Education

Scale and Availability Considerations for Cluster File Systems

David Noy, Symantec Corporation

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the 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.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Abstract

- Scale and Availability Considerations for Cluster File Systems
  - This session will appeal to server administrators looking to improve the availability of their mission critical applications using a heterogeneous tool that is cross platform and low cost. We will learn how you can use Cluster File System to improve performance and availability for your application environment.

Simultaneous Access To File System

Education SNIA

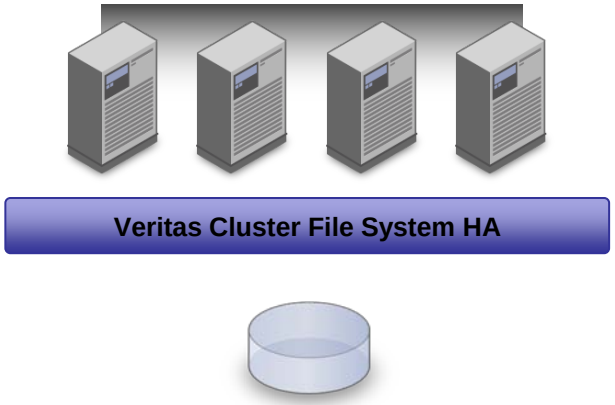
BEFORE

Without a Clustered File System

A traditional file system can only be mounted on one server at any given time, otherwise data corruption will occur.

AFTER

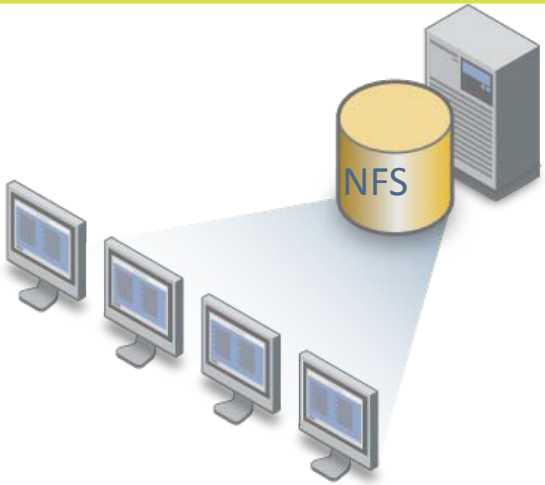
With a Clustered File System



With a cluster file system, all servers that are part of the cluster can safely access the file system simultaneously

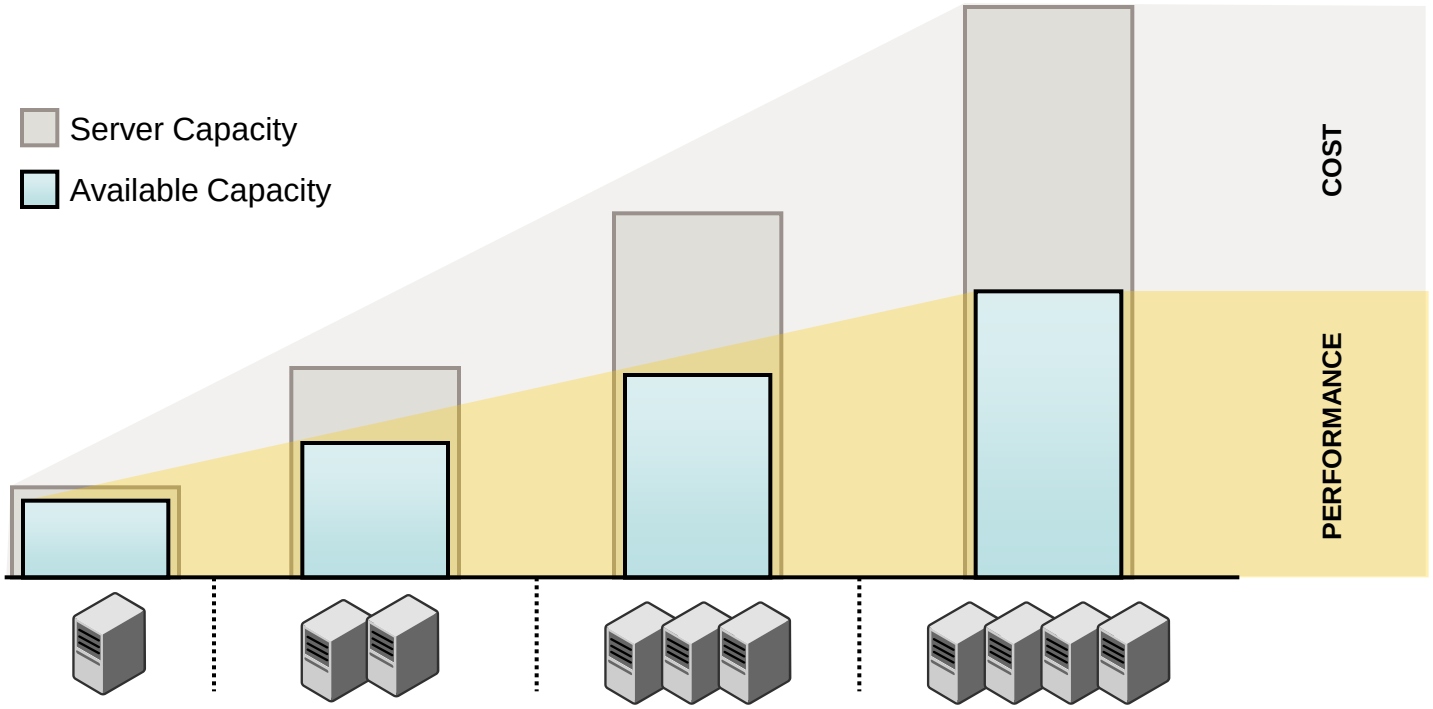
How is a CFS different from NAS?

- Network Attached Storage (NAS)
  - Uses TCP/IP to share a file system over the Local Area Network
  - Higher latency and overhead
  - A file system is mounted via a network based file system protocol (CIFS on Windows, NFS on Unix)



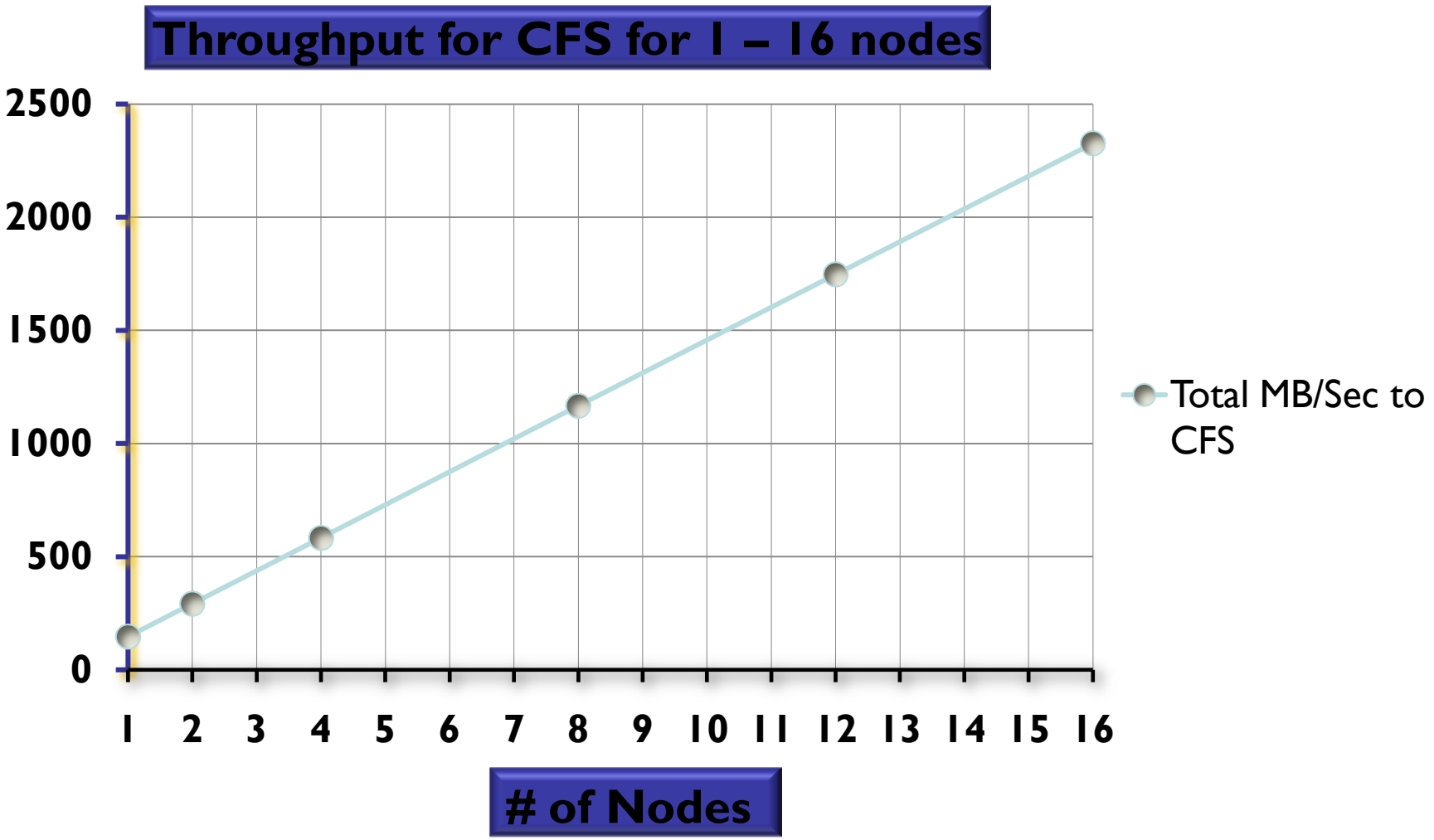
- Cluster File System
  - Looks and feels like a local file system, but shared across multiple nodes
  - Uses a Storage Area Network to share the data,
    - And dedicated network interfaces to share locking information
  - Tightly coupled with clustering to create redundancy
  - With a Cluster File System the application can run on the same node as the file system to get the best performance

Typical Performance Decay at Scale



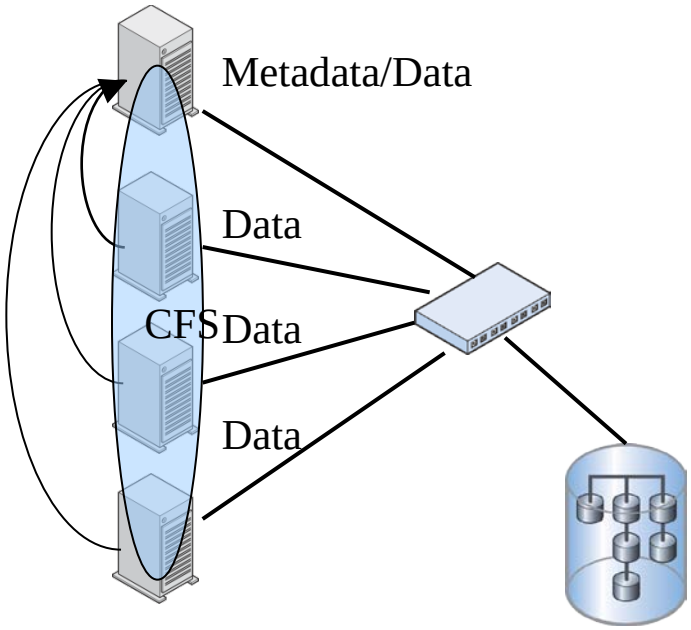
- Bottlenecks as a result of legacy technology
- Performance per node can degrade with more nodes
- Linear scalability generally infeasible (results vary based on app)

Expected Scalability for a CFS



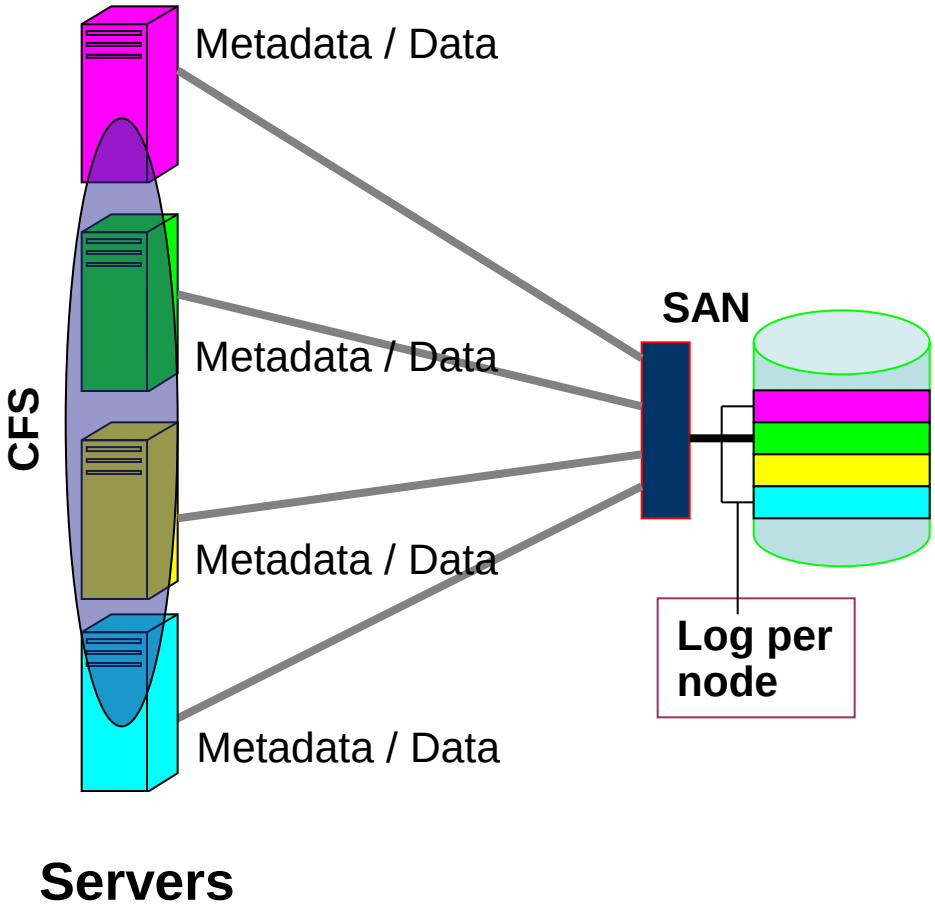
Legacy Meta-Data Management

- File system transactions done primarily by one master node per file system
- That node becomes the bottleneck in transaction intensive workloads
- The amount of transactions performed locally can be improved but it is still the bottleneck
- Acceptable performance for some workloads, not for others

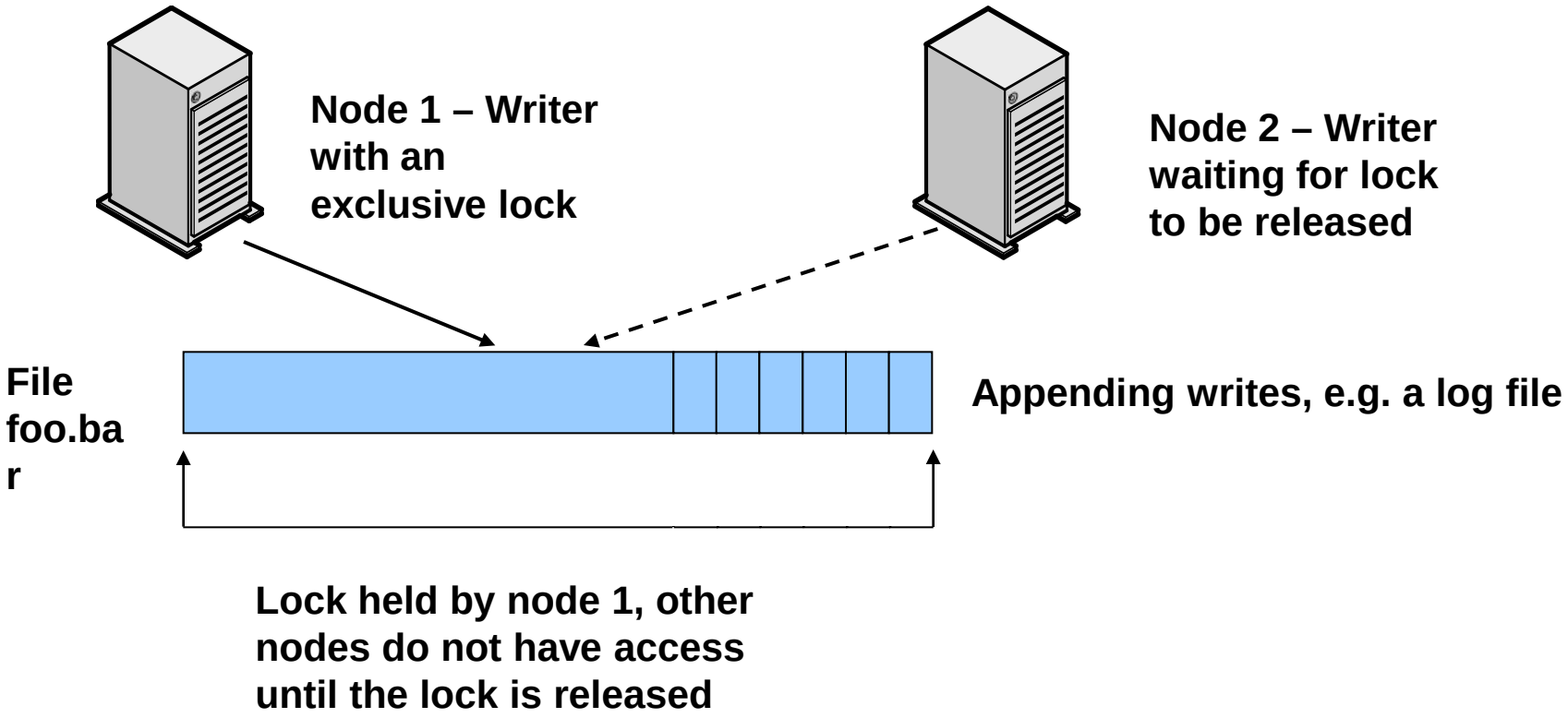


Distributed Meta-Data Management

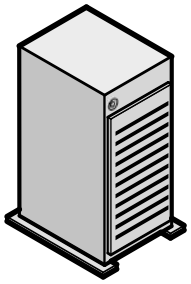
- All nodes in the cluster can perform transactions – No need to manually balance the load
- Not directly visible to end users
- Performance tests show linear scalability
- Scale-out becomes complex beyond 64 nodes



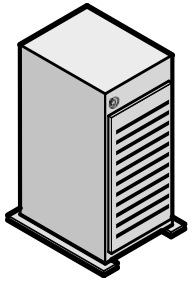
Without Range Locking: Exclusive locking



Range Locking

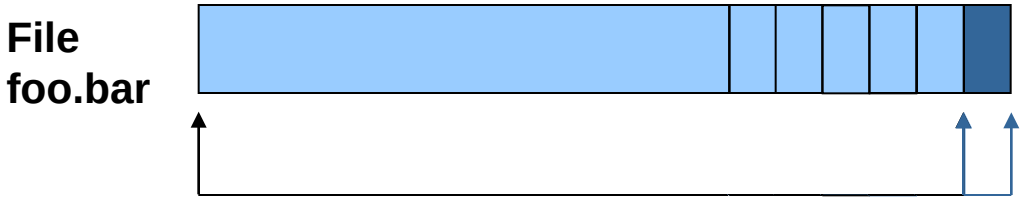


Node 1 - Writer



Node 2 - Writer

Appending writes, e.g. data ingest



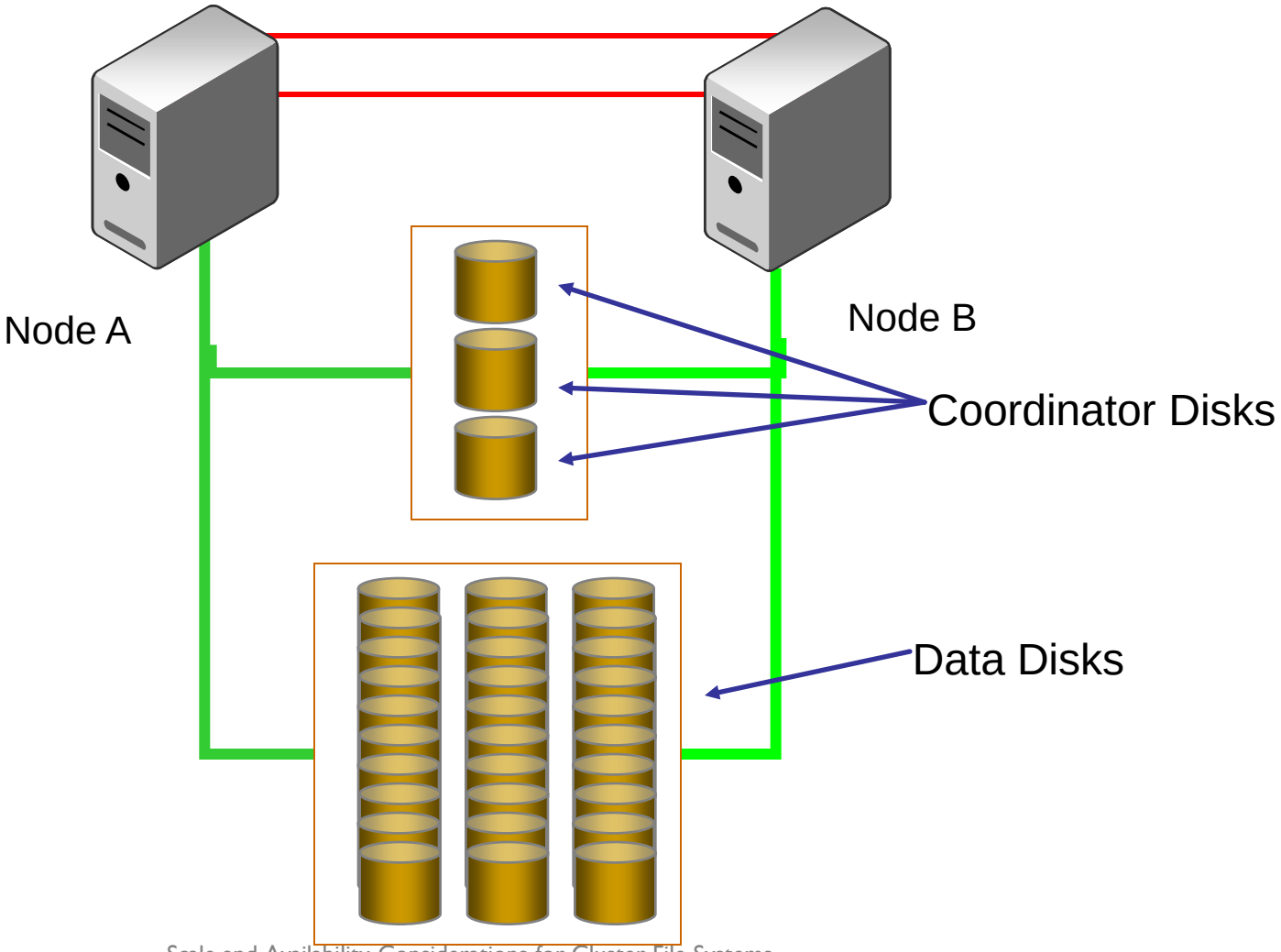
Available for read or write by any nodes

Lock held by node 1, other nodes do not have access

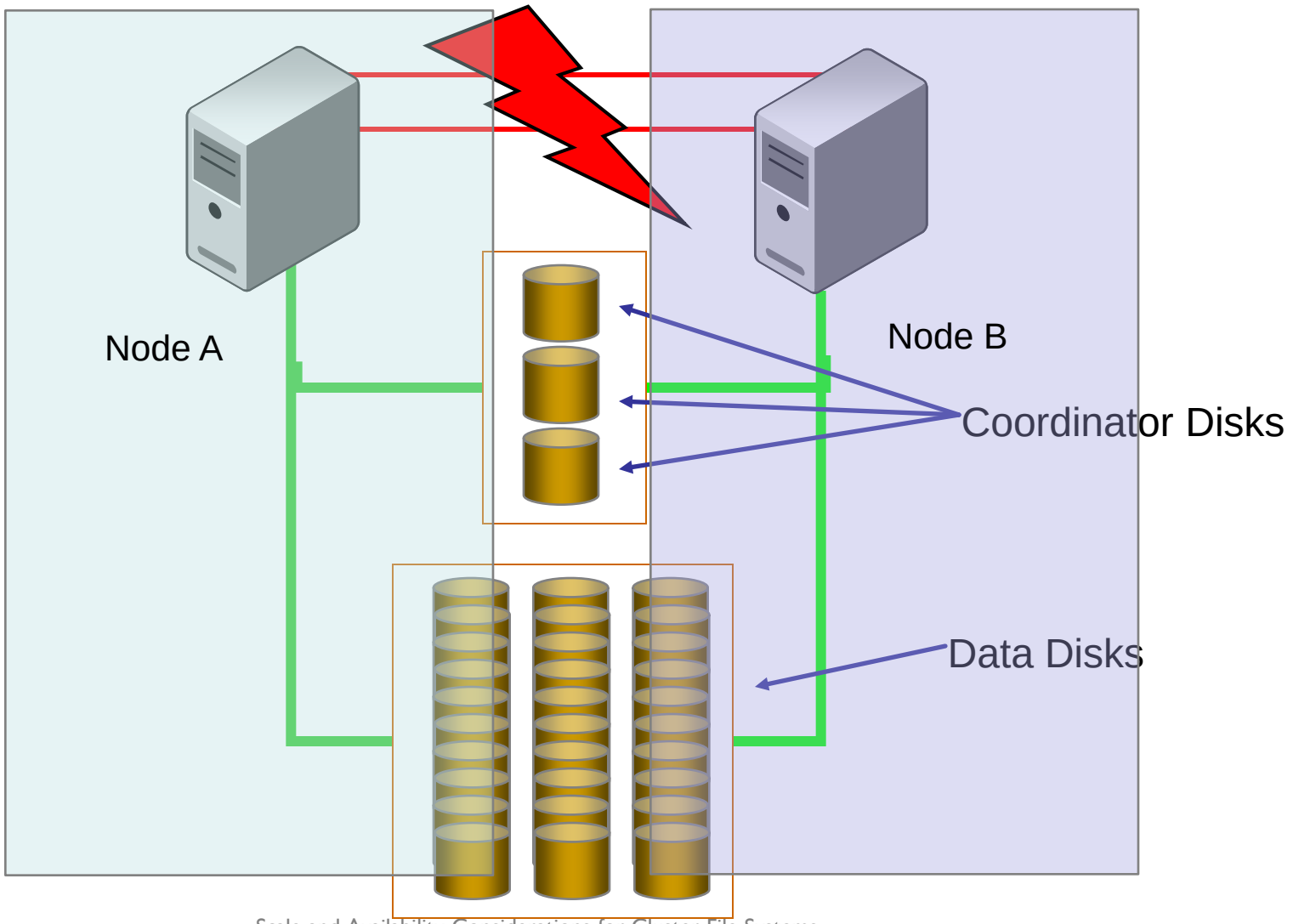
Other Scalability Considerations

- Distributed Meta-Data Management
- Distributed Lock Management
- Minimize intra-cluster messaging
- Cache optimizations
- Reduce or eliminate fragementation

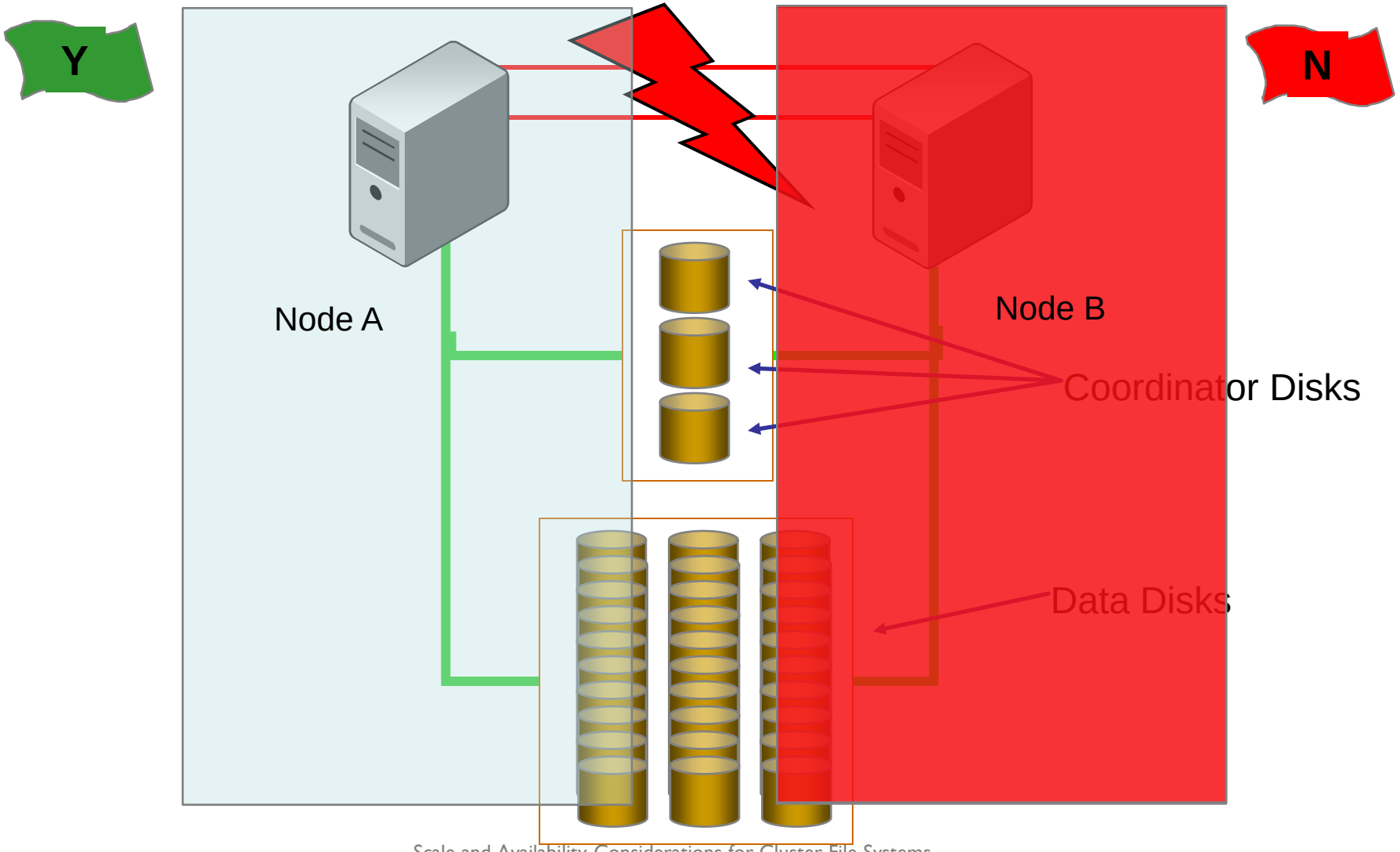
Availability : I/O Fencing



I/O Fencing : Split Brain…

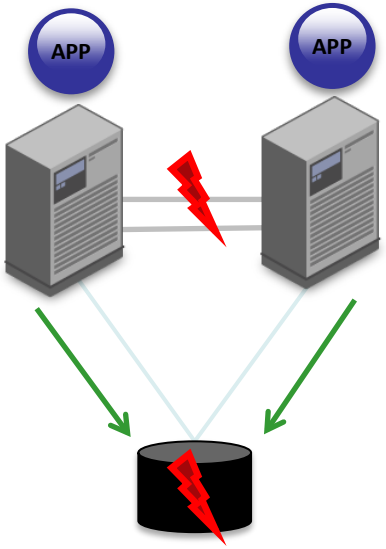


I/O Fencing : Split Brain Resolved



Cluster Fencing

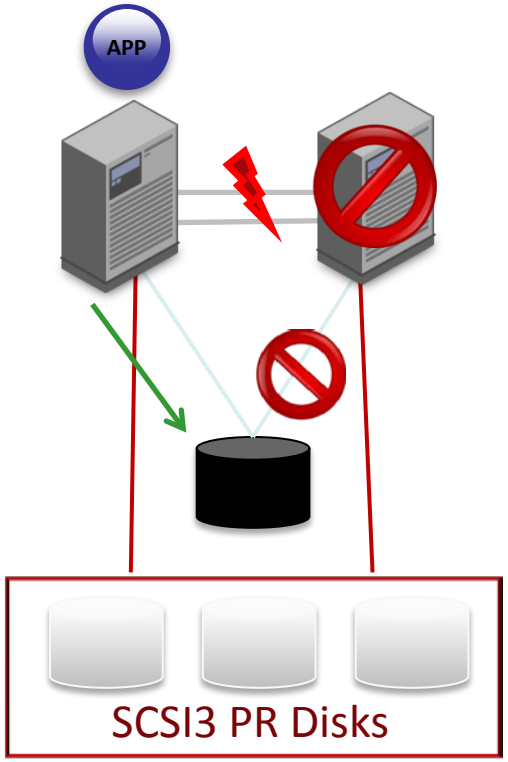
Why Fencing?



Data Corruption

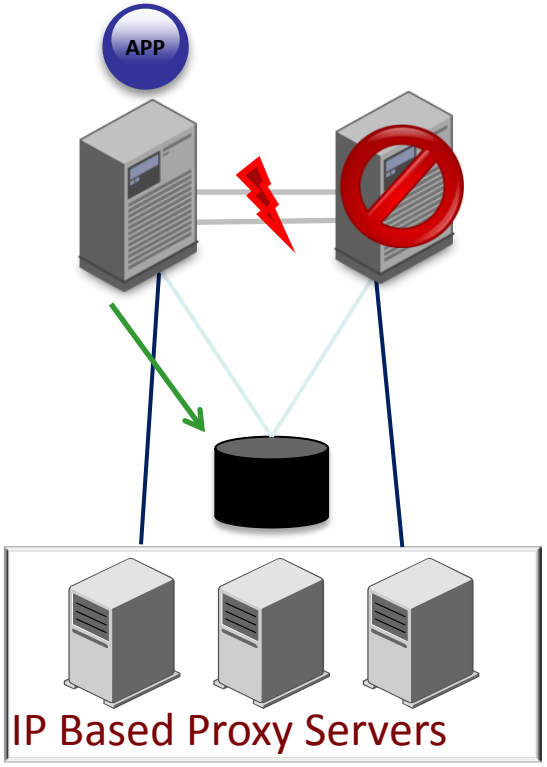
- Need to restrict writes to *current* and *verified* nodes

SCSI-3 Based Fencing



- SCSI3 disks for i/o fencing
- Maximum data protection

Proxy Based Fencing



- Non SCSI3 fencing
- Virtualized environment

Other Availability Considerations

- Robust fencing mechanism
- Tight integration between application clustering and storage layer
  - File System
  - Volume Manager
  - Multi-Pathing
- Quick recovery of CFS objects
  - Lock Recovery
  - Meta-Data Redistribution

Cluster File System Use Cases

Education SNIA

Faster Failovers

- Databases
- Messaging applications
- CRM applications

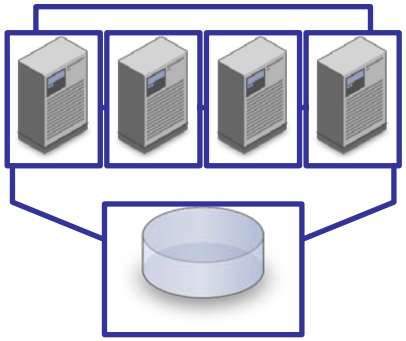
- Dramatically reduce application failover times
- When appropriate, eliminate the cost and complexity of parallel databases

Performance for Parallel Apps

- ETL applications
- Grid Applications
- Parallel databases

- Linear scalability and performance for parallel applications
- Ideal for grid compute

Improve Storage Utilization



- Pool storage between servers eliminating storage islands
- Eliminate multiple copies of data

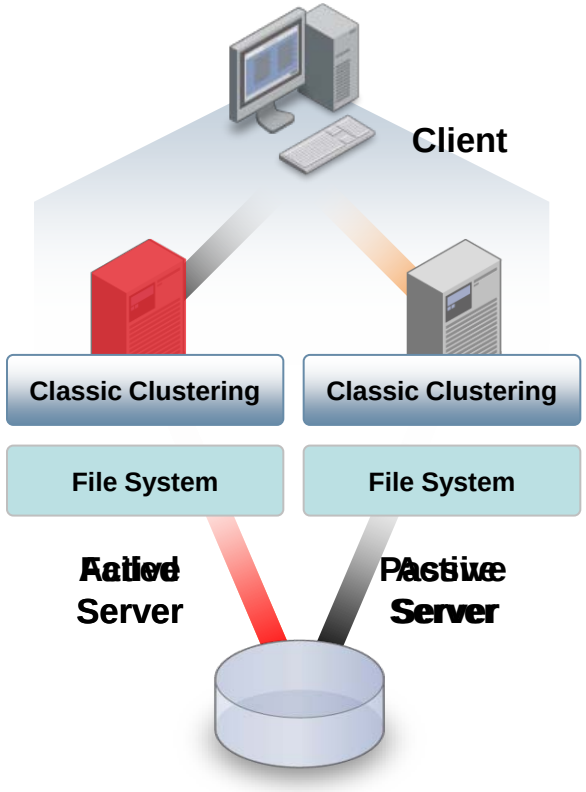
Accelerate Application Recovery

Education SNIA

For Applications that Need Maximum Uptime

- Databases
- Messaging applications
- CRM applications

Classic Clustering Requires Time



Recovery Steps

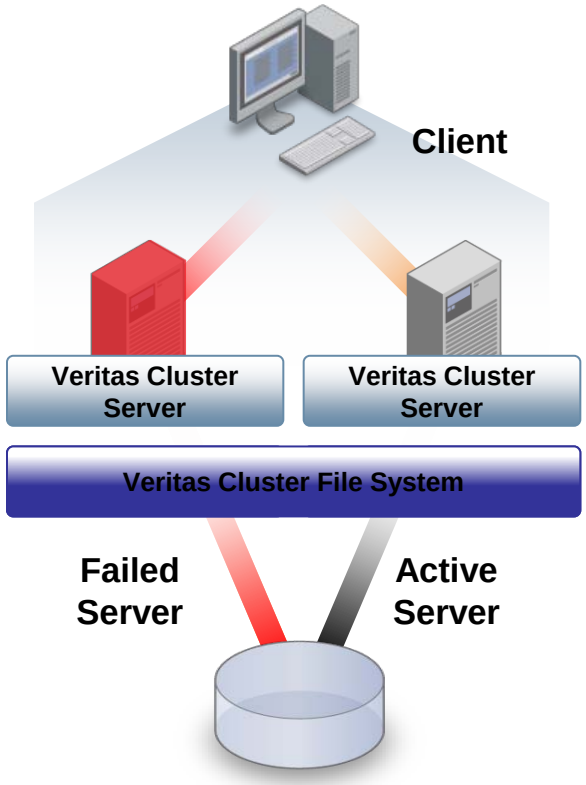
- Detect failure
- Un-mount file system
- Deport disk group
- Import disk group
- Mount file system
- Start application
- Clients reconnect

Accelerate Application Recovery

For Applications that Need Maximum Uptime

- Databases
- Messaging applications
- CRM applications

Failover as Fast as Application Restart

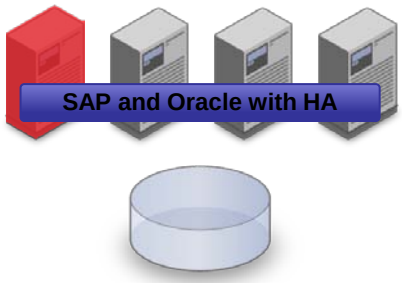


Recovery Steps

- Detect failure
- ~~Un-mount file system~~
- ~~Deport disk group~~
- ~~Import disk group~~
- ~~Mount file system~~
- Start application
- Clients reconnect

Case Study : Reduced Downtime Cost

Before: Ground Control System Using CRM and DataBases

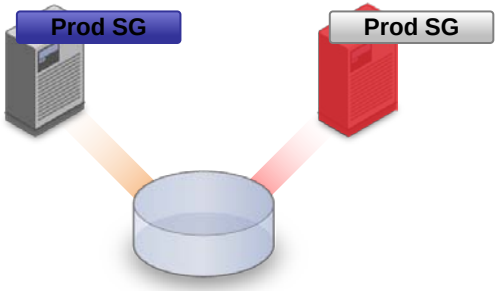


6x 30 Minute Failovers / Year

Traditional High Availability

- Downtime cost = €1,000,000/plane/day
- # Failovers/yr = 6 Failures @ 30 mins each
- # Planes = 240
- Total Downtime/Year = **3 hours**
- @ 1,000,000/plane/day x 3 hrs x 240 planes
- **Downtime Cost = € 30,000,000**

After: Ground Control System Fast Failover with CFS



< 1 Minute Downtime / Failure

Fast Failover with CFS HA

- Downtime cost = €1,000,000/plane/day
- # Failovers/yr = 6 Failures @ 1 min each
- # Planes = 240
- Total Downtime/Year = 6 minutes
- @ 1,000,000/plane/day x 6 min x 240 planes
- **Downtime Cost = € 1,000,000**

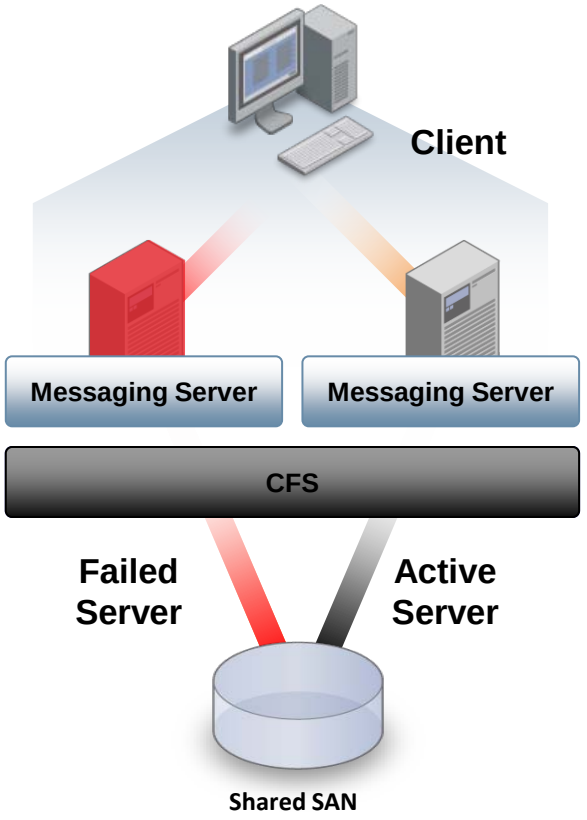
Case Study : Large Casino

Messaging Services	FT Cluster

Messaging server

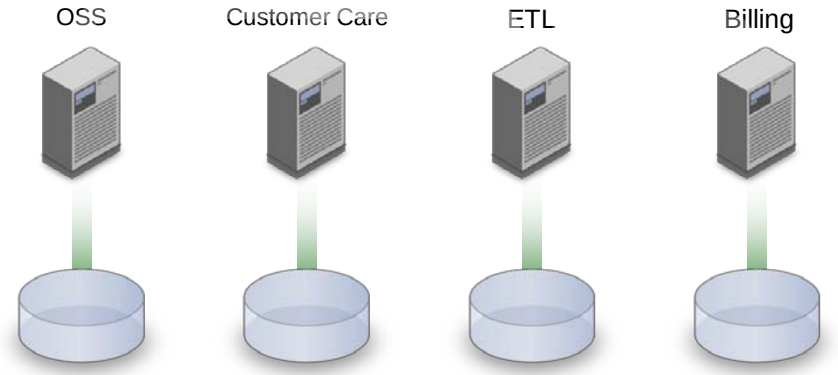
- Using messaging services to run entire casino gaming floor
- Experienced server outage
- Storage failover was very quick
- Failover server required 20 minutes to recover to rebuild message queue
- **Entire casino floor came to a complete stop**

- **Total Savings**
  - **With CFS, a hot-standby server could recover in seconds instead of 20 minutes**



Case Study : Reduce Billing Cycle

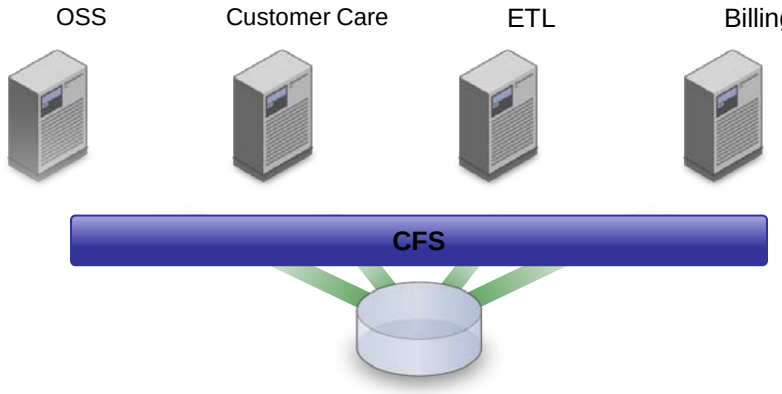
Without CFS



Drawbacks

- Time required to process customer billing included 12 hours of copy time
- For billing systems, time is money
- **Redundant copies of data at each server means 2x the storage requirements**

With CFS



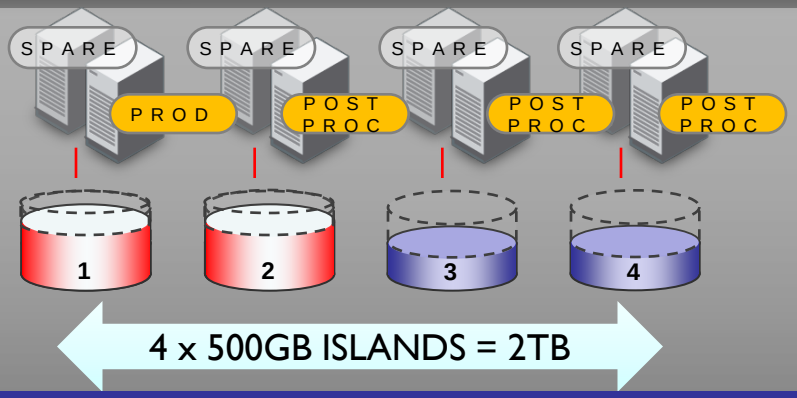
Benefits

- CFS eliminates copy time so one process can start when another completes
- Single copy of data shared among servers
- 12 hour reduction in billing cycle

Case Study : Storage Consolidation

Education SNIA

Traditional Application Architecture



Traditional File System

- Islands of storage zoned to each server
- Storage over-provisioned due to unknown storage growth needs
- When storage is filled, new storage must be provisioned

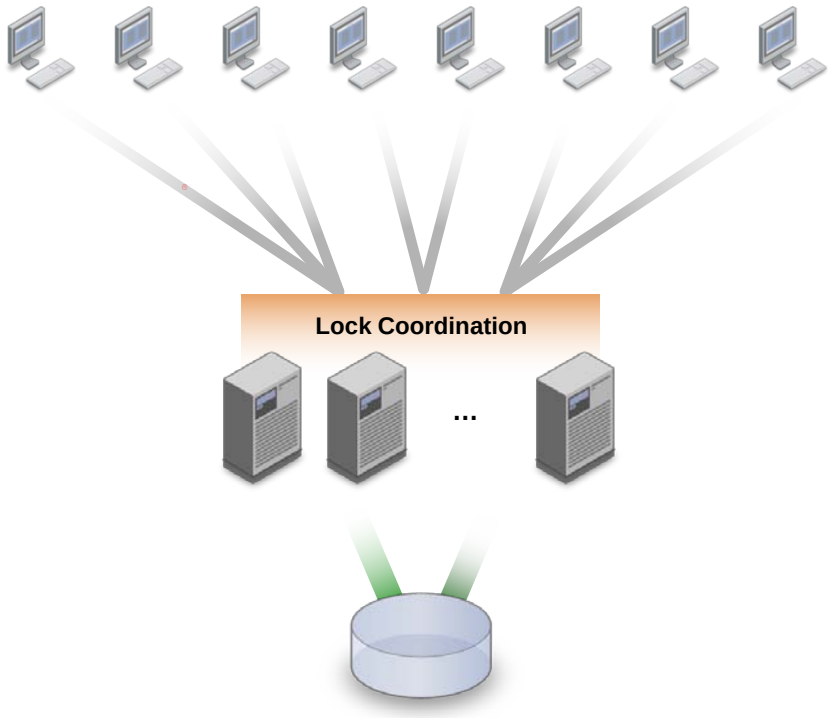
Shared Storage Architecture

Shared Cluster File System

- Storage is accessible to all nodes
- Reduce upfront over-provisioning
- All nodes share common free space
- Minimize idle server and storage resources

Availability and Scale for Clustered NAS

- High Availability of NFS and CIFS service
- Distributed NFS load balancing for performance
- Scale servers and storage independently
  - More servers gives linear performance
  - Flexible storage growth
- Stretch your NFS/CIFS cluster (up to 100 km active/active)
- Choose your platform (Solaris, Sol x86, AIX, RHEL5)
- Integrated with
  - Dynamic Storage Tiering
  - Dynamic Multi Pathing
  - Thin Provisioning Reclamation
- **Increased Price / Performance compared to a similar NAS appliances**



Q&A / Feedback

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

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

- SNIA Education Committee

Karthik Ramamurthy
David Noy