開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

open

USE IMPROVE EVANGELIZE

NAS Implementation Using OpenSolaris

Sonny Hastomo
IT Architect
Sun Microsystems Indonesia

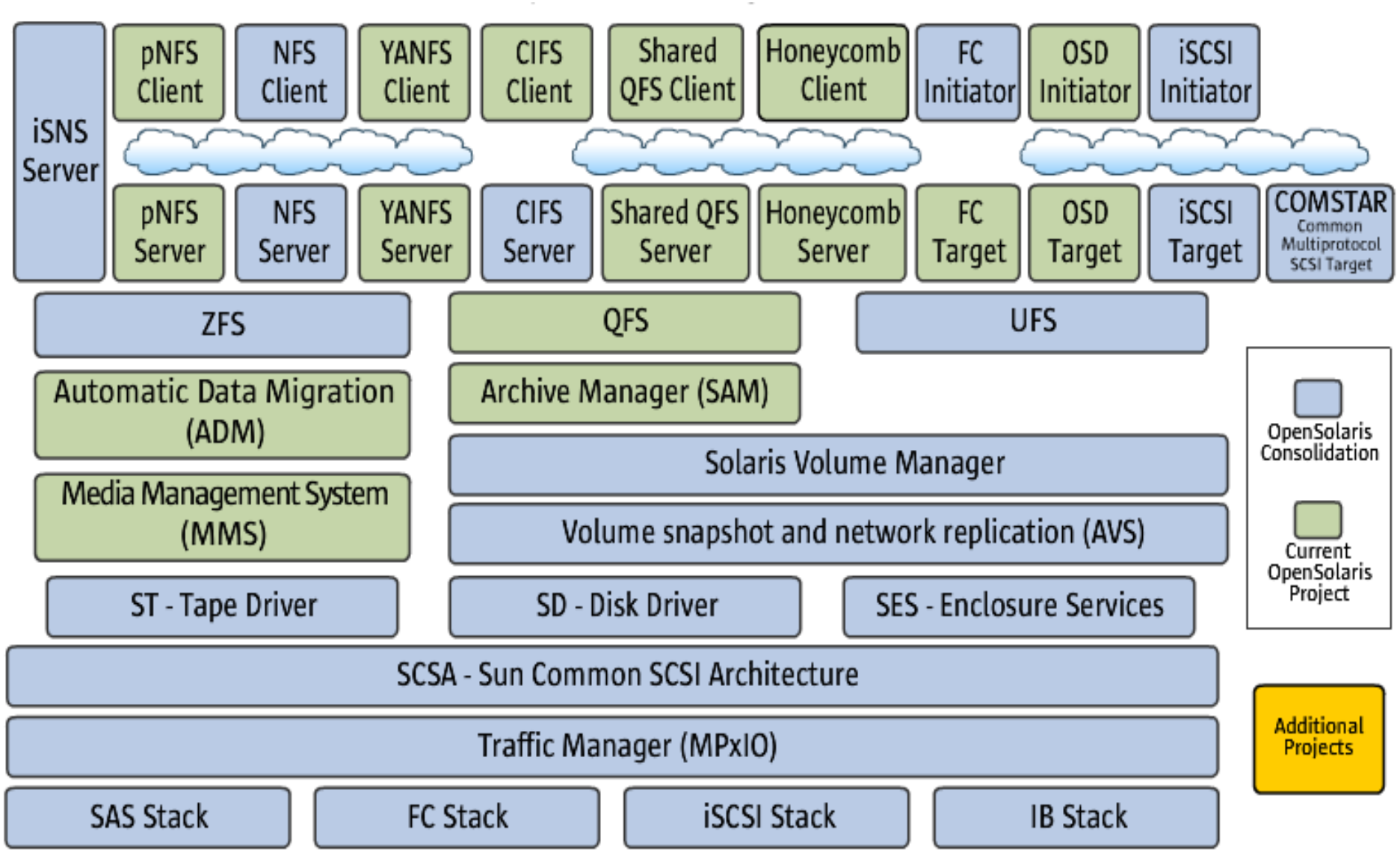
Agenda

- Introduction of OpenSolaris NAS
- Technical deep dive of OpenSolaris NAS
  - NFS Step-by-step Guide + Demo
  - CIFS Step-by-step Guide + Demo
- Getting to know iSCSI Stack
- Technical deep dive iSCSI in OpenSolaris
  - Configuring iSCSI target with ZFS
  - Configuring iSCSI initiator in Solaris 10
  - CHAP Authentication in iSCSI
  - Configuring iSCSI Initiator in Windows Vista

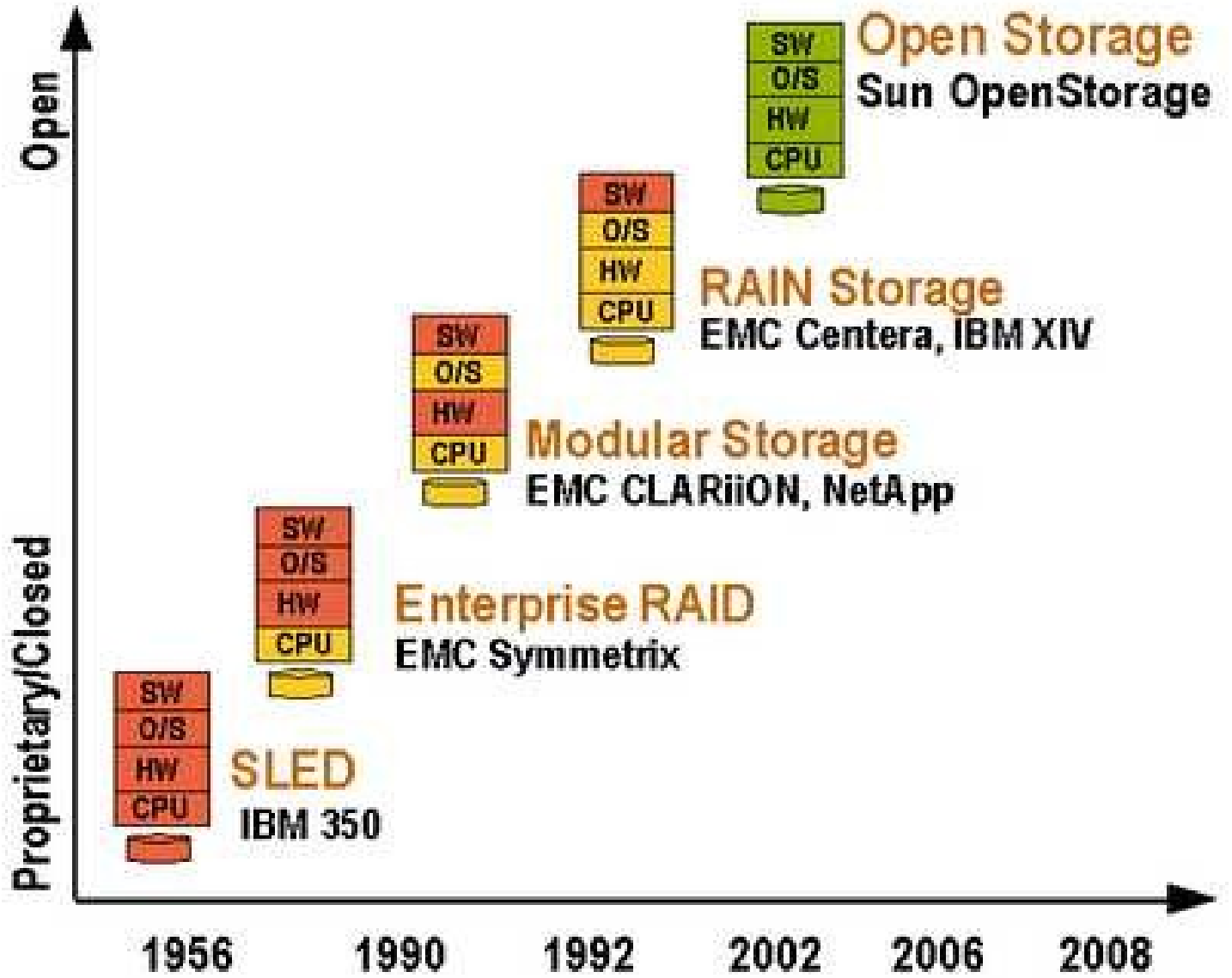
What is NAS ?

- A NAS unit is essentially a self-contained computer connected to a network, with the sole purpose of supplying file-based data storage services to other devices on the network (NAS Description)
- OpenSolaris provides functionality for NAS Unit (data storage, file system, and management to access file), including NFS, CIFS, and iSCSI.

OpenSolaris Storage Framework



Open Storage Technology Graph



What is NFS ?

- Provides transparent remote access to shared files across networks. The NFS protocol is designed to be portable across different machines, operating systems, network architectures, and transport protocols.
- Network file system protocol originally developed by Sun Microsystems in 1983, allowing a user on a client computer to access files over a network as easily as if the network devices were attached to its local disks. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. The Network File System protocol is specified in RFC 1094

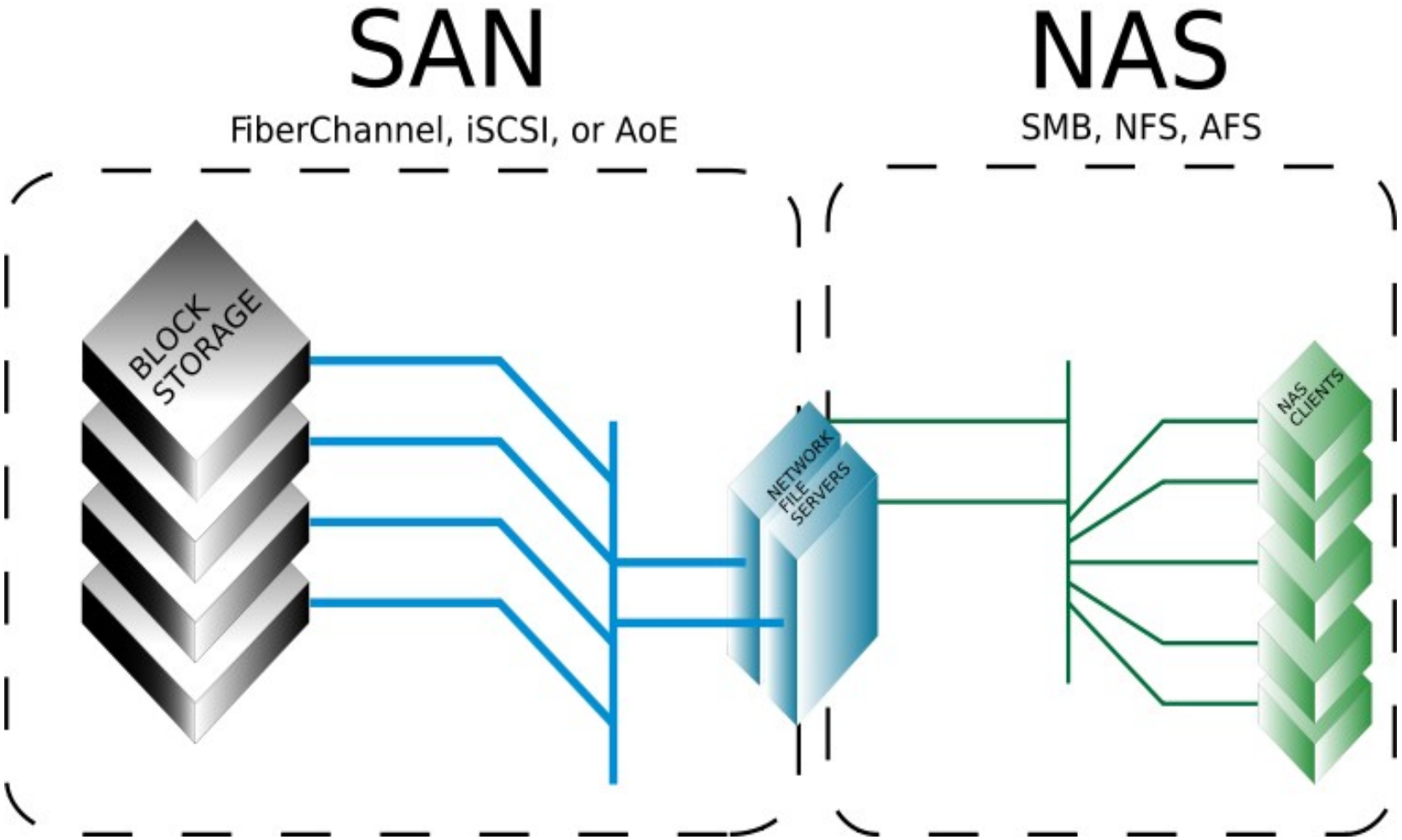
What is CIFS ?

- Provide an open cross-platform mechanism for client systems to request file and print services from server systems over a network. It is based on the standard Server Message Block (SMB) protocol widely in use by personal computers and workstations running a wide variety of operating systems
- OpenSolaris implement CIFS/SMB protocol and supporting this protocol through samba software through integrated SMF facility

NAS, Fibre Channel SAN, and iSCSI SAN

	NAS	Fibre Channel SAN	iSCSI SAN
Network	Ethernet	Fibre Channel	Ethernet
Data Transfer & Protocols	File level data transfer over industry standard network protocols (TCP/IP & IPX) via industry standard file sharing protocols (SMB, CIFS, NCP, AFP, NFS, FTP & HTTP)	Raw, block-level data requests directly to disk drive or RAID LUN using SCSI commands over Fibre Channel	Raw, block-level data requests directly to disk drive or RAID LUN using SCSI commands over IP
File System	The file system is located at the storage	The file system is located at the application server	
Data Sharing	True data sharing between heterogeneous clients because file system is at the storage side and data is transferred to client using industry standard file sharing protocols	Software required on all nodes on SAN in order to share files	
Performance	Limited by Ethernet overhead and available network bandwidth	Direct communication with disk over dedicated wire	Direct communication with disk limited by TCP/IP overhead. Can take advantage of 10Gb Ethernet
Environment	Workgroup to Enterprise	Production Workgroup to Enterprise	
Installation	Plug and Play into existing network with no additional components	Utilizes Fibre Channel infrastructure with server/software or appliance to link nodes	Utilizes existing network infrastructure with server/software or appliance to link nodes
Technology	Based on mature industry standard technologies	Interoperability issues may occur	Based on mature SCSI and Ethernet technologies

NAS and SAN Common Implementation



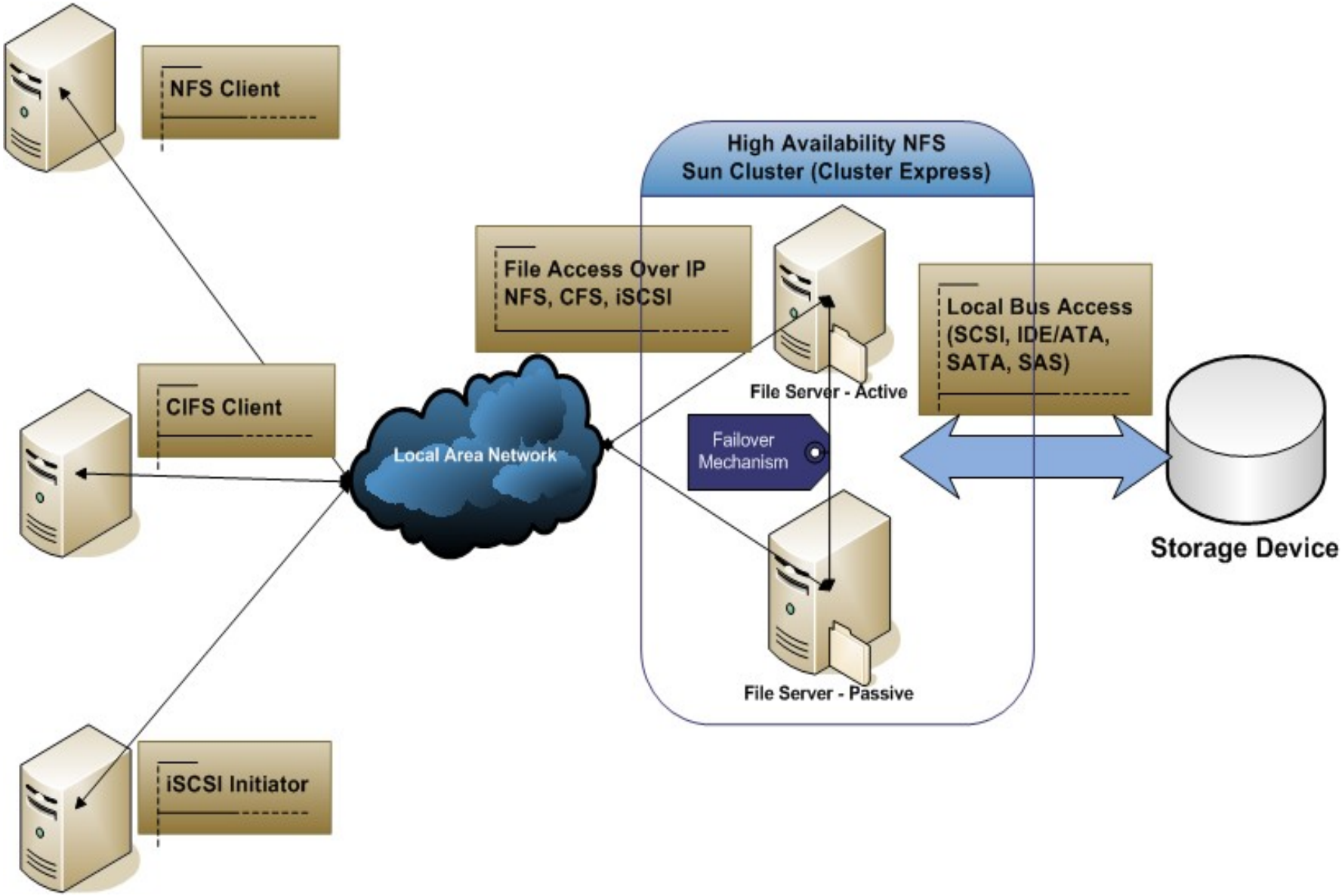
NFS Protocol Features

- RPC and security
- Procedure and operation structure
- File-system mode
  - File-handle types
  - Attribute types
  - File-system replication and migration
- OPEN and CLOSE
- File locking
- Client caching and delegation

CIFS Protocol Features

- File and Printer Access
- File and Record Locking
- Safe caching, read-ahead, and write-behind
- File change notification
- Protocol version negotiation
- Extended attributes
- Distributed replicated virtual volumes
- Server name resolution using DNS
- Batch request
- Operates over connection-oriented or connection-less transports
- Unicode file-names

NAS Implementation Example



Technical deep dive OpenSolaris NAS

NFS Environment

- **NFS Server**
  - Which contains file resources that are shared with other systems
- **NFS Client**
  - Which mounts file resources that are shared over the network and presents the resources as if they were on a local system

Technical deep dive OpenSolaris NAS

NFS Server Files

- /etc/dfs/dfstab, **lists the resources to share at boot time.**
- /etc/nfs/nfslogd.conf, **define the location of the configuration logs that are used for NFS Server**
- /etc/dfs/sharetab, **list local resources that currently being shared by NFS Server**
- /etc/rmtab, **list mounted file systems that remotely mounted by NFS clients**
- /etc/nfs/nfslog.conf, **lists information defining the location of the configuration logs used for NFS Server loging**

Technical deep dive OpenSolaris NAS

NFS Daemon

- mountd, **handles file systems mount request from remote systems and provides access control**
- nfsmapid, **is the NFS user group and mapping daemon**
- nfsd, **handle client file system request**
- statd, **works with lockd daemon to provide crash recovery and functional of lock manager**
- lockd, **support record locking on NFS files**
- nfslogd, **provide operational logging for NFSv2 and NFSv3**

Technical deep dive OpenSolaris NAS

NFS Server Step-by-step (Services)

- List the dependency of NFS server services

svcs -l svc:/network/nfs/server

- Manually enable and disable NFS server service
  - Enable NFS server service

  svcadm enable svc:/network/nfs/server
  - Disable NFS server service

  svcadm disable svc:/network/nfs/server

Technical deep dive OpenSolaris NAS

NFS Server Step-by-step (Sharing File systems)

- Configuring NFS server for sharing
  - Sharing read only file system

share -F nfs -o ro -d "Software Installer" /installer

  - Sharing read write file system with user access permission

share -F nfs -o rw=admin1:admin2 "Working Files" /works

  - List of all file system shared by NFS Server using command

dfshares

  - View /etc/dfs/sharetab for the file system being shared by NFS server

cat /etc/dfs/sharetab

Technical deep dive OpenSolaris NAS

NFS Server Step-by-step (Sharing File systems)

- Configuring NFS server for sharing to be available at boot time
  - Edit /etc/dfs/dfstab and add the following entry:

share -F nfs -o rw=admin1:admin2 -d "Software Installer" /installer

  - Reboot the Operating System
  - Run *dfshares* to check the consistency of shared files

Technical deep dive OpenSolaris NAS

NFS Server Step-by-step (Sharing ZFS through NFS)

- Create new ZFS file system
  - Run ZFS creation command

  zfs create rpool/movies

  - List the ZFS created

  zfs list
- Shares the ZFS through NFS

zfs set sharenfs=on rpool/movies
- Monitor the shared file system

dfshares

Technical deep dive OpenSolaris NAS

NFS Client Step-by-step (Mount NFS File System)

- Enable NFS client service

svcadm enable svc:/network/nfs/client

- Mount NFS server file system in client-side

mount -F nfs purple:/installer /myinstaller

- Edit /etc/vfstab to make NFS server file system available at client host boot time
  - Add the following entry at /etc/vfstab:

purple:/rpool/movies - /mymovies nfs - yes bg

  - Reboot the client host
  - Check the client mount point availability for shared file system

Technical deep dive OpenSolaris NAS

Removing NFS Configuration

- List all mounted NFS file system by client from the NFS server

cat /etc/rmtab

- Unmount NFS file system in the client side

umount /myinstaller

- Removing shared file system in the server side

unshare /installer

- Removing ZFS shares in the server side

zfs set sharenfs=off rpool/movies

Technical deep dive OpenSolaris NAS

CIFS Server Step-by-step (Prerequisite)

- Get the package information of Samba Server Software

pkginfo | grep SUNWsmb

- If the samba server package not installed, use IPS (package manager) to install the samba server software
  - Search package with SUNWsmb prefix
  - Mark necessary samba software
  - Click install/update tab for samba software installation

Technical deep dive OpenSolaris NAS

CIFS Server Step-by-step (Preparation)

- Load the smbsrv module

add_drv smbsrv

- Enable samba server service

svcadm enable -r smb/server

- Configure PAM module for samba (For Join Workgroup Configuration)
  - Edit /etc/pam.conf and add the following entry:

other password required pam_smb_passwd.so.1 nowarn

- Configure Id Mapping (For Active Directory Configuration) Create Mapping Strategy

Technical deep dive OpenSolaris NAS

CIFS Server Step-by-step (Share ZFS through CIFS)

- Join the Workgroup

smbadm join -w workgroup

- Create ZFS file system

zfs create -o casesensitivity=mixed rpool/movies

- Share the new ZFS through CIFS

zfs set sharesmb=on rpool/movies

- List the new CIFS file system share configuration

sharemgr show -vp

Technical deep dive OpenSolaris NAS

CIFS Client Step-by-step (Windows Configuration)

- Click on Network at the start menu
- Type \\<hostname/ip> example \\172.16.20.200 at the windows explorer address
- When the windows pop up with the user and password request, then apply the user and password you already prepared from the CIFS server
- Map the folder for the network drive

Technical deep dive OpenSolaris NAS

CIFS Client Step-by-step (Solaris Configuration)

- Enable the smb/client service

svcadm enable smb/client

- List shares available in the CIFS server
  - Using user and password

  smbutil view //sonny@172.16.20.200

  - Using anonymous method

  smbutil view -A //172.16.20.200

- Mount the CIFS server share folder to the client

mount -F smbfs //sonny@172.16.20.200/rpool/movies /mnt

What is iSCSI ?

- iSCSI is a protocol that allows SCSI commands to be transmitted over a TCP/IP network
- The protocol was published in draft form by the IETF in 2001, and has since been ratified and documented in RFC 3720

Why Using iSCSI ?

- Reduce costs
- Existing tools (e.g., wireshark, snoop) can be used to debug storage problems (who has the money for fibre channel analyzers?!?)
- Existing IP management and monitoring frameworks can be used with iSCSI networks

OpenSolaris Support iSCSI ?

- Yes it does !!!!
- OpenSolaris shipped along with iSCSI functionality (Includes Target service, and initiator service)
- Client visibility for iSCSI is more and more develop (ex. Windows Vista shipped initiator for iSCSI client)

iSCSI

- iSCSI is a protocol for sending SCSI commands over a TCP/IP network
- iSCSI, like SCSI, uses the term initiator to describe the endpoint that initiates SCSI operations (i.e. the client), and the term target to describe the endpoint that accepts and processes SCSI commands from one or more initiators (i.e. the server)

iSCSI Naming

- Each network element can contain one or more iSCSI nodes (e.g., initiators and targets), and each node is assigned a unique iSCSI name
- iSCSI names can be in one of two formats: IQN or EUI
- IQN names contain a date string, the domain of a naming authority, a unique string to identify the node, and are prefixed by the string "iqn."
  - IQN address: iqn.1986-03.com.sun:01:0003ba0e0795.4455571f
- EUI names consist of 16 hexadecimal digits prefixed by the string "eui." (EUI addresses resemble fibre channel WWNs)
  - EUI address: eui.02004567A425678D

iSCSI Connection and Session

- iSCSI uses the TCP protocol to ensure reliable delivery of data
- Initiators can create one or more TCP connections to a target
- Each TCP connection between an initiator and target is associated with a “session,” which is used to link logical connections together, and to ensure the ordered delivery of SCSI commands
- Initiators can create one or more sessions to a target, and each session can contain one or more TCP connections (multiple connections per session is often abbreviated MC/S)
- iSCSI sessions are made unique on an initiator by combining the node name with a unique initiator session ID (ISID), and on the target by combing the node name with the target session id (TSID)

Technical deep dive iSCSI in OpenSolaris

ISCSI Target Step-by-step (Configure Target)

- Create base directory

iscsitadm modify admin -d /iscsi

- Create ZFS Volume for backing store

zfs create -V 1g rpool/sharevol

- Create a target

iscsitadm create target -b /dev/zvol/dsk/rpool/sharevol

Technical deep dive iSCSI in OpenSolaris

ISCSI Target Step-by-step (Configure Target)

- Create an ACL list to associate backing store with node name alias
  - the target allows you to associate an alias with a node name (you can retrieve the node name of a Solaris initiator by running the iscsiadm utility with the “list” command, and “initiator-node” subcommand):

*iscsitadm create initiator -n *
iqn.1986-03.com.sun:01:e00000000000.48f0a459

- Verify the target using iscsitadm command with list sub-command

iscsitadm list target -v

Technical deep dive iSCSI in OpenSolaris

iSCSI Initiator Step-by-step (Configure Windows Initiator)

- Navigate to Control Panel Menu
- Start iSCSI initiator configuration tool
- Add the discovery ip-address and port number of the iSCSI target
- Go to Disk Management under Administrative Tools
- Format the new volume to windows NTFS file system

Technical deep dive iSCSI in OpenSolaris

iSCSI Initiator Step-by-step (Configure Solaris Initiator)

- Configure SendTarget discovery

iscsiadm modify discovery –sendtargets enable

iscsiadm add discovery-address 172.16.20.200:3260

- Configure static-config discovery

iscsiadm modify discovery –static enable

iscsiadm add static-config <target-address>,<ipaddress>:<port>

Technical deep dive iSCSI in OpenSolaris

iSCSI Initiator Step-by-step (Create New File System)

- Verify the target discover on the initiator

iscsiadm list target -v

- Prior to using newly discovered targets, the devfsadm utility needs to be run to create device entries: *devfsadm -Cv -i iscsi*
- Use format to create solaris partition
- Use newfs to create ufs file system

newfs /dev/dsk/c....s0

iSCSI Security

- This can be accomplished with a layered security approach that includes one or more of the following:
  - Dedicated storage networks
  - Client ACLs
  - Auditing
  - IPSEC and header digests
  - Port security on Ethernet switches

iSCSI Performance

- iSCSI performance can be quite good, especially if you follow a few basic rules
  - Use Enterprise class NICs (they make a HUGE difference)
  - Enable jumbo frames on storage ports
  - Use layer-2 link aggregation and IPMP to boost throughput
  - Ensure that you are using the performance guidance listed in bug #6457694 on opensolaris.org
  - Increase send and receive buffers, disable the nagle
  - algorithm and make sure TCP window scaling is working correctly

iSCSI Performance

- Ttcp and netperf are awesome tools for benchmarking network throughput, and measuring the impact of a given network tunable

Conclusion

- OpenSolaris provides various NAS technology including NFS, CIFS, and iSCSI functionality and its free (Open Source)
- NAS uses standard protocol technology whereas easier to control, and manage
- NAS provides simple and cost-effective way to achieve fast data access for multiple clients at the file level

Reference

- OpenSolaris Storage Community

http://opensolaris.org/os/community/storage/

- Introducing NFS fundamental on Solaris

http://www.sun.com/bigadmin/content/submitted/fundamentals_nfs.jsp

- Solaris NFS Environment

http://docs.sun.com/app/docs/doc/806-0916

- Configuring OpenSolaris CIFS

http://blogs.sun.com/timthomas/entry/configuring_the_opensolaris_cifs_server

- Solaris CIFS Administration Guide

http://docs.sun.com/app/docs/doc/820-2429

- Configuring Solaris iSCSI target and initiator

http://docs.sun.com/app/docs/doc/817-5093

- Getting to know iSCSI stack

http://prefetch.net/articles/solarisiscsi.html

開
放
的
열린
مفتوح
libre
मुक्त
ముక్త
livre
libero
ಮುಕ್ತ
开放的
açık
open
nyílt
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

Thank you!
“Practice makes you perfect”

Sonny Hastomo
IT Architect
sonny@sun.co.id
http://epuzzle.wordpress.com

“open” artwork and icons by chandan:
http://blogs.sun.com/chandan