High Availability

Achieving Highly Available and Manageable File Serving with NAS Clusters

Mark Mills

Hewlett-Packard Company

mark_mills@hp.com

HP WORLD 2002 Conference & Expo

HA File Serving with NAS Clusters

Why is it important?

- Nature of some data dictates that it be “always on” and “always available”
- Gradual shift from predominantly block based access to file based access
- Cost of downtime > cost of high availability
- Shift in business to 24x7 and continuing increase in e-commerce
- Downtime results in:
  - loss of revenue
  - loss of productivity
  - lost transactions
  - inaccessibility of data for decision making
  - data integrity issues

HA File Serving with NAS Clusters

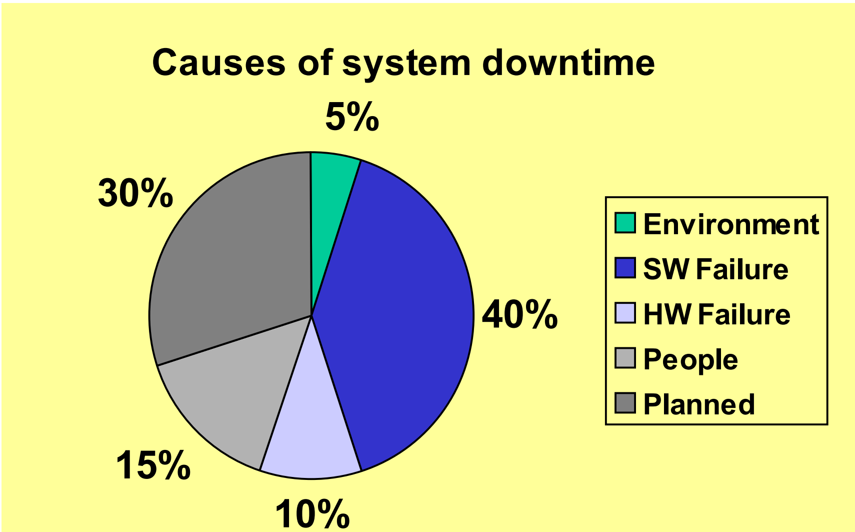
The Cost of System Downtime

Industry & Data Application	Cost per downtime hour
Finance: ATM transaction fee’s	$12K - $17K
Transportation: Package Shipping	$24K - $32K
Media: Ticket Sales	$56K - $82K
Transportation: Airline Reservations	$67K - $112K
Communications: Internet Service Providers	$60K - $120K
Retail: Home Shopping & Catalog Sales	$87K - $140K
Media: Pay Per View	$67K - $233K
Finance: Credit Card Sales Authorization	$2.2 Million - $3.1 Million
Finance: Brokerage Operations	$5.6 Million - $6.45 Million

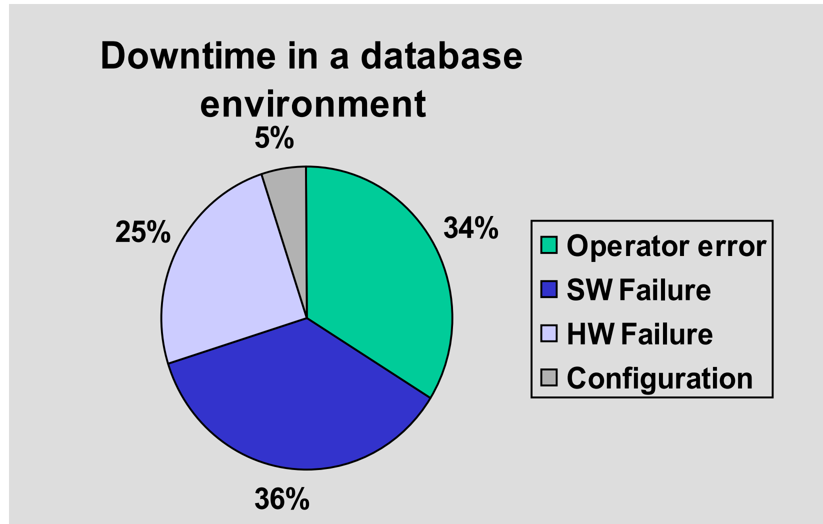
Source: Dataquest, Perspective, Sept. 30, 1996

HA File Serving with NAS Clusters

Sources of System Downtime



Source: IEEE Computer April 1995



Source: Oracle8 Backup and Recovery Manual ISBN 3-446-19459-2

- Software is the leading cause of failure
- Operator error is a major contributor
- Planned downtime is significant and can't be avoided
- Some studies have shown typical ratio (therefore cost) of planned vs. unplanned downtime is 4 to 1.
- HW and complete system failures are rare

HA File Serving with NAS Clusters

Impact of MTBF and MTTR on Downtime

MTBF = Mean Time Before Failure
MTTR = Mean Time To Repair

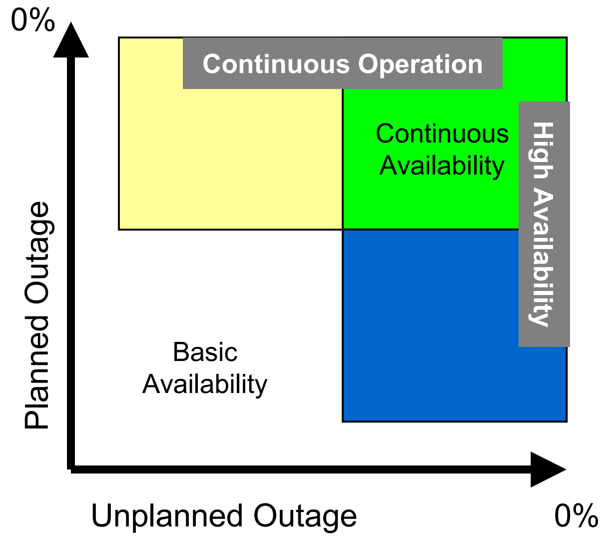
- Once a HW/SW failure has occurred, MTTR is the leading contributor to total downtime
- Maintenance occurs frequently but has a low MTTR
- System failures are rare, but have high MTTR
- Application failures occur as often as system failures but have low MTTR and don't contribute much to total downtime
- On NT clusters, only 4% of the failures were system failures, but their high MTTR made them the highest contributors to total downtime (32%).

Source: Jun Xu, Zbigniew Kalbarczyk and Ravishankar K. Iyer. Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, Hong Kong, China, Dec. 1999

HA File Serving with NAS Clusters

High Availability – How high is "High"?

Availability	Total Accumulated Outage (per Year)	Class (9's)
90 %	More than a month	1
99 %	Just under 4 days	2
99.9%	Just under 9 hours	3
99.99%	About an hour	4
99.999%	About 5 minutes	5
99.9999%	About 30 seconds	6
99.99999%	About 3 seconds	7



Development Costs - Rule of thumb:
System development costs increase by 5x to 10x for each line down the chart
(Source: Marcus & Stern, Blueprints for High Availability, 2000)

HA File Serving with NAS Clusters

How is High Availability Measured?

MTBF = Mean Time Before Failure
MTTR = Mean Time To Repair
A = Availability expressed as a percent
AFR = Anualized Failure Rate

Availability is calculated as: A = MTBF / (MTBF + MTTR)

The net fail-over cluster availability (A_c - redundant components that failover for each other) is calculated by multiplying the downtime of the primary node by the uptime (availability) of the failover node and adding that result to the availability of the node.

$A_c = A_0 + ((1-A_0) * A_1)$

The annualized failure rate of a component or system can be used to estimate availability by multiplying the AFR by the MTTR (in hours), dividing that result by the number of hours in a year (8760) and subtracting that value from 1 (100%).

$A = 1 – ((AFR * MTTR_{hrs}) / (8760))$

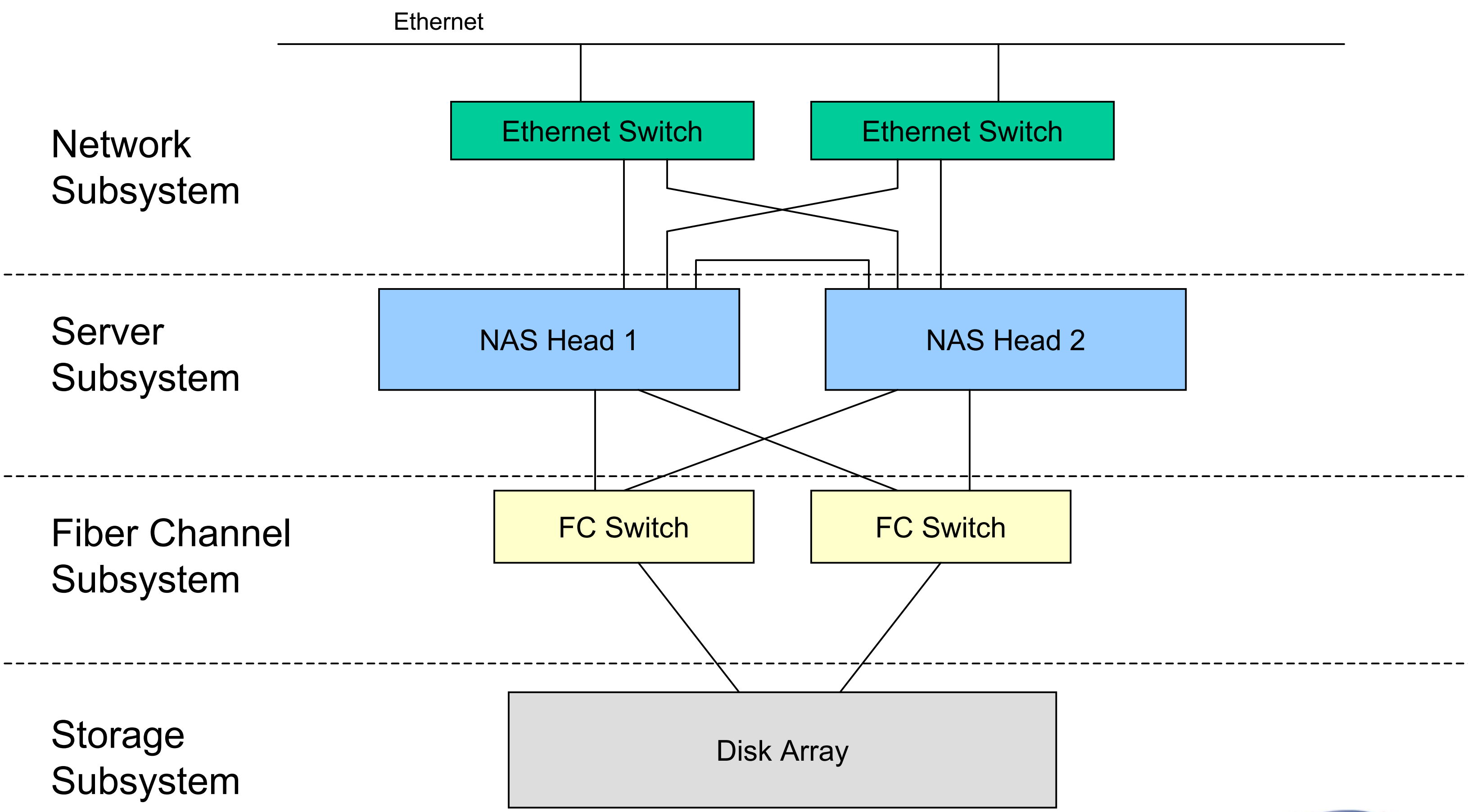
HA File Serving with NAS Clusters

Measuring Availability of a System

• Composite availability of a system is a function of the availability of it's components
• For a NAS cluster, calculating availability of each component is impractical. Instead we can divide the cluster into major subsystems for which availability can be calculated
• Natural boundaries exist that prevent subsystems from being capable of affecting the availability of neighboring subsystems
• In effect the NAS cluster is comprised of a chain of subsystems that have individual availability characteristics
• The overall availability of the NAS cluster is no better than the least-available subsystem in the chain

HA File Serving with NAS Clusters

Measuring Availability - NAS Cluster Availability Zone's



HA File Serving with NAS Clusters

Measuring NAS Cluster Availability - Example

Network Subsystem	MTBF = 60000 hours MTTR = 1 hour AFR = 2.5%	A = 60,000 / (60,001) = 99.99% A = 1 – ((0.025 * 1)/8760) = 99.999%
Server Subsystem	Single Node AFR = 53% MTTR = 4 hours	A = 1 – ((0.53 * 4)/8760) = 99.9% A_c = A + (NodeUptime * NodeDowntime) = 0.999 + (0.999 * 0.001) = 99.9999% * Adjusted for failover time = 99.999%
Fiber Channel Subsystem	AFR = 4.63% MTTR = 1 hour	Single Switch Availability is: A = 1 – ((0.0463 * 1)/8760) = 99.999% Dual Switch Availability is: A_c = A + (0.99999 * 0.00001) = 99.9999% (six 9's)
Storage Subsystem	AFR = 45% MTTR = 3.5 hours	A = 1 – ((0.45 * 3.5)/8760) = 99.98%
Overall Availability		Least available subsystem = 99.98% Availability = between 3 and 4 "nine's"

HA File Serving with NAS Clusters

How can we accomplish HA with NAS?

1) ~~Develop extremely reliable custom components and systems~~

~~**Advantages:**~~
~~**- Low failure rate**~~

~~**Disadvantages:**~~
~~**- Much higher cost**~~
~~**- Longer development time**~~
~~**- Can't use commodity HW/SW**~~

2) Eliminate SPOF with redundant components, paths & systems

Advantages:
- Can use commodity components
- Faster development time

Disadvantages:
- Higher failure rate

HA File Serving with NAS Clusters

Mission Objective #1:

Detect and Eliminate all Single Points of Failure

SPOF

Some common NAS SPOF's:

SPOF	Remedy
Power supply	Redundant hot swappable supplies, NVRAM, IPMI
Cooling fan	Redundant hot swappable fans & IPMI
Hard disk(s)	RAID, hot swappable disks and shuttles, hot spot detection
RAID controller	Dual channel with NVRAM
HBA	Dual channel or dual HBA's, hot swap PCI slots
RAM	Self correcting ECC RAM
NIC	Redundant NIC's, hot swappable PCI slots
Network OS	Heartbeat, Watchdog timer with auto-reboot capability
Network path	Multiple subnets, switches and NIC's
File System	Journaling File System
NAS Head	Failover Clustering

HA File Serving with NAS Clusters

Mission Objective #2:

Ensure Data Integrity

- Prevent simultaneous (write) access to the same data
- Prevent so called “split brain syndrome”
- Ensure that data is not lost during failure and failover

Mission Objective #3:

Invest in failure isolation and recovery

- Prevent problems in one area from affecting another
- Create well defined interfaces between subsystems and components
- Implement effective monitors to quickly and accurately detect failures
- Create failure recovery mechanisms – use system failover as a last resort

HA File Serving with NAS Clusters

Clustering Technology – The Holy Grail of HA?

What is clustering?

- connecting multiple systems together in order to provide improved aggregate availability, performance, scalability or a combination thereof.

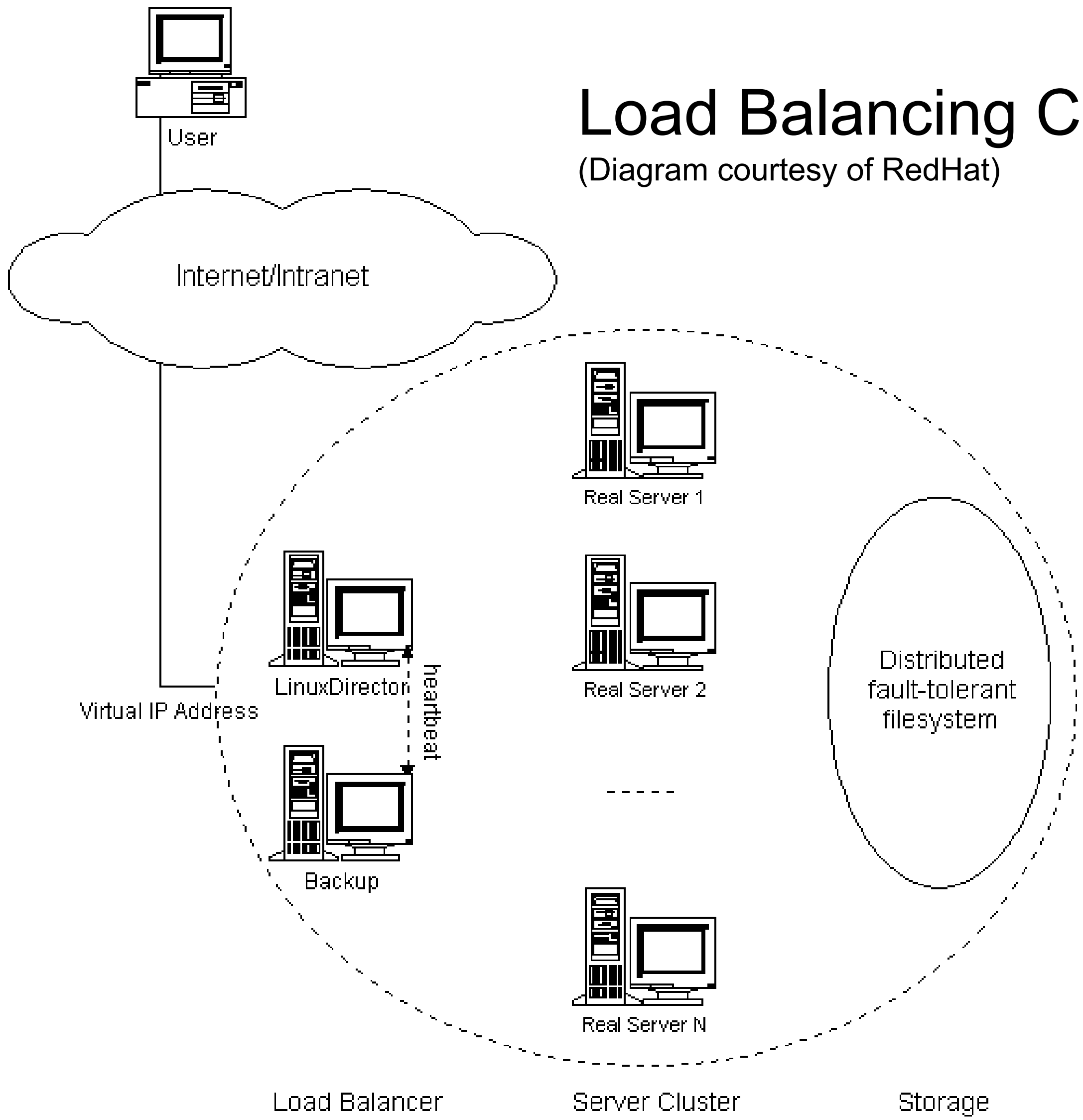
Types of clusters:

Cluster Type	Attributes	Examples
Load Balancing	- Based on inherent trust between nodes - Suitable for ensuring availability of servers of static content - Provides IP services (HTTP, FTP, SMTP, etc.) - Ensures availability by distributing IP requests across multiple cloned systems - No provisions for write synchronization	LVS RedHat HA Server Piranha
Failover	- Based on paranoid distrust among nodes - Suitable for stateful transactional app's (database, file servers, web app servers, etc.) - Ensures availability through failover mechanism	HP MC/SG SGI Failsafe SteelEye LifeKeeper
Parallel	- Distributes computational operations across nodes - Used in technical applications almost exclusively	Beowulf SGI ACE

HA File Serving with NAS Clusters

Load Balancing Cluster

(Diagram courtesy of RedHat)



HA File Serving with NAS Clusters
Failover Clusters

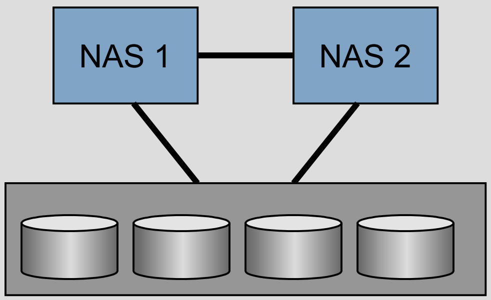
High Level Design Criteria:

Feature	Options	Current "Sweet Spot"	Future "Sweet Spot"
Cluster size	2-way	✓	
	N-way		✓
Node utilization	Active/Passive		
	Active/Active	✓	✓
Resource access	Shared nothing	✓	
	Shared everything		✓
	Mirroring		✓

HA File Serving with NAS Clusters

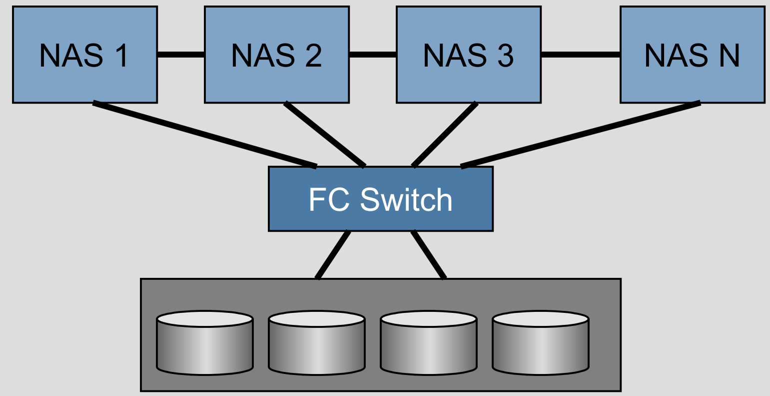
Failover Cluster Design Criteria: Cluster Size

2-Way Cluster



- Cluster consists of only 2 nodes
- 80% of current cluster installations
- Simplified quorum management
- Simplified heartbeat protocol

N-Way Cluster

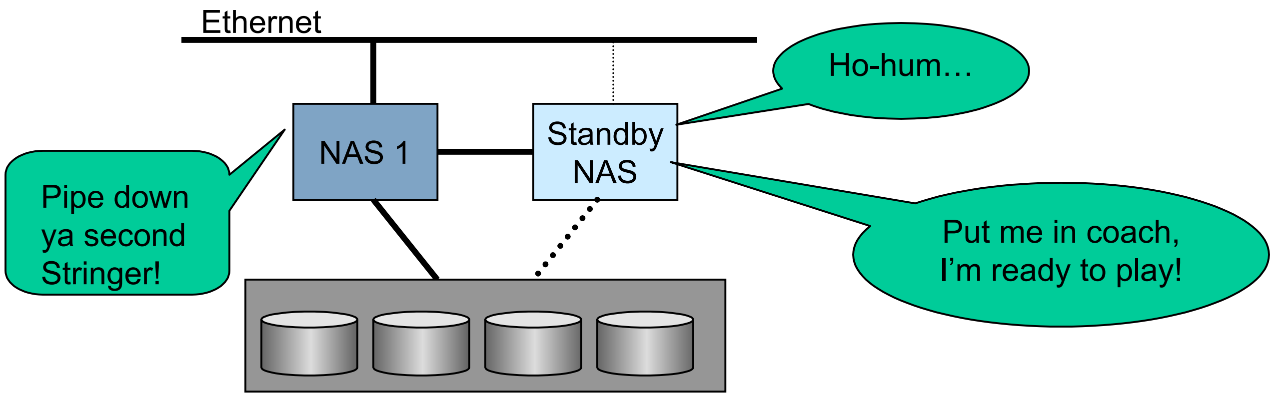


- Clusters consists of 2 to N nodes
- More complex quorum management
- Need broadcast & ring heartbeats
- Higher scalability and load balancing potential
- This is the future direction for clustering

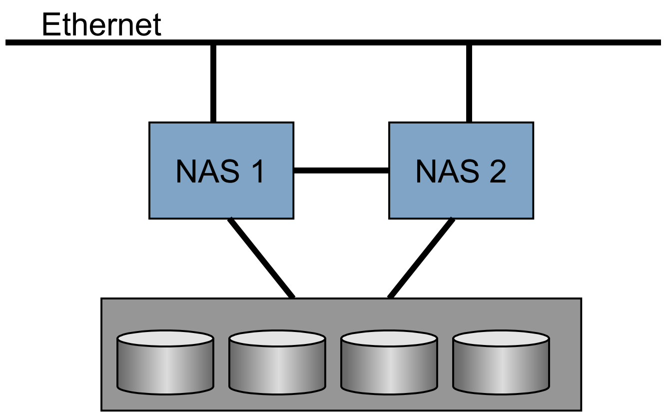
HA File Serving with NAS Clusters

Failover Cluster Design Criteria: Node Utilization

Active/Passive (a.k.a. "Standby" or "Single Active")



Active/Active (a.k.a. "Dual Active" or "Symmetric")



HA File Serving with NAS Clusters

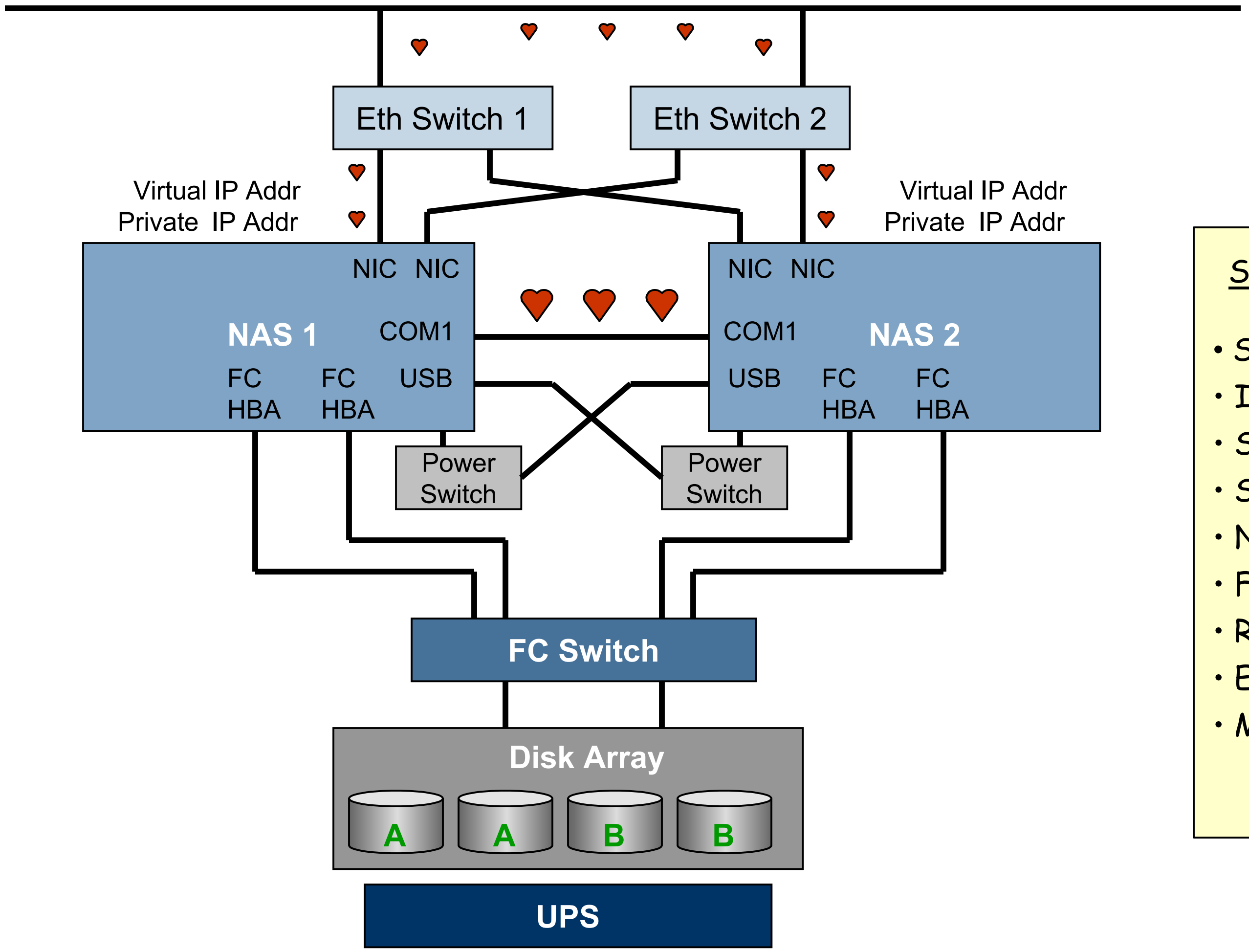
Failover Cluster Design Criteria: Resource Model

Methods for handling access to the pool of available resources:

Method	Description	Trade-off's
Shared Nothing	Nodes are connected to shared storage but each node "owns" it's IP addresses, file systems, mount points, services, etc. Upon failure, a surviving node takes over the resources for the failing node.	+ No write synchronization issues. + No DLM or distributed FS required. + Minimum network overhead. - Limited load balancing.
Shared Everything	Nodes share simultaneous access to the same disk resources. A distributed file system or lock manager controls synchronization of access to prevent data corruption.	+ Excellent load balancing potential + SSI (single system image) - DLM or distributed FS required - high DLM synchronization traffic
Mirroring	Nodes own their disk resources, but continuously perform copy operations to mirror their data and make it available to other nodes.	+ Fast disaster recovery - High overhead for synchronization

Anatomy of a NAS Failover Cluster

Objective: Build a 2-node Active/Active Shared-nothing NAS Failover Cluster using commodity hardware and open source software.



Services/Processes

- Serial Ring Heartbeat
- IP Addr Takeover
- STONITH
- SMB Monitor
- NFS Monitor
- FS Failover
- Resource failback
- Eth. B-cast Heartbeat
- Multi-path HBA driver

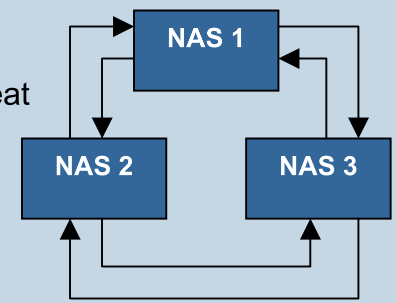
HA File Serving with NAS Clusters

Failover Cluster Heartbeat Mechanisms

Serial Heartbeat Notes:

- Ring architecture has low comm overhead
- Digitally signed and encrypted for security
- Programmable frequency (usually 1-2 Hz)
- Purpose = reset watchdog & exchange state data
- Heartbeat is bidirectional RS-232 over null modem
- Less practical when clusters size > 2 nodes

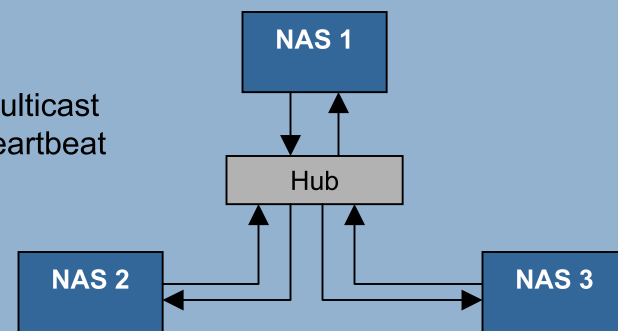
Ring Heartbeat



Ethernet Heartbeat Notes:

- Broadcast architecture has high comm overhead
- Digitally signed and encrypted for security
- Programmable frequency (usually 1-2 Hz)
- Purpose = reset watchdog & exchange state data
- Heartbeat is multicast IP

Multicast Heartbeat



Heartbeat Medium Alternatives:

Medium	Reliability	Latency	Slots	Scaling	Cost
Shared ethernet	Medium	Poor	1	Good	Low
Dedicated ether.	Medium	Good	1	Good	High
Serial	Good	Good	None	Poor	Low
IrDA	???	Good	None	Good	Low

HA File Serving with NAS Clusters

Mechanisms for network resource takeover

Type	Description/Attributes
IP Address Takeover	• Virtual IP addr is bound to MAC address of a NIC on the failover node using ARP spoofing • Simplest of the 3 options, but slower and less reliable than MAC address takeover due to the need to perform ARP cache refresh for each attached client
MAC Address Takeover	• MAC address is virtualized and bound to the virtual IP on the failover node • Nearly instantaneous failover and resolves the problem of the stale client ARP cache • Messy implementation due to having to “manufacture” a unique MAC address • Not all NIC’s allow this, so may need 1 spare NIC per IP addr
Dynamic DNS Reconfiguration	• Reconfigure the DNS to reflect the new IP – MAC address mapping • Slowest of the three options • Excellent load balancing potential
Net Address Translation	• IP packet header is rewritten and forwarded with a potentially masqueraded address • Excellent for load balancing and firewalling • Requires a “front end” system to receive, modify and forward packets

<u>About the Address Resolution Protocol (ARP):</u>
ARP allows a host to find the physical address of a target host on the same physical network, given only the target’s IP address.

The ARP Protocol - To determine P_B (B’s physical addr) from I_B (IP addr):
1) Host A broadcasts an ARP request containing I_B to all machines on the net
2) Host B responds with an ARP reply that contains the pair (I_B, P_B).

HA File Serving with NAS Clusters

Key Development Challenges:

- Preventing “Split Brain Syndrome”
  - Quorum Lock Disk
  - Quorum Server
  - SCSI Reservations
- Ensuring data integrity
  - Enforce shared nothing policy
  - Distributed Lock Manager
- Handling file protocol failover as seamlessly as possible
  - NFS is stateless
  - SMB/CIFS is stateful
- Synchronizing configuration changes between nodes
  - Real-time synchronization
  - Real-time operation logging, fulfillment at failover time
- Reducing/Minimizing Failover Time
  - Avoid false failures
  - Retries & failure qualification → longer failover

HA File Serving with NAS Clusters

NFS Failover:

- Stateless nature makes it very suitable for failover. It was originally designed to “survive” through a reboot cycle.
- MUST run NFS in SYNCHRONOUS mode!
- Numerous external daemons were added to overcome deficiencies due to it’s stateless design goal
  - statd (status monitor daemon)
  - lockd (lock manager)
  - mountd (mount manager)
- Honoring client mount points after failover
  - Synchronize rmtab, xtab, exports, etab
- Preventing stale NFS file handles
- Lock failover
- Impact on NFS clients
  - Delayed acknowledgement of I/O request

HA File Serving with NAS Clusters

SMB/CIFS Failover:

- Stateful nature makes it challenging for failover
- Connection oriented, if lost clients can easily reconnect
- Supports rich locking mechanisms and semantics, making lock failover very difficult and complicated
- Single configuration file to synchronize (smb.conf - Linux/Unix)
- Honoring client connections after failover
    - Client driven, when connection is lost client must reconnect
- Impact on CIFS clients
    - Pending I/O request is lost
    - Client attempts retries and will timeout and lose connection if failover time is too long
    - Connection lost
    - Locks lost, client must re-request locks after reconnecting unless lock failover is supported
    - Many client applications, such as MS Office, use a file-based semaphore instead of locks supported by the protocol to avoid losing locks and causing data corruption

HA File Serving with NAS Clusters

Future Direction for HA NAS:

- Larger clusters (4 node, 8 node)
- NAS-SAN fusion
- “Universal Network Fabric” (file and block access)
- Automated configuration and management
- More “9’s” – less downtime
- More fault-tolerant hardware – with built-in failover
- Distributed File Systems - Single System Image
- Tighter integration with remote mirroring
- Improved disaster recovery
- Changes to the file protocols to accommodate failover, lock preservation, session restore
- Client app’s modified to tolerate failover (retries, auto-reconnect, lock refresh)
- Automatic load balancing between cluster nodes