable high availability. In normal mode, each of the nodes is active and supplies services to clients. When a failure occurs, the peer node takes ownership of the failed node's services and continues to provide all regular services to the client.

4.0.1 User Experience in failover

When a failure occurs, a client's experience depends on the protocol used to access the system.

Clients connected to the failed node with CIFS or AFP experience session termination. When retrying to connect, the remaining active node generates a new session. Clients using NFS automatically reconnect to the active node without losing the current session.

4.0.2 Handling Failover Scenarios

ExaStore recovers automatically from both hardware component and internal software service failures. ExaStore maintains service while the failed component is replaced. After the fault is repaired, fully redundant service resumes. The following describe ExaStore's maintenance capabilities.

Service Availability

There are two different general service models. The first is called Active/Active. In this method, all nodes provide service simultaneously. When a component fails, it restarts and continues to provide service. The second service model is called Active/Stand-By. In this architecture, a single source provides service; it only moves between the nodes if problems occur.

ExaStore usually exercises the Active/Active architecture. ExaStore is designed to utilize all of the system's available resources. When a failure occurs, it isolates only the failed component, keeping the remainder active and running.

Handling Hardware Failures

ExaStore can handle all hardware failures. Examples of major hardware failures include the following:

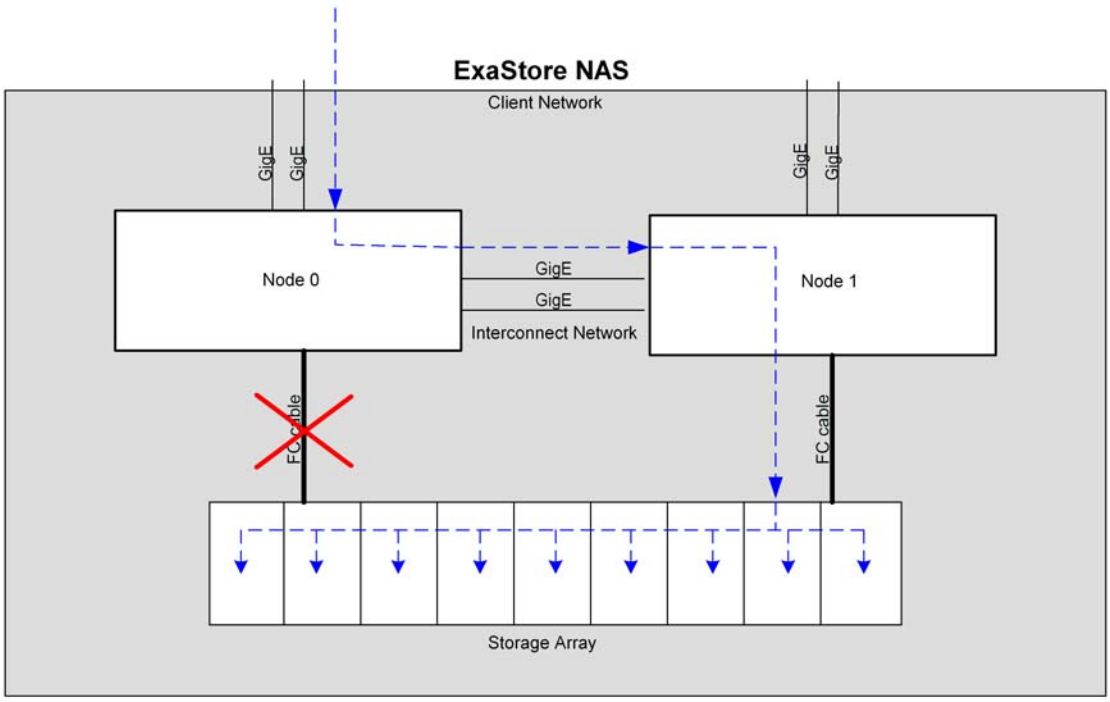
Subsystem	Component	Comment
RAID	Fan (Power supply) Disk RAID Controller	Replaced without impact on service. Hot spare and hot swap allows replacement of faulty disk. Active-Active. Replace faulty controller.
Node	Motherboard, RAM, NIC, Host Bus Adapter (HBA)	The node is down. The remaining node provides service in non-redundant mode.
Node	Power supply (Fan)	Replaced without impact to service.
Interconnect switch	Faulty switch	The remaining switch runs in non-redundant mode.
RAID	Fan (Power supply) Disk RAID Controller	Replaced without impact on service. Hot spare and hot swap allows replacement of faulty disk. Active-Active. Replace faulty controller.

In the examples of hardware failures above, ExaStore's software is able to detect the faulty component and use its scheme to resume service as fast as possible. When a faulty component is replaced, ExaStore detects the change and returns to its normal mode, as defined by its current availability configuration. For example, if a FC connection from one of the nodes fails, the other node takes control of its peer LUNs.

Both nodes continue to receive client requests. Meanwhile, the node that lost its LUNs accesses the LUNs physically attached to its peer node through the interconnect network.

The system utilizes nodes, CPU, and RAM to produce the maximum response for ExaStore clients.

Figure 5 – ExaStore cluster, FC connection failure



5. Summary

Exanet designed the ExaStore grid NAS architecture for maximum reliability, scalability, and performance.

- ExaStore's architecture's ability to scale easily allows storage administrators to buy and implement what they require today, knowing that they can grow as their organization needs tomorrow. All the while, ExaStore maintains and utilizes the same solution, greatly reducing hardware and maintenance costs.
- ExaStore's flexibility encourages evolving, high performance environments. It can be changed modularly to support specific applications' bandwidth requirements. If bandwidth needs increase, ExaStore scales the number of nodes to support new bandwidth demands easily and immediately.
- ExaStore's highly available architecture ensures data availability and reduces both planned an unplanned downtime.
- ExaStore's unique distributed file system provides a single global file name space. This greatly simplifies information sharing and system administration, thus enabling maximum flexibility and agility.