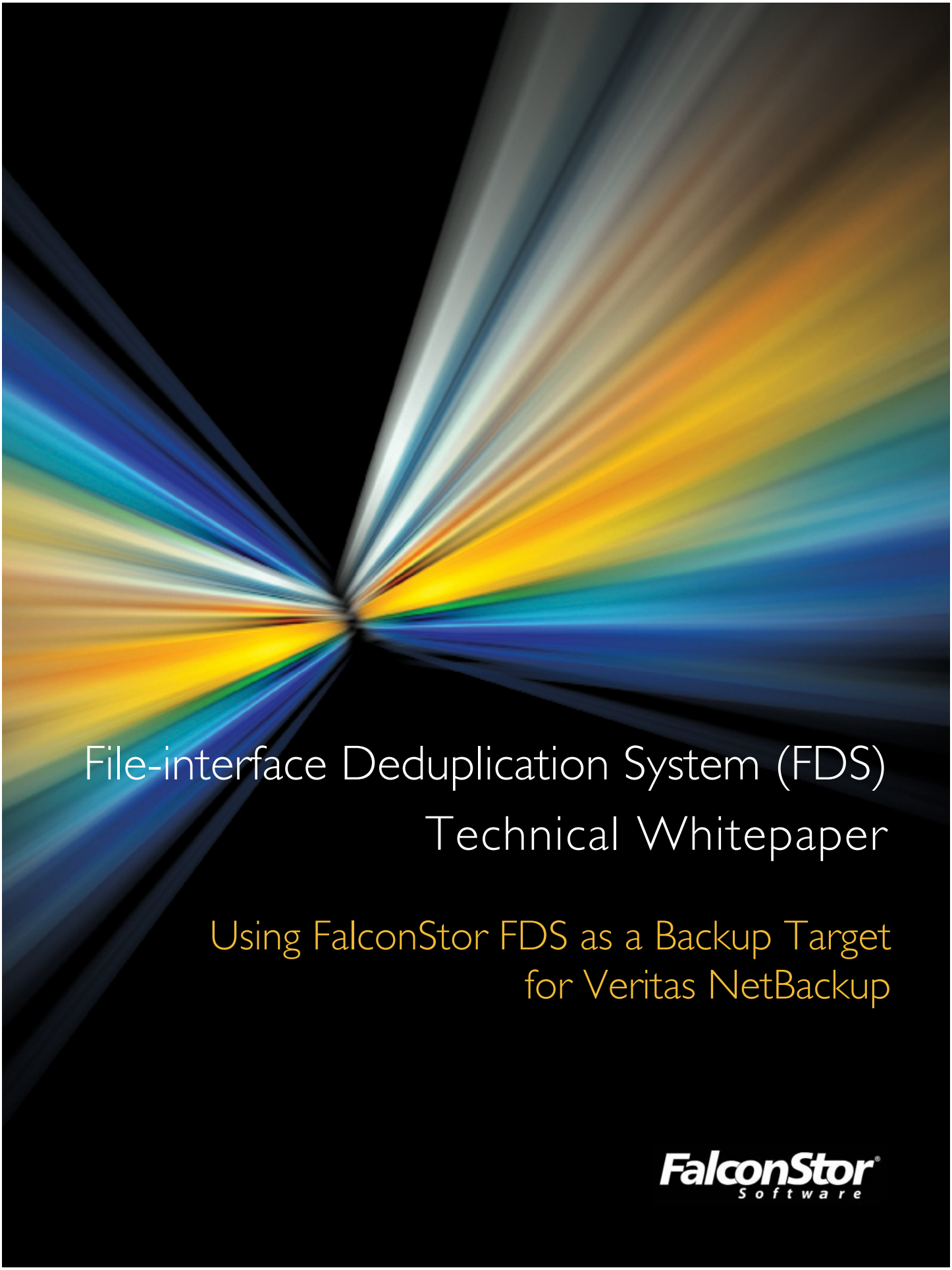


File-interface Deduplication System (FDS) Technical Whitepaper

Using FalconStor FDS as a Backup Target for Veritas NetBackup

FalconStor® Software

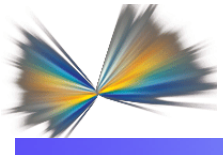
Using FalconStor FDS as a Backup Target for Veritas NetBackup

File-interface Deduplication System (FDS) Technical Whitepaper

FalconStor Software, Inc.
2 Huntington Quadrangle, Suite 2S01
Melville, NY 11747
Phone: 631-777-5188
Fax: 631-501-7633
Website: www.falconstor.com

8.27.2009

Contents



Introduction

Abstract

The FalconStor® File-interface Deduplication System (FDS) is a block-level data deduplication tool that provides a space-efficient repository for data from backup systems. With FalconStor FDS, you can reduce your backup management costs by dramatically reducing your disk storage needs, reducing your dependency on tape, and reducing off site tape storage costs by enabling you to achieve longer retention periods on disk and facilitating replication of your backups to meet off-site requirements.

Document scope

This document describes the basic concepts and integration guidelines for FalconStor FDS in a Veritas NetBackup™ environment. The document provides an architectural overview of FalconStor FDS when used with Veritas NetBackup and explains the benefits of a combined solution. It is intended to provide best practices for configuring all components. The information in this document is presented in the form of guidelines. This document is not meant to be a technical Best Practices Guide.

Audience

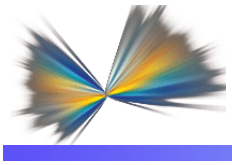
The audience for this document includes storage consultants, pre-sales specialists in charge of projects involving backup optimization concepts, and partners interested in FalconStor FDS. This document is especially beneficial for IT directors, storage administrators, backup administrators, data center managers, architects, and others involved in the administration of backup architecture including Veritas NetBackup. This document can also be valuable to IT staff in charge of disaster recovery (DR) projects.

Assumptions

We assume that the reader is familiar with:

- Veritas NetBackup
- Operating systems
- Network-attached storage and protocols (i.e., NFS, CIFS)
- LAN-based data protection
- Backup challenges
- Deduplication (refer to the FalconStor whitepaper *Demystifying Data Deduplication: Choosing the Best Solution* for an introduction to deduplication).

We also assume that this may be the reader's first exposure to FalconStor FDS, so we are including the basics of FalconStor FDS.



Overview and Benefits

In today's business environment, many customers face increased challenges in protecting their vital data from loss, theft, corruption, and disaster. Traditional backup operations constantly reproduce data for protection and recovery purposes, therefore the amount of data keeps increasing and IT costs keep rising. Even though disk prices are lower each year and tape drive and SAN performance have increased, coping with the exponential data growth remains a significant challenge for most organizations.

With the introduction of FalconStor FDS, it is now possible to control data growth resulting from producing multiple copies of the same backup data. FalconStor FDS is a block-level data deduplication tool that provides a space-efficient repository for data from:

- Third-party tape backup software, such as: EMC NetWorker, IBM Tivoli Storage Manager (TSM), Veritas NetBackup, Symantec Backup Exec, CA ARCserve, Arkeia Network Backup, and VMware Consolidated Backup.
- Database backup utilities, such as: Oracle RMAN and SQL-BackTrack.
- Archiving applications, such as: Mimosa™ Systems NearPoint™, Arkivio® auto-stor, CommVault DataArchiver™, FalconStor Capacity-on-Demand™, and Enigma Data Solutions' SmartMove.
- Any other mechanism for delivering data to a network share, such as FalconStor FileSafe™.

With FalconStor FDS, you can reduce your disk storage needs dramatically, allowing you to maintain far more data on disk while incurring minimal additional storage costs. FalconStor FDS can also function as a nearline data repository for project archives, storing older files, etc.

FalconStor FDS supports many-to-one data replication, providing a cost-effective disaster recovery solution. Only deduplicated data is sent over the WAN, providing bandwidth savings. Smaller offices and remote sites can eliminate tape backup entirely using the FDS repository. Data restore is quick and efficient from native-format files rather than from tape-backup formats.

FalconStor FDS uses standard network protocols such as Common Internet File System (CIFS) or Network File System (NFS) to present a simple, network-based file share as the target for backed up data. Connection to FalconStor FDS is a simple matter of mapping to a share, making it compatible with any application that uses an IP network to store data.

Each FDS file share holds incoming data, acting as a "disk" for disk-to-disk (D2D) backup. Based on user policy, deduplication occurs at a scheduled time or on an as-needed basis.

During deduplication, the system analyzes blocks of data and determines whether the data is unique or has already been copied to the FDS repository (virtualized disks that hold deduplicated data). The process then passes only single instances of unique data to the FDS repository and replaces each deduplicated file with a small file (called a *stub* file), whose function is to point to the repository and is used to retrieve stored data.

Even though the user interface is file-based, deduplication is done at the block-level, not at a file level. Block-level deduplication examines small sub-blocks, making it far more effective at reducing storage consumption than file-based deduplication.

Because it uses network-based file shares for backed up data, restoring data is faster and easier with FalconStor FDS. The administrator has direct access to all files without the need for

a restore job. Even after deduplication occurs, pointers (*stub* files) on the share point to the full file in the repository. Restoring is as simple as copying the necessary files from a share back to the appropriate location.

FalconStor FDS Benefits

Easy Deployment: FalconStor FDS is qualified to seamlessly work with Veritas NetBackup by presenting a file interface or a CIFS or NFS network share. This ease of integration allows for seamless deployment into the existing nearline storage infrastructure that doesn't require any changes to current backup and archiving processes.

High-performance backup: FalconStor FDS was built with performance in mind. Its post processing and concurrent block-level deduplication technology are optimized to ingest backup data without affecting backup speed. Its concurrent processing options allow the deduplication process to take place in the background while its file interface maintains the high performance characteristics needed to meet the backup window.

Flexible deduplication: Data deduplication is policy driven; the deduplication processes can be set by the user to start immediately after the backup or can be scheduled to occur at a set time on a regular basis. This flexibility allows the end user to accommodate different operations on non-duplicated data such as data copies, restore operations, or other operations such as data mining or database testing.

High-performance restore: FalconStor FDS is optimized to enable high-performance data access for both non-deduplicated data as well as deduplicated data. This allows for quick backup data restore processes when needed. Data is striped across the deduplication repository to maximize read operation performance. In addition, the data deduplication repository has direct block-level access with no file system overhead, resulting in no performance degradation during read operations.

Flexible, scalable architecture: FalconStor FDS can scale from a small footprint deployment up to petabytes of logical storage capacity. Its physical managed capacity can scale from 1 TB up to 64 TB of deduplication repository in a single node.

Multi-site Disaster Recovery: FalconStor FDS offers global deduplication capability for quick and cost-effective disaster recovery deployments. Connecting remote offices via FDS appliances allows organizations to eliminate tape shipments between sites and ensures that data is readily available online when needed. FalconStor FDS is enabled with intelligent global data replication technology; unique data is sent only once from remote sites to the main data center. This WAN-optimization method allows for cost effective data replication and significant bandwidth savings; up to 97% reduction in production bandwidth usage.

Terminology

The primary components of the FalconStor FDS solution are the FDS appliance, FDS clients, and the console. These components all sit on the same network segment, the *storage network*. The terminology and concepts used in FalconStor FDS are described here. For additional information, refer to the *FalconStor FDS User Guide.*

Component	Definition
Appliance	An industry-standard server that provides all data deduplication functions. The appliance can function as a standalone appliance with internal storage or it can function as a gateway to storage on an existing network. FDS storage is used to store the original data as well as the unique data blocks and the indexes to the data. The FDS appliance can be attached to physical SCSI and/or Fibre Channel storage devices.
Clients	FDS clients are NetWorker storage nodes that use an FDS share to store data. Storage resources appear to client operating systems (Windows, Linux, Solaris, etc.) as network-attached devices.
Shares	The logical entities presented to FDS clients via the IP network. Clients access FDS shares using either the NFS or CIFS network protocol.
Console	The administrative tool that allows you to create shares, configure deduplication, and monitor resources and deduplication. This Java application can be run on any Windows machine or Linux platform that supports the Java 1.5 Runtime Environment (JRE).
Physical Resources	The actual physical LUNs used to create logical resources, as seen by the RAID controller/storage HBA within the FDS appliance. Clients do not have access to physical resources.

Logical Resources	These are all of the logical/virtual resources defined on the FDS server.
	Logical resources consist of sets of storage blocks from one or more physical hard disk drives. This allows the creation of logical resources that contain a portion of a larger physical disk device or an aggregation of multiple physical disk devices. For example, the logical resources listed below can all be created from a single large physical device.
	● FDS Resources: The staging area for the files.
	● Repository Resources: Virtualized disks configured as storage (data disks, index disks, and folder disks) for deduplicated data.
	  • Data repository disks: Where the unique data blocks are stored.
	  • Repository index and folders: Where the metadata data is stored.

FalconStor FDS Architecture

The typical FalconStor FDS architecture is made of three major components. Use Figure 1 as a reference.

- The first component (the front end) communicates to the clients (i.e., database and backup servers) via Common Internet File System (CIFS) and/or Network File System (NFS) protocols.
- The second component (the FDS appliance itself) performs deduplication functions and handles replication between FDS appliances.
- The third component (the back-end) is the disk used by the FDS appliance to store the deduplicated backup data.

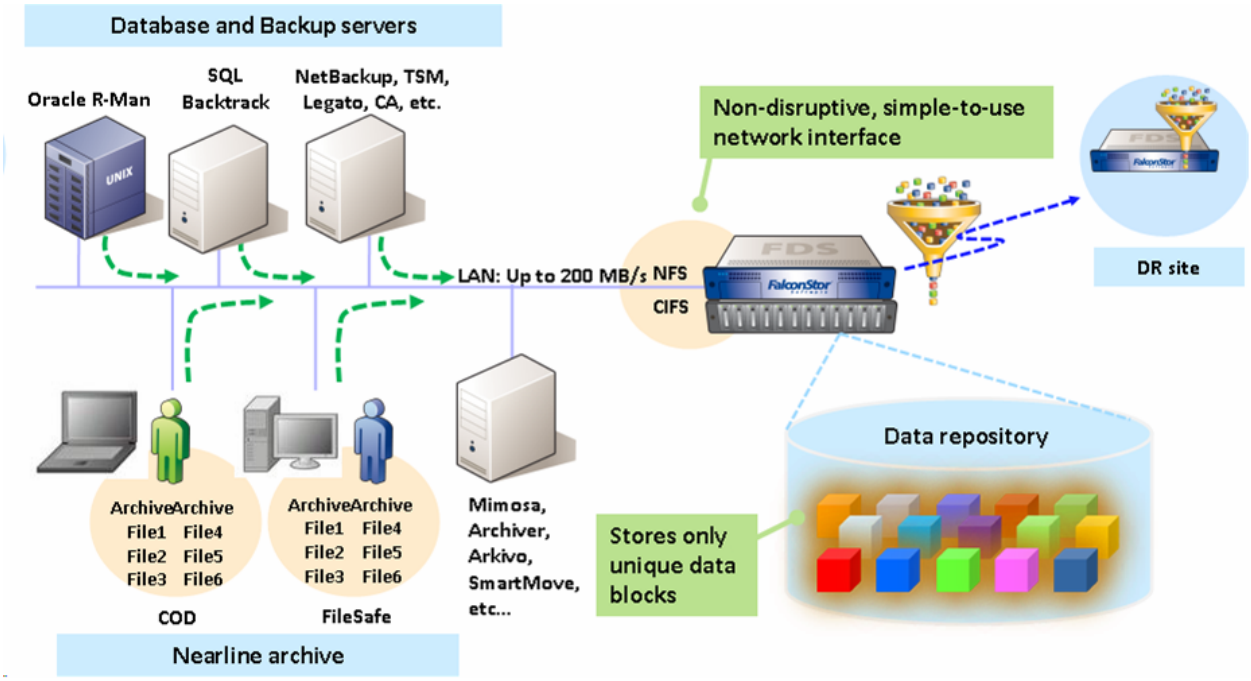


Figure 1. Typical FDS Architecture

Depending on the FalconStor FDS licensing and packaging, the storage can be embedded or not; if so, the second and third components are integrated together. The following configurations are available:

- **Virtual Appliances**: Small footprint, ideal for small environments without demanding performance requirements, such as remote offices.
- **Physical Appliances**: Provide easy-to-deploy and easy-to-manage, self-contained deduplication repository.
- **Gateway Appliances**: Integrate with existing storage infrastructure to provide storage capacity optimization over existing resources.

Veritas NetBackup

Overview

Veritas NetBackup is a high-performance data protection application. Its architecture is designed for large and complex distributed computing environments. NetBackup provides scalable storage servers (master and media servers) that can be configured for network backup, recovery, archiving, and file migration services.

Typical NetBackup architecture

The NetBackup administrator can allow users to back up, restore, or archive the files from their computers, as shown in the diagram below.

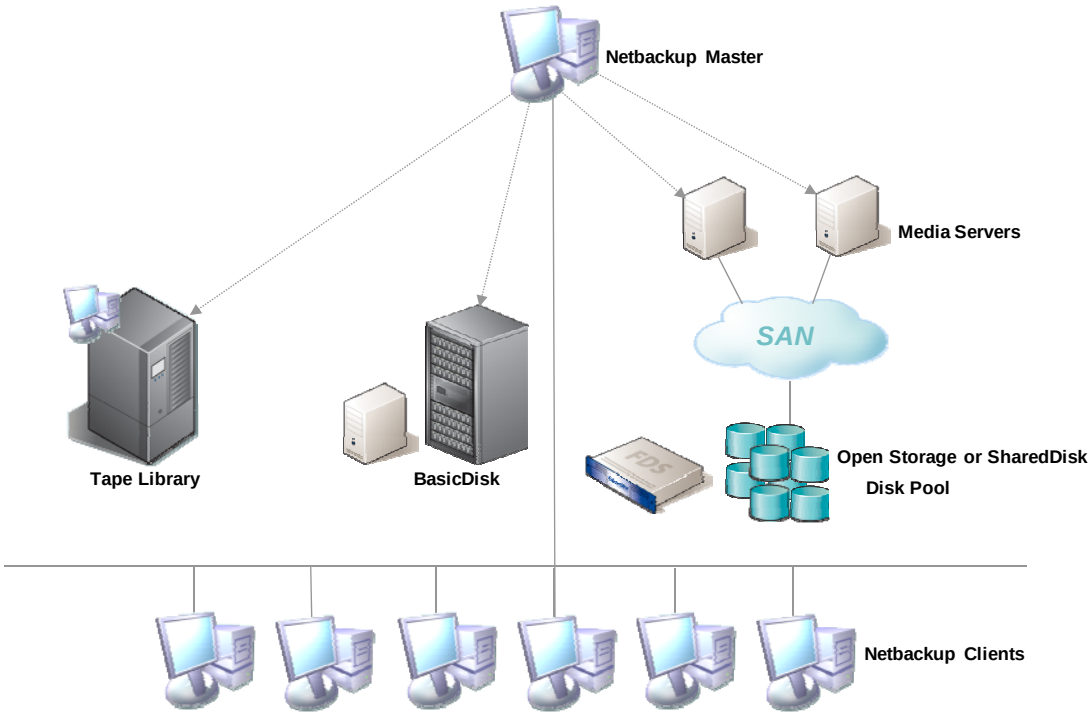


Figure 2. Typical NetBackup configuration

How does NetBackup write to disk?

Overview

Backing up data with NetBackup to an FDS appliance follows the same rules and procedures as standard backup to disk using file shares. Given that fact, it is very important for FDS users to correctly understand how NetBackup is configured for backup to disk.

NetBackup 6.5 provides a new model for disk storage called the *Flexible Disk Option*. This model provides new logical entities, uses existing entities to provide new capabilities, and introduces new terminology. All disk types (collectively known as the *Enterprise Disk Options*) except *BasicDisk* use the new model for disk storage.

Three storage models are currently available:

- **Data mover**: An entity that moves data between the primary storage (the NetBackup client) and the storage server. NetBackup media servers function as data movers. Depending on the selected Enterprise Disk Option, a NetBackup media server may also function as a storage server.
- **Storage server**: An entity that writes data to and reads data from the disk storage. A storage server is the entity that mounts a file system on the storage. Depending on the Enterprise Disk Option, the storage server is either a host that is part of a storage appliance or filer, or it is a NetBackup media server.
- **Disk pool**: A collection of disk volumes that are administered as an entity. In NetBackup, a disk pool is a storage type that aggregates disk volumes into a pool of storage that you can use for backups. When you create a storage unit, you select the disk type and then you select a specific disk pool.

The disk types available under the *Flexible Disk Option* (*AdvancedDisk* and *SharedDisk*) allow NetBackup to fully utilize file systems native to the host operating system of the media server. These two options complement the *BasicDisk* option. The *Advanced Disk* selection is available only when the *Flexible Disk Option* is licensed.

NetBackup assumes full ownership of these file systems and at the same time uses the storage server capabilities of the host operating system.

The difference between the two disk types is that *AdvancedDisk* does not require any specialized hardware, while *SharedDisk* depends on the availability of SAN attached storage. Both disk types are managed as *disk pools* within NetBackup.

The next chapters will focus on performance issues and best practices related to the *AdvancedDisk* option, as it is the most appropriate target for FalconStor FDS.

In practice, the entire data path between client and storage, including both hardware and software stacks, determines the overall performance of the backup and restore process. It is therefore essential that the performance of the disk storage is not considered independently of the entire data path and the effect of infrastructure when seeking to resolve overall performance issues.

BasicDisk versus AdvancedDisk

A QUICK INTRODUCTION

A *BasicDisk* storage unit consists of a directory on locally-attached disk or network-attached disk that is exposed as a file system to a NetBackup media server. NetBackup stores backup data in the specified directory.

- A volume or file system cannot be included in multiple *BasicDisk* storage units.
- *BasicDisk* storage units cannot be used in a storage lifecycle policy.

An *AdvancedDisk* storage unit is used as a dedicated disk that is directly attached to a NetBackup media server.

NetBackup assumes exclusive ownership of the disk resources that comprise an *AdvancedDisk* disk pool. If the resources are shared with other users, NetBackup cannot manage disk pool capacity or storage lifecycle policies correctly.

For *AdvancedDisk*, the NetBackup media server functions as both a data mover and a storage server. NetBackup does not support the use of CIFS disk volumes with Windows Media Servers because the mapped devices are not visible to the Windows services and thus cannot be discovered by NetBackup (`nbdevconfig` command).

ADVANCEDDISK BENEFITS

AdvancedDisk can be used with any disk type and offers efficiency improvements and other benefits over the traditional *BasicDisk* model, including the following:

- Disk pools, which simplify expansion and improve space usage
- The ability to use Storage Lifecycle Policies
- The ability to use Storage Unit Groups with media server load balancing
- The ability to use common storage “shared” across multiple media servers for load balancing and redundancy

Given the fact that FalconStor FDS is mainly designed for consolidating the backup infrastructure, the *AdvancedDisk* option is a must. The next chapter covers the *AdvancedDisk* option.

DISK POOLS

Disk pools form one of the key concepts underlying the *Flexible Disk Option*. Disk pools significantly change the way in which available disk space is used. A disk pool groups a set of disks together to form a single block of storage that can be shared among multiple Storage Units and, in some cases, multiple media servers.

With the *Flexible Disk Option* disk types of *AdvancedDisk* and *SharedDisk*, the disk pool provides a pool of storage for use by the Storage Units, replacing the more conventional one-to-one mapping between disk and Storage Unit provided by the *BasicDisk* model. In *BasicDisk* configurations, each Storage Unit has access to a single disk volume or part thereof. With the *Flexible Disk Option*, the Storage Units can access all of the disks in a disk pool and the disk used for a particular backup is selected based on the amount of space available. In effect, the entire disk pool appears as a single disk to Storage Units.

IMPLEMENTING ADVANCEDDISK IN A FALCONSTOR FDS CONTEXT

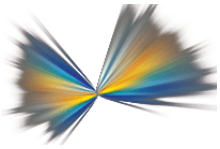
Before creating a disk pool, the NetBackup administrator must determine whether the NetBackup server is already configured as a storage server by entering the following command:

```
/usr/openv/NetBackup/bin/admincmd/nbdevquer y -liststs
```

To configure an *AdvancedDisk* in NetBackup, the administrator enters the following command:

```
nbdevconfig -creatests -storage_server storage_server -stype
AdvancedDisk [-st storage_type] -media_server media_server
```

The disk pool can be created directly via the NetBackup console. The volume name(s) point to the FDS shares.



Integrating FalconStor FDS with Veritas NetBackup

Methodology

FalconStor FDS deployment is a very straightforward process. If the appliance is built with internal storage, there are no particular configuration steps to follow, except to create and map FDS shares to the NetBackup hosts. If the FDS appliance is connected to external storage, volume groups and LUNs must be created and mapped to the FDS appliance.

If the FDS appliance is to be integrated into a Windows domain, it must be added to the domain by specifying the name of the Domain Controller. If no Domain Controller is present, the FDS appliance can be installed using shared mode authentication.

Once the shares are mapped to the NetBackup host, NetBackup configuration can start. Integrating FalconStor FDS into a NetBackup environment is a ten-minute process that requires only IP connections to the customer IP network.

Optimizing the environment

When implementing FalconStor FDS in a NetBackup environment, a variety of factors that can influence the overall FDS/NetBackup architecture must be considered:

- Operating system configuration including network interface cards (NICs) and mount points
- The IP infrastructure
- The FDS appliance itself, including configuration of virtual and physical resources. Deduplication is part of the FalconStor FDS implementation process. FalconStor FDS deduplication configuration is another factor that can influence the overall architecture.
- NetBackup server configuration including:
  - Backup policies
  - Disk pool definition
  - Client to server communication
- Communication between the backup servers and the FDS system

Multi-NIC support

When considering the number of IP paths between the FDS appliance and the media server, pay attention to the fact that there are obvious limits to the theoretical exponential formula “the more paths there are, the higher the bandwidth will be”. These limits are linked to the number (and the characteristics) of available ports, especially on the server side.

Avoiding bottlenecks between the backup servers and the FDS system is quite simple: the data transfer rate increases with the number of available channels. A server usually has PCI slots of different speeds. If possible, choose a fast PCI bus for data transfer. The speed of a NIC is heavily dependent on the PCI bus. The only way to efficiently multiply communication paths is to use the maximum number of PCI slots.

In addition, implement IP bonding in order to take advantage of the full speed of the NIC.

Multiple sites with replication

Implementing FDS replication in a NetBackup environment is a simple process, since replication is a transparent process. As a general rule, we recommend preparing the replicated mount point so that the FDS shares that will contain the replicated data can be easily detected by NetBackup. When a restore is needed, it can be done by mounting the replicated share to the NetBackup host. In this context, the recommendations made in the next sections must be applied to both local and remote FDS appliances.

FDS as a primary storage pool or destination storage unit?

There are several ways to approach implementing FalconStor FDS into a NetBackup architecture.

The first method is to use FalconStor FDS as a primary storage pool in the NetBackup hierarchy. In this scenario, the FDS shares that are presented to the NetBackup media server will be used as a primary pool. Backup policies will be configured so that the backup jobs go primarily to the FDS shares. In a migration scenario, it will be necessary to redirect the existing backup policies to this new disk pool and migrate the existing backup to this new storage unit.

The second method is to use the FDS shares as a destination storage unit (DSU). In this case, the FDS shares are used as a secondary storage pool that will contain the result of NetBackup deduplication. In some cases it is beneficial to continue using a tape drive as a primary pool because tape drives are correctly fed. In this particular scenario, it is not necessary to migrate the existing backup images, since FalconStor FDS will be used to archive the secondary copies on disk.

The introduction of FalconStor OpenStorage Option (OST) in the next FalconStor FDS release will affect the way that FalconStor FDS is configured in a NetBackup environment. Specific best practices will be provided at that time.

Best Practices

This section presents the methods for optimizing each system component, from a technical perspective. These include the operating system, the communication channel, the NetBackup product, network performance, and FalconStor FDS configuration. The guidelines below assume that system administrators already know how to configure these components.

Operating system-related tuning factors

NetBackup master server tuning is recommended for new FDS system implementations using NFS/CIFS and IP protocols.

NICs

In order to optimize FDS system performance with NetBackup using NFS/CIFS and IP protocols, the following parameters must be considered:

AIX

Before AIX 4.3.3 and AIX 5.1, AIX provided a single set of system-wide values for the key IP interface network tuning parameters, making it impossible to tune a system that has widely differing network adapter interfaces. Beginning with AIX 4.3.3 and AIX 5.1, Interface Specific Network Options (ISNO) allow system administrators to tune each TCP/IP interface individually for best performance.

There are five ISNO parameters for each supported interface:
rfc1323
tcp_nodelay
tcp_sendspace
tcp_recvspace
tcp_mssdflt

When set, the values for these parameters override the system-wide parameters of the same names that had been set with the *no* command. When ISNO options are not set for a particular interface, system-wide options are used. When options have been set by an application for a particular socket using the *setsockopt* subroutine, such options override the ISNOs. FalconStor recommends setting the *tcp_recvspace* and *tcp_sendspace* values to 64K and the *tcp_nodelay* to 1.

Refer to IBM documentation for additional information about setting parameters: (http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/com.ibm.aix.commadmn/doc/commadmndita/interfaces_options.htm)

Solaris

Modify the default receive/send window size. FalconStor recommends setting these values to 256K.

Refer to the following SUN tuning guide for information on modifying parameters: http://dlc.sun.com/pdf/817-0404/817-0404.pdf .

In order to speed up NFS transfers, you can modify the following parameters:

nfs3_max_threads - This parameter controls the number of kernel threads that perform asynchronous I/O for NFS version 3 clients and affects NFS throughput. It must be set to 16.

nfs3_async_clusters - This parameter controls the mix of asynchronous requests that are generated by NFS version 3 clients. It must be set to 8.

HP-UX

The same tuning must be applied as for Solaris and the default receive/send window size must be modified as per HP recommendations. FalconStor recommends setting these values to 256K.

Linux

For Linux, FalconStor recommends the following settings:

- Set min, default, and max TCP transmit window sizes to 4K, 256K, and 1M, respectively.
- Set min, default, and max TCP receive window sizes to 4K, 256K, and 1M, respectively.

On the 2.2 and 2.4 kernels, the socket input queue (where requests sit while they are being processed) has a small default size limit of 64k. This means that if you are running eight instances of `nfsd`, each will only have 8k to store requests while it processes them. FalconStor recommends increasing this number to at least 256k for `nfsd`.

Windows

Configure the TCP window size and the TCP max size within Windows to 64K.

MOUNT OPTIONS

Certain parameters can speed up access to CIFS/NFS shares. FalconStor recommends the following NFS configuration settings for mounting a FDS share to a NETBACKUP master / media server:

Platform	NFS Configuration
AIX 5.3	`nfso -o nfs_use_reserved_ports=1` `mount –v nfs –o proto=tcp,vers=3,intr,hard,llock, combehind,rsize=65536, wsize=65536, FDS@IP:/FDS_path /FDS_Mount_Point`
Solaris 10	`mount -F nfs -o hard,llock,intr,vers=3,proto=tcp,rsize=65536, wsize=65536 FDS@IP:/FDS_path /FDS_Mount_Point`
HP-UX 11v3	`mount –F nfs –o rsize=65536,wsize=65536,hard FDS@IP:/FDS_path /FDS_Mount_Point`
Linux	`mount -t nfs -o hard,nolock,intr,nfsvers=3,tcp,rsize=65536,wsize=65536,bg FDS@IP:/FDS_path /FDS_Mount_Point`
Windows	FDS shares can be configured in Active Directory or shared mode.

NetBackup server configuration guidelines

DISK POOLS

Implementing disk pools in an FDS context is essential for aggregating FalconStor FDS in a single NetBackup instance. When configuring a disk pool or *BasicDisk* in an FDS environment, pay close attention to the fragment size, which affects where tape markers are placed and how many tape markers are used. (The default fragment size is 512 GB for disk.) As a rule, a larger fragment size results in faster backups but may result in slower restores when recovering a small number of individual files. Moreover, it is also important to keep in mind that NetBackup creates a new fragment at each checkpoint when the Checkpoint restart option is enabled.

For fragment size, consider the following rules:

- Larger fragment sizes usually favor backup performance, especially when backing up large amounts of data. Smaller fragments can slow down large backups, since the backup stream is interrupted each time a new fragment is created.
- Larger fragment sizes do not hinder performance when restoring large amounts of data. But when restoring a few individual files, larger fragments may slow down the restore.
- Larger fragment sizes do not hinder performance when restoring from non-multiplexed backups. For multiplexed backups, larger fragments may slow down the restore. In multiplexed backups, blocks from several images can be mixed together within a single fragment. During restore, NetBackup positions to the nearest fragment and starts reading the data from there, until it comes to the desired file. Splitting multiplexed backups into smaller fragments can improve restore performance.
- During restores, newer, faster devices can handle large fragments well. Slower devices, especially if they do not use fast-locate block positioning, restore individual files faster if

fragment size is smaller. (In some cases, SCSI fast-tape positioning can improve restore performance.)

Given these assumptions, we recommend setting fragment size to 512 GB. If backup performance is as expected, you can decrease this value to optimize performance when restoring from FDS shares. This best practice is particular true if the checkpoint option is not enabled.

NETBACKUP CATALOG AND BACKUP POLICIES

Backing up data on FalconStor FDS increases the number of backup images in the NetBackup catalog. In addition, deduplicating backup images allows NetBackup to store more and more images. This will increase the number of entries in the NetBackup catalog. Indexing the catalog for faster access to backups and backup policies is strongly recommended. Moreover, we recommend always using a scheduled Hot Catalog Backup to secure the NetBackup catalog.

As a best practice, we recommend the usage of synthetic backups. Synthetic full backups are faster when incremental backups are stored on FalconStor FDS. In this context, FDS systems facilitate NetBackup synthetic full backups, the goal of which is to create a full backup image from existing backup data.

In addition, it is also very important to keep in mind that NetBackup is also able to create Synthetic Cumulative backup images from a set of cumulative or differential images. This allows an administrator to recover from one or two images, greatly simplifying restore operations. Using this option is a must when using FalconStor FDS.

As a last best practice, we recommend regular review of the disaster recovery plan in order to verify that this plan continues to be operational.

MULTIPLEXING/COMPRESSION AT THE CLIENT LEVEL

Only in rare cases is it beneficial to use client (software) compression. Those cases usually include the following characteristics:

- The client data is highly compressible
- The client has abundant CPU resources
- You need to minimize the data that is sent across the network between the client and server

In other cases, NetBackup client compression should be turned off and the hardware should handle the compression.

In a FalconStor FDS context, we recommend turning NetBackup compression off in order to optimize the deduplication ratio.

In addition, multiplexing is not recommended as it can significantly impact the deduplication ratio. As an alternate solution, we recommend creating additional FDS shares, which will allow NetBackup to handle multiple data streams without affecting the NetBackup client. Streaming from several FDS shares permits backups to take advantage of *Read Ahead* on an FDS share or set of FDS shares.

NetBackup tuning: Network performance

GENERAL GUIDELINES

Before starting a FalconStor FDS implementation, the administrator of the storage infrastructure must pay attention to the network interface cards:

- All NICs for NetBackup servers and clients must be set to full-duplex.
- Both ends of each network cable (the NIC card and the switch) must be set identically as to speed and mode. (Both NIC and switch must be at full-duplex.) Otherwise, link down, excessive or late collisions, and errors can result.
- If auto-negotiate is used, make sure that both ends of the connection are set with the same mode and speed.
- In addition to NICs and switches, all routers must be set to full-duplex.
- Using AUTOSENSE may cause network problems and performance issues.
- Consult the operating system documentation for instructions on how to determine and change NIC settings.

Regarding the network load, consider:

- The amount of network traffic
- The amount of time that network traffic is high

Small bursts of high network traffic for short durations can decrease the data throughput rate. However, if network traffic remains high, the network is probably the bottleneck. Try to schedule backups when network traffic is low. If your network is loaded, you may want to implement a secondary network which is dedicated to backup and restore traffic.

MEDIA SERVER BUFFER SIZES (NETWORK)

The NetBackup media server has a tunable parameter that you can use to adjust the size of FDS shares. This share either receives data from the network (a backup) or writes data to the network (a restore). The parameter sets the value for the network buffer size for backups and restores.

UNIX

The default value for this parameter is 32032.

Windows

The default value for this parameter is derived from the NetBackup data buffer size using the following formula:

For backup jobs: (<data_buffer_size> * 4) + 1024

For restore jobs: (<data_buffer_size> * 2) + 1024

For disk: Because the default value for the NetBackup data buffer size is 262144 bytes, the formula results in the following: a default NetBackup network buffer size of 1049600 bytes for backups and 525312 bytes for restores.

To set the network buffer size, create the following files:

UNIX

/usr/openv/NetBackup/NET_BUFFER_SZ

/usr/openv/NetBackup/NET_BUFFER_SZ_REST

Windows

install_path\NetBackup\NET_BUFFER_SZ

install_path\NetBackup\NET_BUFFER_SZ_REST

Note that buffer files contain a single integer that specifies the network buffer size in bytes. For example, to use a network buffer size of 64 kilobytes, the file would contain 65536.

- If the files contain the integer 0 (zero), the default value for the network buffer size is used.
- If the NET_BUFFER_SZ file exists and the NET_BUFFER_SZ_REST file does not exist, NET_BUFFER_SZ specifies the network buffer size for backup and restores.
- If the NET_BUFFER_SZ_REST file exists, its contents specify the network buffer size for restores.
- If both files exist, the NET_BUFFER_SZ file specifies the network buffer size for backups.

The NET_BUFFER_SZ_REST file specifies the network buffer size for restores.

MEDIA SERVER BUFFER SIZES (SHARED MEMORY)

The NetBackup media server uses shared memory to buffer data between the network and the tape drive or disk drive. (If the NetBackup media server and client are the same system, the media server buffers data between the disk and tape.)

The number and size of these shared data buffers (tape and disk buffers) can be configured on the NetBackup media server so that NetBackup optimizes its use of shared memory. A different buffer size may result in better throughput.

You can use this formula to calculate the amount of shared memory that NetBackup requires:

*Shared memory required = (number_data_buffers * size_data_buffers) * max_multiplexing_setting*

For example, assume that the number of shared data buffers is 256 and the size of the shared data buffers is 256 kilobytes. Also assume one FDS share and a maximum multiplexing setting of two. Following the formula, NetBackup requires 8 MB of shared memory:

(256 * 262144) * 2 = 128 MB

FalconStor recommends adapting these parameters to the size of the memory available on the backup server (do not exceed 50% of the total shared memory). Increasing these values will impact the overall performance of the NetBackup system.

CLIENT BUFFER SIZES

The NetBackup client has a tunable parameter to adjust the size of the network communications buffer that writes data to the network for backups. This client parameter is the counterpart to the network buffer size parameter on the media server. The network buffer sizes are not required to be the same on all of your NetBackup systems for NetBackup to function properly. However, if the media server's network buffer size is the same as the client's communications buffer size, you may achieve better performance if they are the same.

FDS SHARE CONFIGURATION

The system administrator must pay particular attention to the NetBackup setup with CIFS shares if the share is mapped to a Windows hosts using *shared mode* authentication. When installing NetBackup software on a Windows platform, the current user is used by default (generally *administrator*). This NetBackup user must have access to the CIFS share in order to use it as a storage unit.

In order to ensure this, two NetBackup services (Client service and remote manager and Monitor service) must be started by a specific user that has been created previously (such as a *Guest* user). If this specific setup is not done, NetBackup won't have access to the CIFS share and the NetBackup job will fail.

This user must have the access to FDS shares and must also be member of the Administrators and Backup Operators groups.

FalconStor FDS configuration guidelines

This section proposes different ways to optimize FalconStor FDS entities in a NetBackup environment.

FALCONSTOR FDS VIRTUAL DEVICE CREATION AND LAYOUT

When configuring FalconStor FDS for the first time, different scenarios are possible:

- The FDS appliance is an all-in-one appliance. Virtual and physical resources are already defined. The only entities that must be configured are the FDS shares.
- The FDS appliance is used as a gateway. In this particular scenario, the physical resources (mainly the disk buffer) and the virtual resources must be configured.

TUNING FALCONSTOR FDS VIRTUAL LAYOUT

FalconStor FDS configuration consists of creating logical resources and creating/mapping FDS shares.

- **Logical resources**: For performance reasons, we recommend creating consolidated FDS resources instead of individual FDS resources in order to spread the workload across multiple physical resources. (**Note:** This option is not available for preconfigured all-in-one appliances.)

  For instance if you created four LUNs of 500 GB each, we recommend creating one FDS resource of 2 TB that includes the four physical LUNs so that the workload is distributed to the four physical LUNs.

- **Mapping**: We do not recommend mapping an FDS share to multiple hosts (even if it is technically possible). This configuration is recommended only when the share must be accessible from multiple hosts. Since you can create a lot of shares, there is no need to limit the number of FDS shares.

TUNING DISK I/O PERFORMANCE:

When FalconStor FDS is configured as a gateway, pay attention to the following items:

- **Insufficient FC paths**: This is the same problem that was described for NICs. The number of FC paths between the FDS server and its disk buffer cannot be customized. If you determine that the bottleneck in your system is the number of FC paths, then you should consider upgrading to a "bigger" FalconStor FDS solution running more FC ports.
- **Under-usage of disk controllers**: This requires special attention and customization when configuring virtual resources. Balancing the load over the disk controllers is part of the FDS configuration.

SIZING

The ratio of data processed to data stored (after deduplication and compression) depends entirely on factors such as:

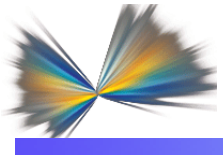
- Backup methods being used and backup policy standards
- Retention policies
- Rate of data change

The sizing of a FalconStor FDS solution must take these factors into account. As NetBackup is not aware of deduplication, there are some general rules that must be followed:

- Propose an assessment of the backup solution so that the FalconStor FDS sizing can be done in collaboration with the backup team that is in place at the customer site. Indeed, discussion with the people in charge of the backup environment will significantly enhance the success of the FalconStor FDS implementation.
- When planning the size of the FDS buffer, take into consideration the size of the initial backup cycle and anticipate a possible deduplication ratio. This should help the FDS administrator to determine (in a first phase) the amount of data that can be retained on disk.
- As everybody knows that the benefits of data deduplication are realized over time, FalconStor recommends reconsidering FDS sizing after the FDS system has been used to send a significant subset of production backup data to a NetBackup media server.

By adopting an incremental approach, the FDS administrator will be able to better forecast the space needed to sustain data growth. This requires ongoing measurement that should help the FDS administrator understand how FalconStor FDS will behave when additional backup jobs are issued.

Ultimately, this approach is certain to help the FDS administrator to better provision the FDS buffer.



Conclusion

Adapting Veritas NetBackup to the FalconStor FDS landscape to take full advantage of the virtual storage layout is not a complicated process. FalconStor FDS is specifically designed to address this challenge. FalconStor FDS is the ideal backup-to-disk target for your Veritas NetBackup backup environment because it:

- Will reduce your backup storage requirements by deduplicating the backup data
- Will reduce/eliminate your tape expenses by extending your retention-on-disk policies
- Will reduce/eliminate your off-site tape management costs, if replication between a source and target FDS appliance is implemented
- Will enhance the security of your backups by avoiding tape transportation

FalconStor FDS is a scalable solution that can work with heterogeneous hardware connected to SMB, SME, and enterprise solutions, overwhelmingly confirming that FalconStor FDS is a totally open solution that can be adapted to different environments, including NetBackup.

Overall logical configuration tuning is very important to make sure that one can realize top level numbers in a real customer environment. This means that the FDS buffer is tuned properly, NetBackup is tuned properly, and each backup server has enough front-end FDS shares to spread the load to the FDS appliance over the IP infrastructure.

Logical configuration becomes very important when you endeavor to take full advantage of FDS appliance performance. It is also important to correlate the performance expectations and their configuration requirements to other FalconStor FDS benefits, such as flexibility in adding physical resources and fast restore operations.

When these technical aspects are taken into consideration before, during, and after FalconStor FDS integration with NetBackup, FalconStor FDS offers a best-in-class data protection system that significantly reduces the costs incurred by traditional backup infrastructures.