SNIA

Education

High Availability Using Fault Tolerance in the SAN

Mark S Fleming, IBM

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced without modification
  - The SNIA must be acknowledged as source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.

Abstract

High Availability Using Fault Tolerance in the SAN

- Availability is typically expressed as a percentage of uptime in a given year. Modern SANs have developed numerous methods using hardware and software fault tolerance to assure high availability of storage to customers. From the storage controller through the paths to the end servers, various methods exist to ensure continued access to storage in the event of faults and hardware/software failures. This tutorial will help the end user understand:
  - **The concept of availability vs. uptime**
  - **How High Availability and Fault Tolerance are related**
  - **Availability methods on the controllers, paths and servers**
  - **Availability methods on the software side, such as clustering**
  - **Common single points of failure**

The 50,000 Foot View

- High Availability Using Fault Tolerance in the SAN
  - Promises a certain amount of uptime
  - Promises access to critical functions of the system
  - Allows system to handle faults or failures in a system
  - Involves redundant components
  - Allows component upgrades or maintenance without impacting availability
- What This Is Not
  - Not a guarantee of 100% availability
  - Not an inexpensive solution
  - **Susceptible to poor implementation**

The Language of HA

- Uptime
  - Measure of the time a computer system has been "up" and running (does not imply availability)
- Availability
  - The proportion of time a system is production capable
- High Availability
  - System design protocol and associated implementation that ensures a certain absolute degree of operational continuity during a given measurement period
- Fault Tolerance
  - The ability to continue properly when a hardware or software fault or failure occurs. Designed for reliability by building multiples of critical components like controllers, adapters, memory and disk drives.
- Redundancy
  - The duplication of components to ensure that should a primary resource fail, a secondary resources can take over its function

Definition sources : Google, wikipedia

And Now For The Parts

- Storage Controller/Controller
  - The control logic in a storage subsystem that performs, among other things, command transformation and routing, I/O prioritization, error recovery and performance optimization
- Fabric
  - Interconnection method that allows multiple hosts and/or storage devices connected with a multi-port hub, simultaneous and concurrent data transfers
- Adapter
  - Circuit board that provides I/O processing and physical connectivity between a server and storage device
- Multipathing
  - The use of redundant storage networking components (adapters, cables, switches) responsible for the transfer of data between the server and the storage

Definition sources : Google, wikipedia

The Seven R's Of High Availability

- Redundancy
  - Eliminate single points of failures
- Reputation
  - What's the track record of the key suppliers in your solution?
- Reliability
  - How dependable are the components and coding of the products?
- Repairability
  - How quickly and easily can suppliers fix or replace failing parts?
- Recoverability
  - Can your solution overcome a momentary failure and not impact users?
- Responsiveness
  - A sense of urgency is essential in all aspects of High Availability
- Robustness
  - Can your solution survive a variety of forces working against it?

Five Nines…And Then Some

Availability	Downtime/Yr	Downtime/Mo	Downtime/Wk
90%	36.5 days	72 hours	16.8 hours
95%	18.25 days	36 hours	8.4 hours
98%	7.30 days	14.4 hours	3.36 hours
99%	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes
99.9%	8.76 hours	43.2 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99%	52.6 minutes	4.32 minutes	1.01 minutes
99.999%	5.26 minutes	25.9 seconds	6.05 seconds
99.9999	31.5 seconds	2.59 seconds	0.605 seconds

Hit Me Again!

- Redundancy Based Fault Tolerance
  - Designed to ensure it takes two independent local faults, in a short time period, to cause end-to-end failure
- Even the best designs can be defeated
  - Undetected faults : single local fault not detected and addressed
  - Dependent faults : two faults assumed to be independent are actually related
  - Out-of-scope Faults : additional categories of faults not addressed by the design

Why Didn't I See That?

- Undetected Faults
  - Single local fault not detected and addressed
  - Complexity of SAN access paths
    - Often hundreds or thousands of SAN access paths
    - Many thousand SAN configuration changes, often undocumented
    - Relationship between components often not understood well enough to ensure that redundant fault tolerance is adhered to
- Example
  - Planned switch upgrade brings down critical business because a fault in the redundant access path was not detected
  - Vendor merges have combined previously separate redundant paths into one when combined under new controller

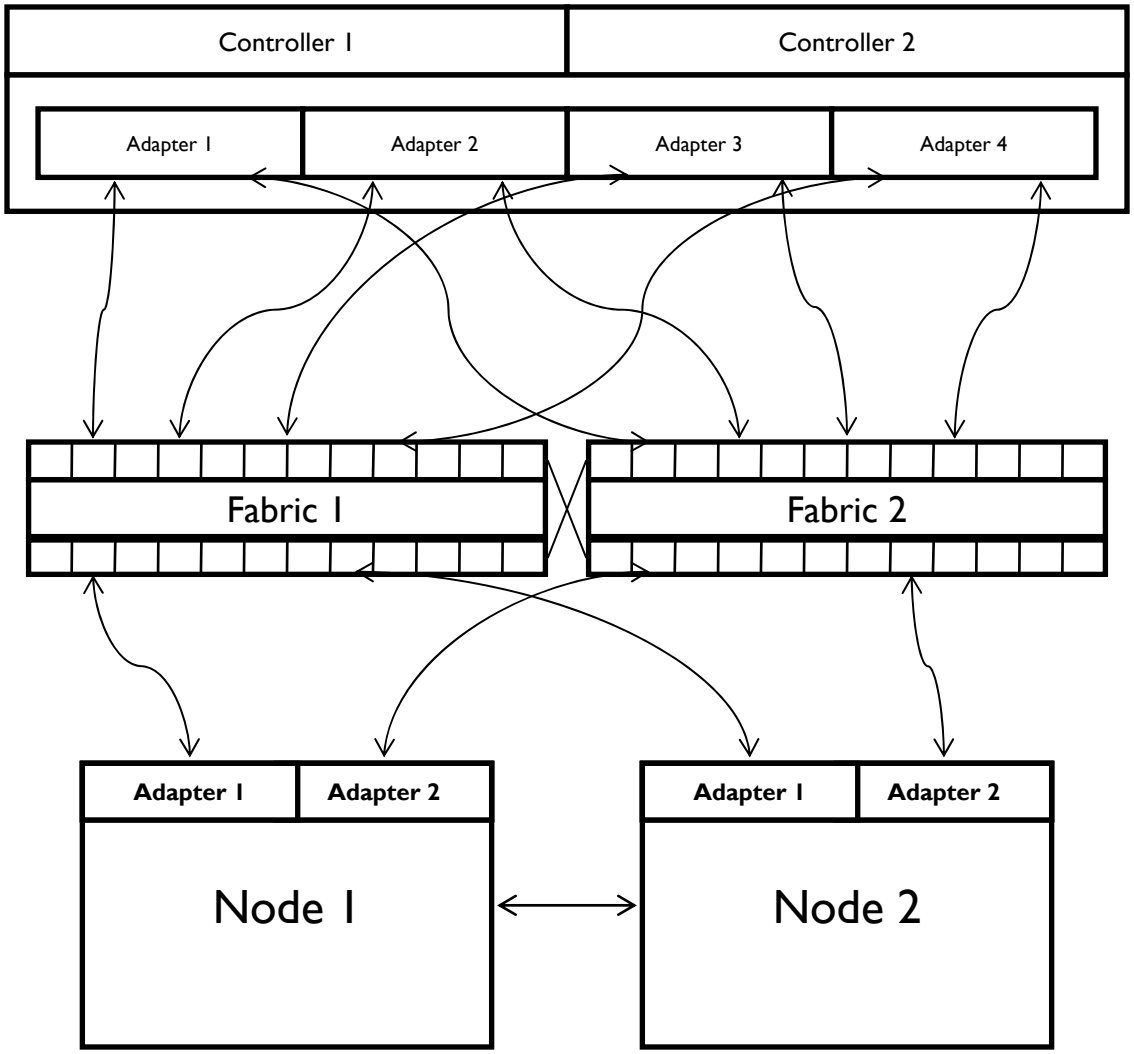
The Two Of Us Together

- Dependent Faults
  - Two faults assumed to be independent are actually dependent
  - Operations often performed twice, once for each redundant system
    - Operation error initiated in one system can be replicated to redundant system
- Example
  - Application downtime caused by zoning errors made in one fabric repeated across redundant dual-fabrics

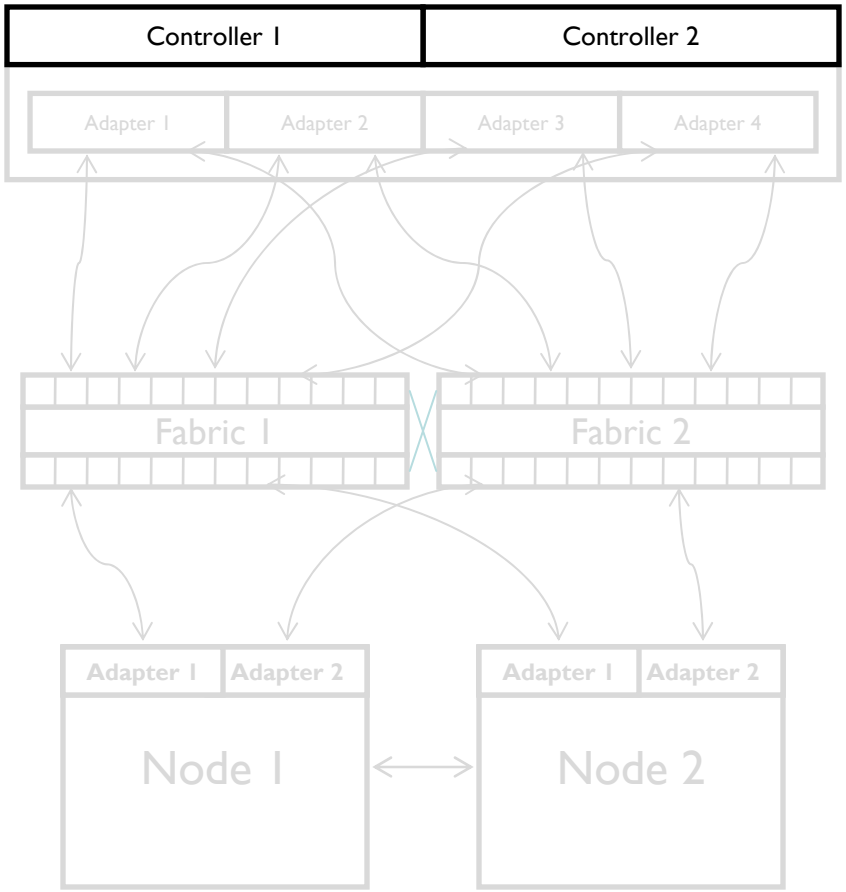
That Can't Happen Here

- Out-Of-Scope Faults
  - Faults that were not anticipated in the original fault tolerant design
  - Misconfigurations or failure to clean up old configurations
- Example
  - LUN accidentally assigned to two different hosts, resulting in data corruption
  - Reuse of an old HBA in a new server caused downtime because previous zonings using that HBA had not been cleaned up

A "Simple" Fault Tolerant SAN

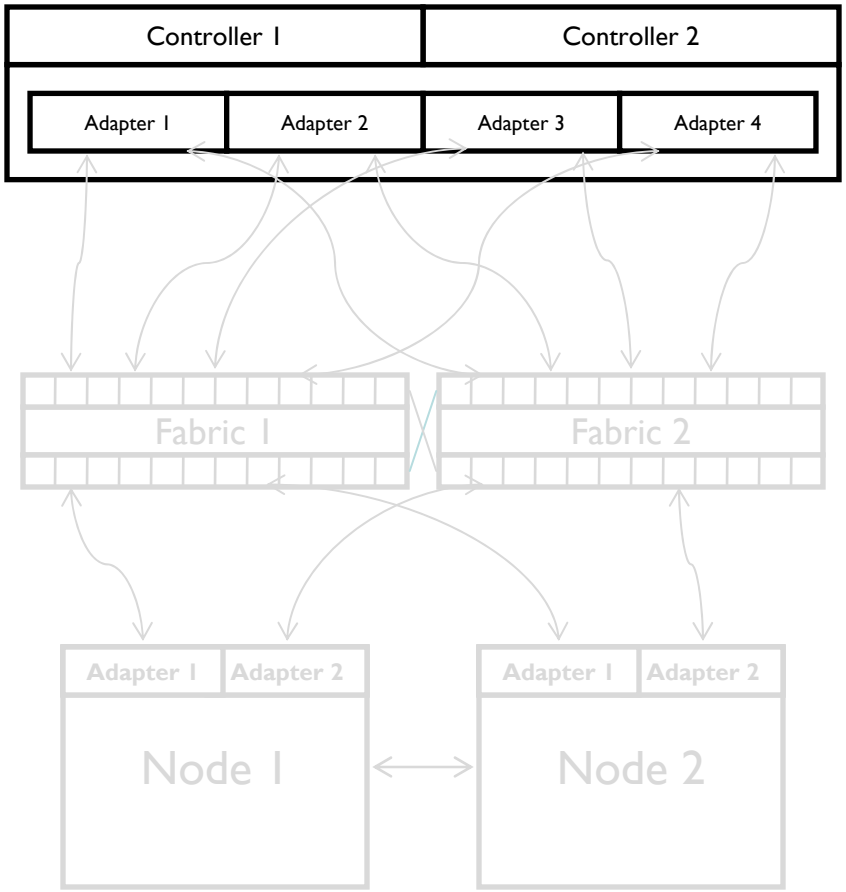


Can't Have Too Much Control



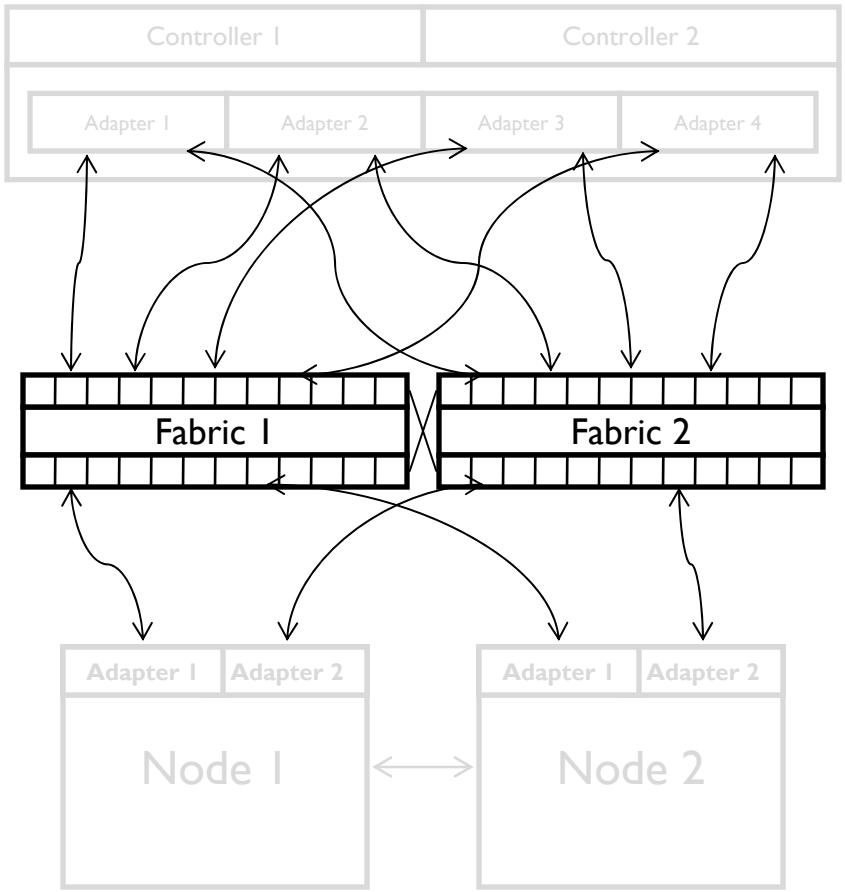
- Storage Controller/Controller
  - Typical features
    - Redundant internal HDs
    - Fault Tolerant Internal Fabric
    - Hot swappable components
    - Predictive failure analysis
  - Redundant controllers allow for
    - Scheduled maintenance
    - Single controller faults
    - System upgrades
  - Single controller boxes can result in downtime if they experience a fault

Like A Six Lane Highway



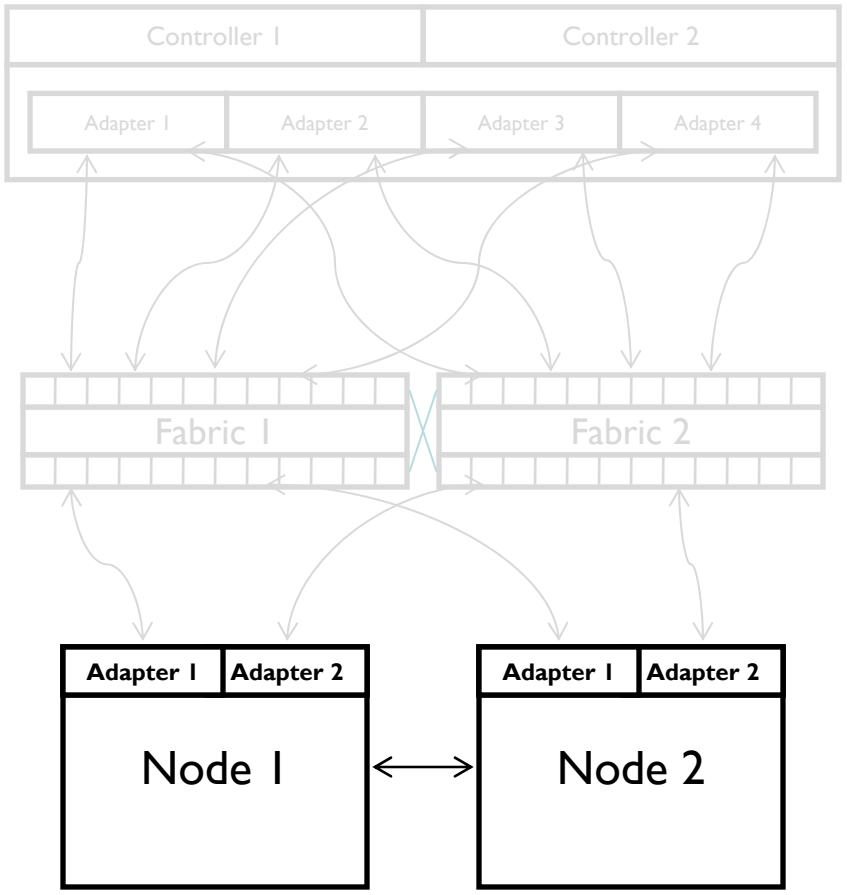
- Adapters
  - Multiple adapters allow redundant access
  - In a good fault tolerant system, the adapters can be managed by either controller, providing greater availability

Are We In "The Matrix?"



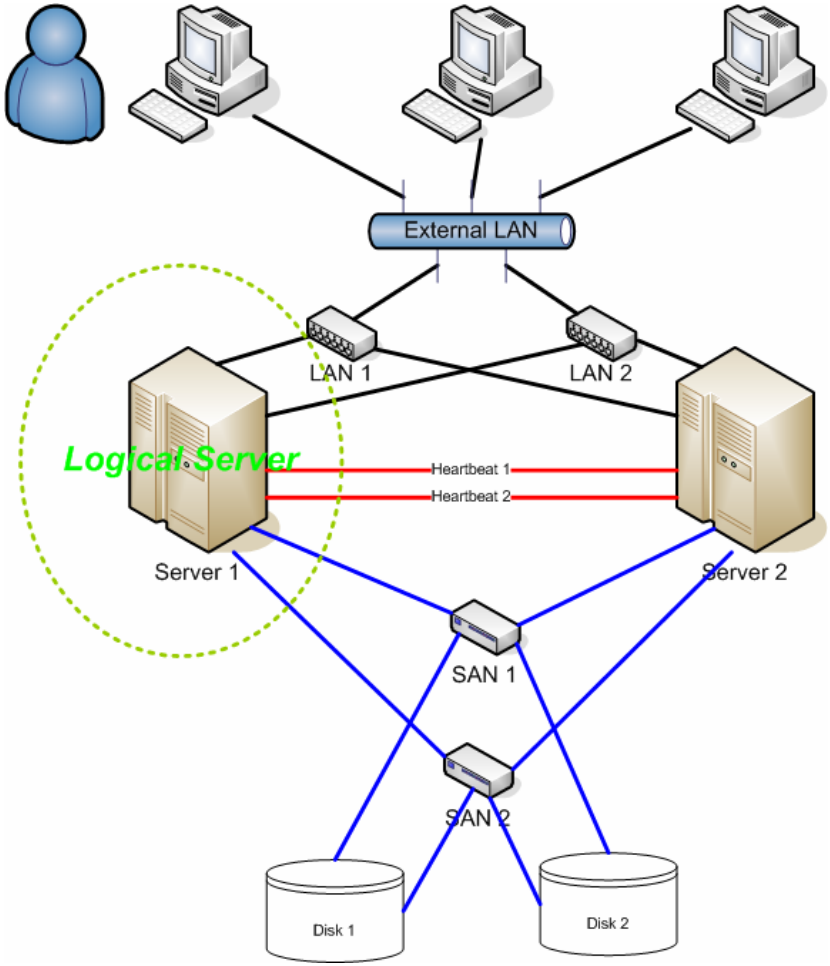
- Fabric
  - Fabric redundancy allows for fault tolerance as well as multiple paths for delivering I/O
  - Need at least two switches
  - Multipathing
    - When path fails, it's disabled and I/O routed to other paths
    - Active/Active or Active/Passive

All The User Really Cares About



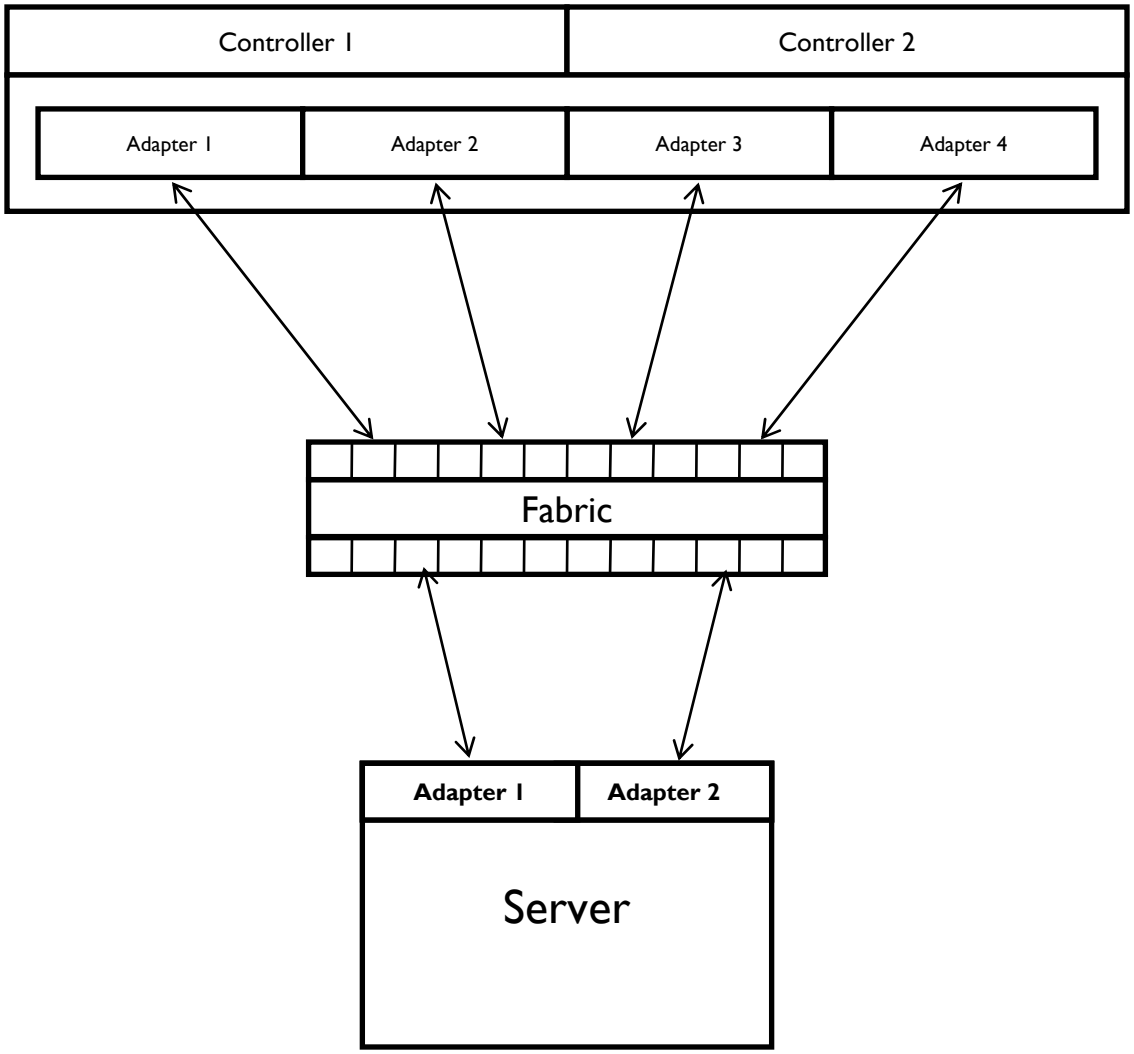
- End Servers
  - There are different ways to provide fault tolerance on end servers
    - Configure a high availability cluster
    - Multiple adapters in each server
    - Fault tolerant hardware configuration on each servers (disks, CPUs, etc)

High Availability Failover Cluster

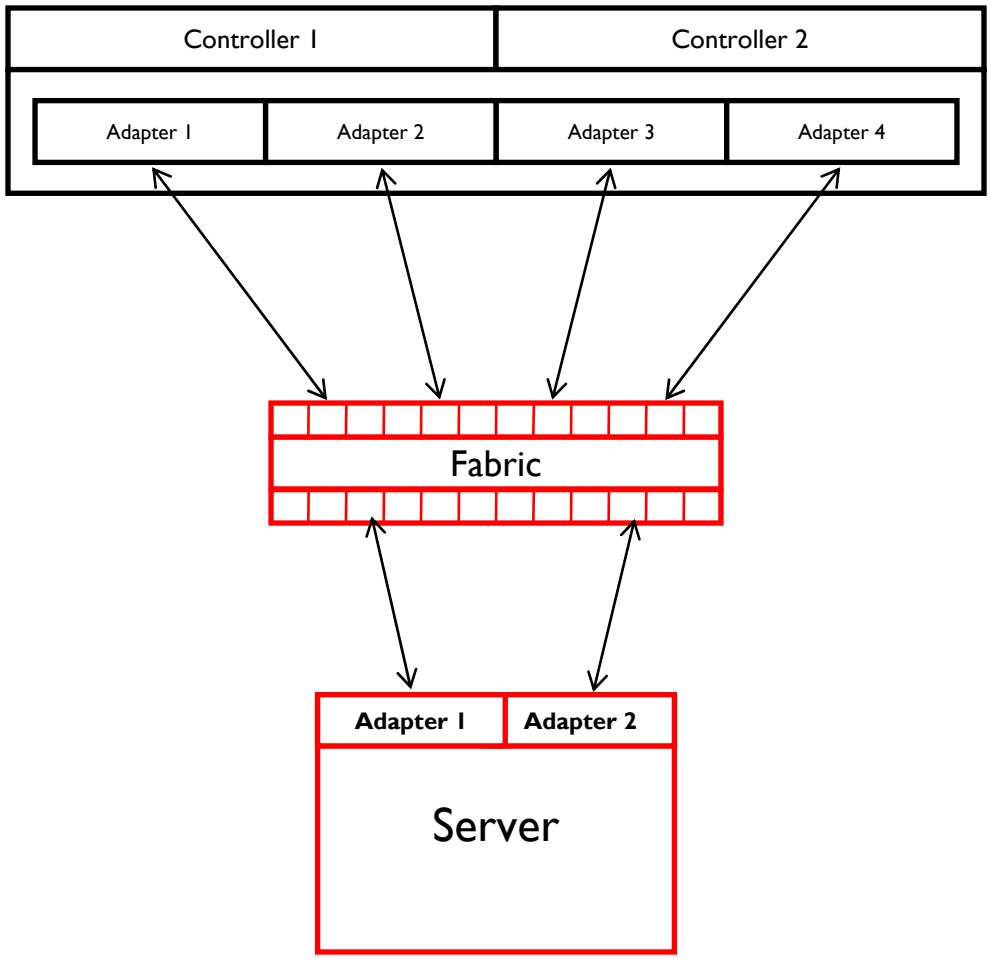


- Server-side Protection
  - Ensure service by providing redundant nodes
  - Builds redundancy through multiple network connections and multiple SAN connections
  - Detects hardware or software faults on one node, restarts application on alternate node
  - Minimum two nodes per cluster, but can scale depending on vendor
  - Active/Active
    - I/O for failed node passed on to surviving nodes, or balanced across remaining
  - Active/Passive
    - Redundant node brought online only when a failure occurs

Looks Good…Feels Good…But…



You're Halfway There



- Despite redundant paths and adapters, there are several single points of failure
- Fabric
  - Single point of failure with only one switch
  - Unable to take unit offline for maintenance, etc, without impacting users
- Server
  - Single server offers no redundancy
  - Unable to take unit down for maintenance or upgrade
  - Multiple paths out, but at mercy of system as a whole

Of Course There's A Downside!

- High Availability Using Fault Tolerance
  - Requires at least two of each component
  - Increases system complexity
  - Increases administrative responsibility
  - Increases capital expenditures
  - Can be incorrectly configured
- Questions You Should Ask
  - Is it worth it?
  - How important is your data to you?
  - Can your business stand any downtime?
  - What is acceptable recovery time?
  - How soon do you want to be back in business?

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA: trackapplications@snia.org

Many thanks to the following individuals for their contributions to this tutorial.

- SNIA Education Committee

Lisa Martinez
Jeffry Larson
Brad Powers
Joshua Tseng

Appendix

Resources

- http://en.wikipedia.org/wiki/Graceful_degradation
- http://en.wikipedia.org/wiki/Redundancy_%28engineering%29
- http://en.wikipedia.org/wiki/High_availability
- http://searchstorage.techtarget.com/news/article/0,289142,sid5_gci1080870,00.html#
- http://en.wikipedia.org/wiki/Myth_of_the_nines
- http://www.informit.com/articles/article.aspx?p=28299&seqNum=3
- http://en.wikipedia.org/wiki/High-availability_cluster