Building a Clustered NAS System

Gradimir Starovic

Storage and Availability Management Group

Symantec

Outline

- Reasons for building a clustered network attached file server
- The building blocks we used and what we had to add, remove or change
- Closer look at the way CIFS service was clustered
- Look back at what we've done and where we are

Who we are, and why we do it

- Symantec Storage and Availability Management group
- Have developed storage and communication software (cluster volume manager and file system, group communication and membership, …), for host-based and over NFS access
- When using the same in a network attached appliance: new features required, and some become less important or need to be changed
- Why do it: demand for unstructured storage and with our experience we were already close to this space

Our initial targets…

- Scalable (now up to PB range, increasing)
- Reliable (tolerate 1-point failures), with fast recovery
- Based on commodity hardware components
- With good performance (100's of MB/s)
- Easy to install, administer and use (appliance-like)
- Access over the standard protocols (NFS, CIFS)

… and some early decisions

- When possible, use the existing components; this included Symantec Veritas stack and Open Source software
- No agents or any other changes on the clients
- Use the soft appliance model, with the option of shipping it pre-installed on the qualified hardware
- Initially implement the basic features and no more

A quick look at where we are now

- Storage Foundation Scalable File Server (SF SFS) – a product used by real customers
- Shipping for over a year, in its 3rd software release, qualified on a number of hardware platforms
- New use cases are pushing the scale/performance boundaries. PB's of data, 100's of millions of files
- Users are coming with requests for new features

About the hardware

- A cluster with shared storage and up to 16 nodes
- Each node (recommended minimum)
  - 64bit dual core and 16GB RAM
  - 4 1 Gigabit Ethernet
  - 2 FC HBA
  - Redundant PS, PXE capable BIOS
- Storage: FC, iSCSI (HCL is the same as for SFHA)

Hardware, examples

- HuaWei-Symantec N8000 series
- Dell PowerEdge 1950 and 2950
- IBM x3250
- SuperMicro 5015M
- also on Virtual Machines (for dev, test, training)

Symantec products used …

- VxVM, Veritas cluster volume manager
- VxFS, Veritas cluster file system
- VCS, Veritas cluster server
- Backup client*
- AV for Linux*

(*) Was not in the initial release, later addition

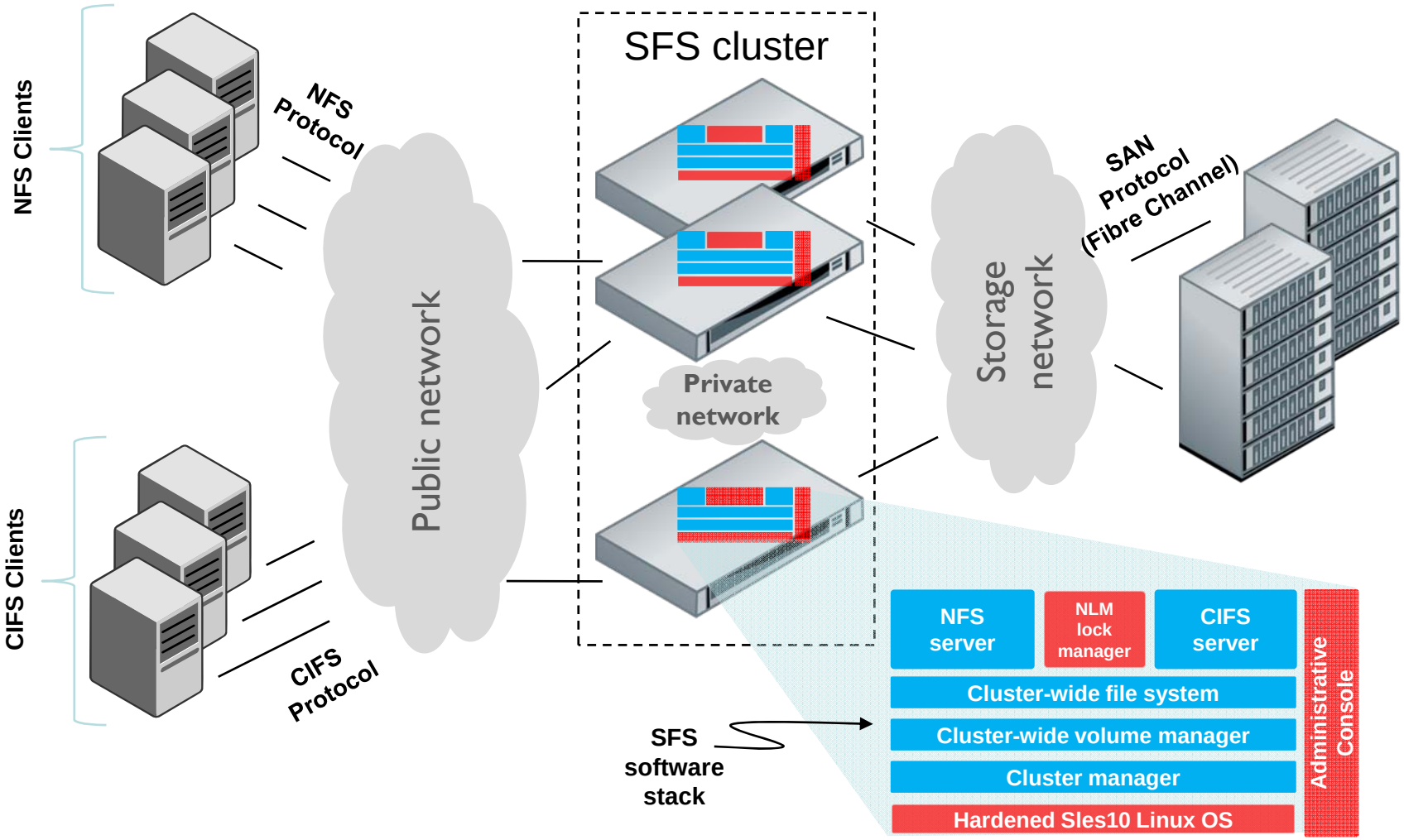
… and their features

- VxVM - virtualizes storage, resizing, replication, migration, scales to 1000's of LUN's, with Dynamic Multipathing
- VxFS - Posix compliance and cache consistency, file and byte-range locking, symmetric nodes with almost linear scalability, with Dynamic Storage Tiering
- VCS - cluster-wide communication, monitoring and failover of critical resources, I/O fencing

Open Source software used

- Linux (Sles10), used as it is except
  - Modified install procedure
  - Some rpms not needed/included
  - Some tuning (TCP, NFS)
- Samba (currently 3.2), used as it is, server side only

SFS architecture



New software in SFS

- Provide greatly simplified view of the system and its storage, networking, event reporting, NFS, CIFS, FTP, and other services
- Hide all the underlying components, no need to access them directly in order to install, configure, tune each of them
- Role-based administration with easy to use menu-driven interface

Admin's view of storage

- New abstractions (old, file system and volume manager ones, are seen as too complex for an appliance)
  - *Disks*, auto discovered
  - *Storage pools*, a pool is a collection of disks
  - *File systems*, created over a number of pools and/or disks
  - *Shares*, a share is an NFS or CIFS export
- Different layouts (e.g. mirrored, striped)
- Online resize

Admin's view of storage (cont)

- Snapshots
  - Existing volume manager (2) and file system (2) snapshot mechanisms replaced with a single mechanism
  - Simplified, automated administration
- Dynamic Storage Tiering, transparent placement and migration of files with configurable policies
  - Based on multi-volume file systems
  - Simplified (2 tiers, simple policies)

Admin's view of networking

- Clients see: a single cluster name, number of IP addresses
- From the SFS point of view these are virtual addresses, each is assigned to a physical IP address
- IP addresses distributed over the nodes, can be moved, with auto failover
- Rely on round robin DNS for load distribution

NFS service

- Active-active, all the nodes serve all the shares
- A small number of export options at the share level and some tunables at the service level
- NFS service made HA, failover of the related resources including the IP addresses
- NLM locking had to be clusterized

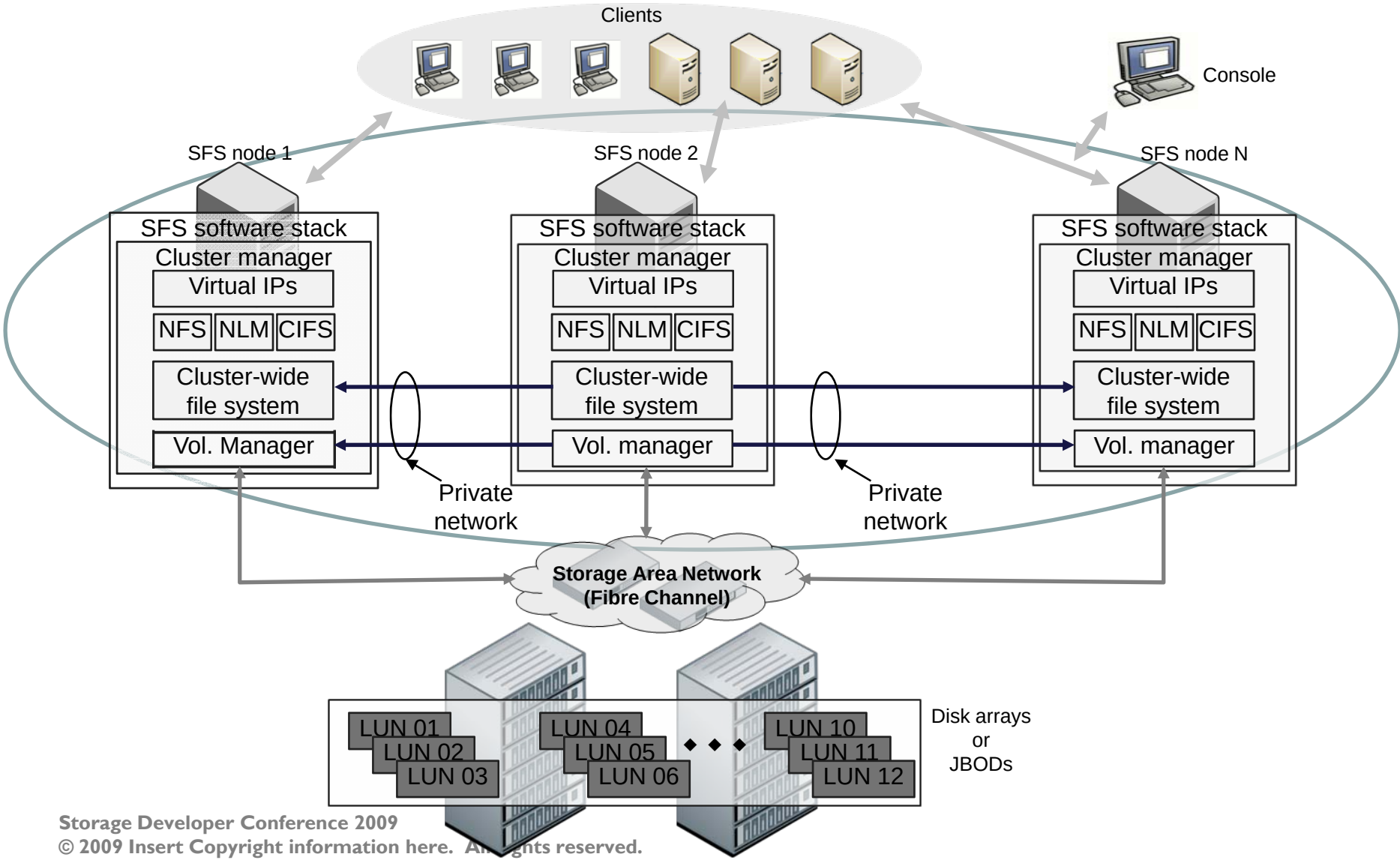
CIFS service

- A new approach to clustered Samba
- Clients see: each share served by all nodes, but internally: each share "owned" by one node (more detail later)
- Allows some load balancing
- Simplified administration of Samba
- Auto and manual creation of home directories
- CIFS service made HA, failover of the related resources including IP addresses

SFS works with remote services

- NIS
- LDAP
- Kerberos
- Active Directory
- NTP
- Admin's access to SFS: single console for the whole cluster; HA resource

SFS architecture



Simple way to clustered CIFS

- Clustered Samba using CTDB exists but we don't use it
  - It wasn't stable at the time and it seemed to have dependencies on other clustering technologies
- Simple, using standard DFS namespaces
- No changes for clients

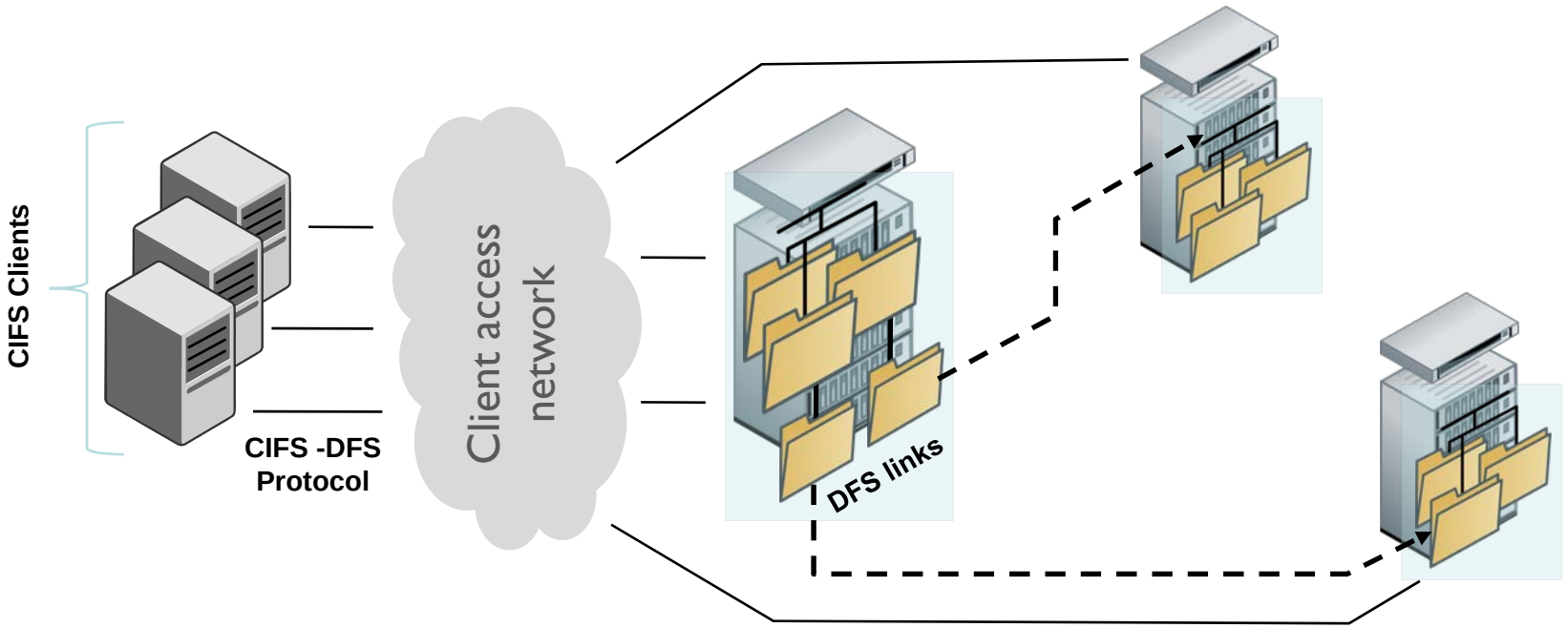
Simple way to clustered CIFS (cont)

- Samba daemons running on every node
- Underlying file systems mounted on every node
- Clients can access an exported share on any node
- In case of failure
  - The existing connections will be broken
  - Clients can reconnect after a short delay, using the same IP address and same share name

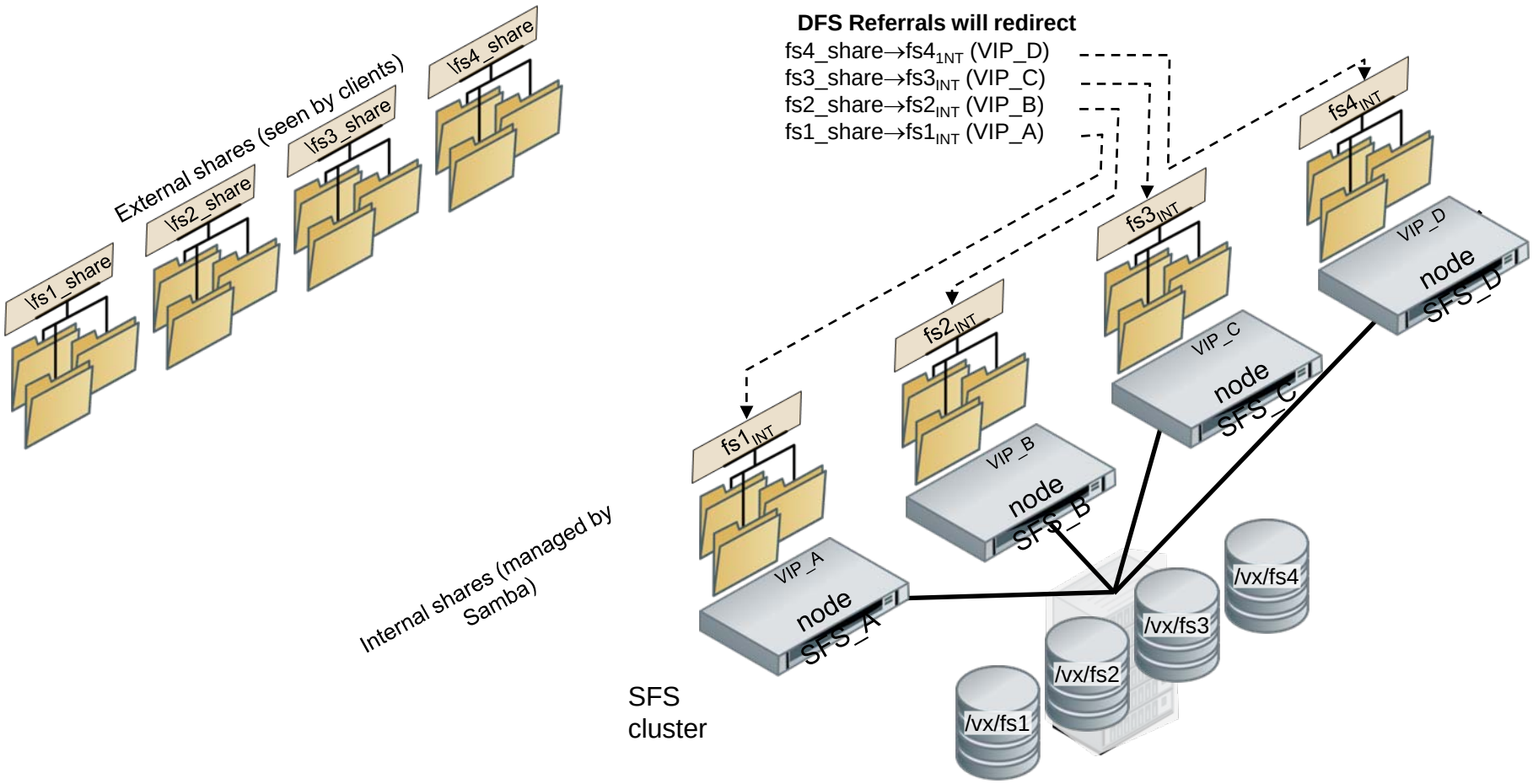
Simple way to clustered CIFS (cont)

- Externally visible share is a DFS link
- Resolving this link adds an extra step for the initial access, when not present in the client cache
- The link is a list of internal targets, not visible to apps running on clients
- Each internal share is served by one of the nodes
- Balanced distribution of internal shares across the nodes with failover for HA

DFS Namespaces



Example: SFS use of DFS



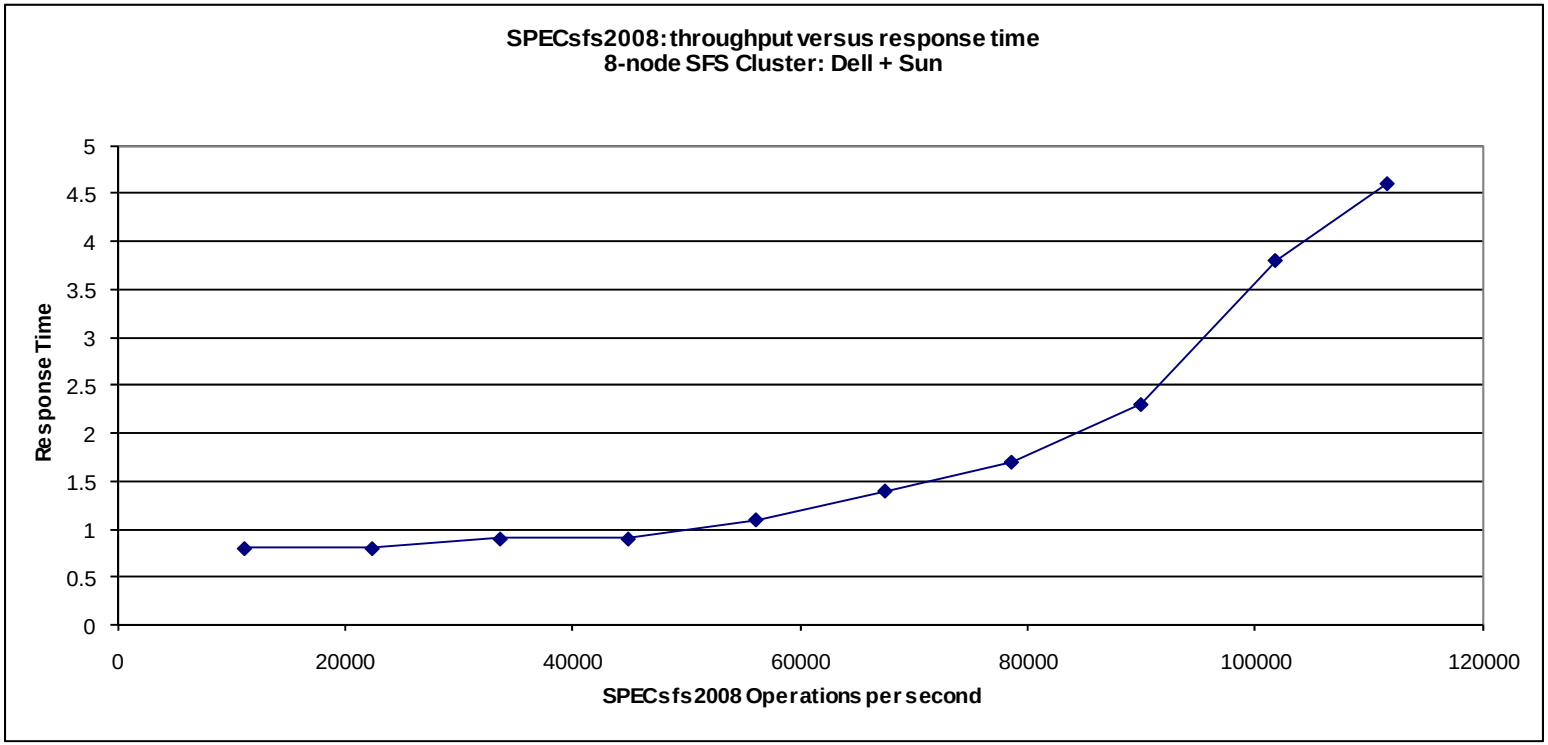
Example: SFS performance

- It obviously depends on hardware and storage used
- Example, data from throughput testing

I/O Type	Sequential read I/O	Sequential write I/O	100% Random read I/O	Random write I/O
2 Node Dell PE1950 w/Sun StorEdge 3510	448.58MB/s	360.54MB/s	153.58MB/s	326.26MB/s
4 Node Dell PE1950 w/Sun StorEdge 3510	904.96MB/s	718.56MB/s	289.24MB/s	688.52MB/s

Example: performance (cont)

Preliminary SPEC2008 NFS results (no tuning)



Where we are

- SFS has satisfied the initial scalability, HA and performance goals
- More things to do: better integration Samba and VxFS (ACL's, quotas)
- Since the initial release requests for more features, such as
  - Web based GUI
  - Backup
  - AV scan
  - Replication

Links

- HuaWei-Symantec link

www.huaweisymantec.com/en/Product___Solution/Products/Storage/NAS/NAS_Series/N8000

- Storage Hardware Compatibility List

seer.entsupport.symantec.com/docs/283161.htm

Thank You!