HITACHI
Inspire the Next

Hitachi NAS Platform Best Practices

Tiered File Systems

By Francisco Salinas (GSS Engineering)

Hitachi Data Systems

MK-92HNAS038-00

Notices and Disclaimer

The performance data contained herein was obtained in a controlled isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While Hitachi Data Systems Corporation has reviewed each item for accuracy in a specific situation, there is no guarantee that the same results can be obtained elsewhere.

All designs, specifications, statements, information and recommendations (collectively, "designs") in this manual are presented "AS IS," with all faults. Hitachi Data Systems Corporation and its suppliers disclaim all warranties, including without limitation, the warranty of merchantability, fitness for a particular purpose and non-infringement or arising from a course of dealing, usage or trade practice. In no event shall Hitachi Data Systems Corporation or its suppliers be liable for any indirect, special, consequential or incidental damages, including without limitation, lost profit or loss or damage to data arising out of the use or inability to use the designs, even if Hitachi Data Systems Corporation or its suppliers have been advised of the possibility of such damages.

This document has been reviewed for accuracy as of the date of initial publication. Hitachi Data Systems Corporation may make improvements and/or changes in product and/or programs at any time without notice. No part of this document may be reproduced or transmitted without written approval from Hitachi Data Systems Corporation.

Contents

Document History

Revision	***Description***
1	Initial publication
1.1	Document updated. Added fs-analyze-data-usage information.
1.2	Added updates for TFS in newer code releases.
1.3	Format and edit by GSS Technical Publications.

Introduction

This document provides best practices and recommendations for using the Hitachi NAS Platforms Tiered File Systems.

Intended Audience

This document is intended for Hitachi Data Systems field personnel and customers who use Hitachi NAS Platform products.

Overview

The Hitachi NAS Platform offers a feature that automatically and intelligently separates data and metadata onto different Tiers of storage called Tiered File Systems (TFS). In a TFS, the underlying Storage Pool (a logical container of one or more system drives, on which one or more file systems can be created) can support up to two Tiers and is called a Tiered Storage Pool (TSP). All file systems derived from a TSP are by default Tiered File Systems and leverage the intelligent metadata separation. A TSP defines the two Tiers as Tier 0 and Tier 1. This is not to be confused with the storage Tiers referenced in the Storage Subsystem Guide, because a Storage Pool Tier can still leverage any form of storage array Tier.

For example, SSD drives is usually referred to as Tier 0, FC drives and SAS drives are generally referred to as Tier 1, and both NLSAS and SATA are often referred to as Tier 2 Storage. A Storage Pool Tier can use any of these storage array Tiers.

The Storage Pool Tier 0 is the Tier that supports the metadata transactions, which will tend to be very random, small IO. As such, it is highly recommended that the Storage Pool Tier 0 be composed of high speed disk technologies like SSD or SAS drives. The second half of the Storage Pool, Tier 1, will continue to hold the user and application data. The Storage Pool Tier 1 has the potential to use cost effective disk technologies depending on the overall workload profile and the level of metadata offloading. This new ability to create Tiered Storage Pool combinations creates the availability of new price points for storage when mixing the technologies together. Since metadata is nearly always required for file system operations, separating it onto fast storage can greatly improve the responsiveness of the Hitachi NAS Platform. Isolating metadata IO from data IO can in many cases improve the performance when compared to file systems that only implement a single Tier of storage.

Figure 1, below, illustrates the layout of a Tiered File System.

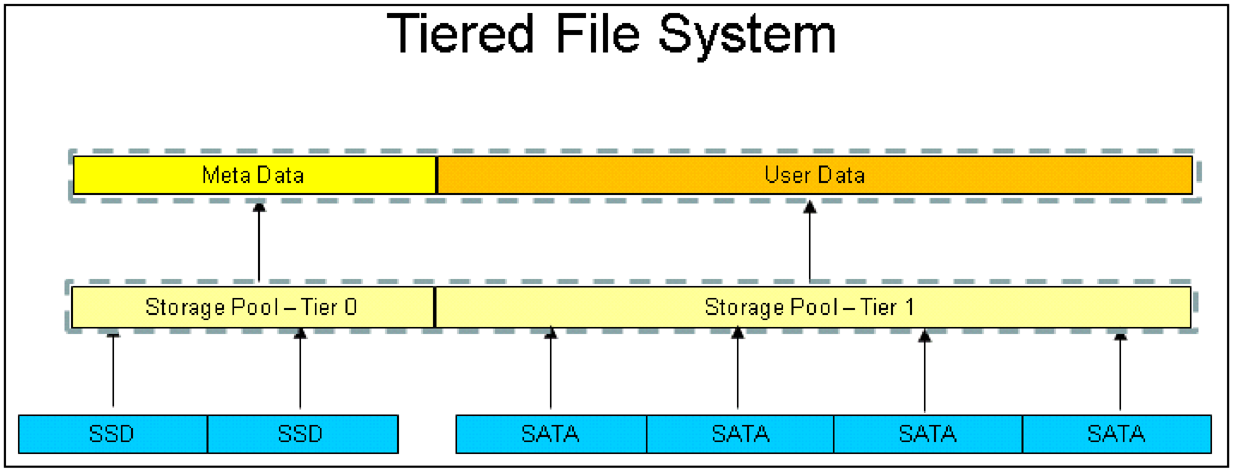

Figure 1 - Tiered File System

Metadata

What is Metadata?

Metadata is data that describes or provides information about other data for the file system and typically for applications. There are many different uses for metadata in information technology and it is important to understand what the file system considers metadata.

There are two types of metadata that a file system recognizes:

1. **Administrative metadata** - This is essentially technical information about the data. In TFS, this includes UNIX and/or Windows properties such as permissions/ACLs, ownership attributes, dates and times about creation, modification and access amongst others. It also includes directories and their properties.

2. **Structural metadata** - Information about how the data is organized on the storage media. In the context of WFS-1, and WFS-2, this is object data called root and leaf onodes, free space bitmaps and other file system internal objects.

   Other types of metadata, such as *descriptive*, *guide,* and *application* metadata, are not recognized by the file system. For example, a database may store metadata in tables about business transactions. To the file system, this is just data. Another example is key words or descriptions in documents. These types of metadata are only recognizable to applications and are not understood by the file system. Archiving, backup and compliance software often generates application specific metadata that is typically held in the same directory as the file for which they represent. This form of application specific metadata is seen as only normal data by the file system.

Important Note: In this document, when the term “metadata” is used, it refers to the types of metadata described in this section.

TFS Performance

What kind of performance improvements does TFS provide?

Performance improvements, as always, will vary. Global Solutions and Services (GSS) Engineering have seen improvements ranging from 30% to 800%. Hitachi NAS Platform file servers are designed to maintain high levels of metadata cache hit, so improvements are only likely to be seen when the server has to go to disk to access metadata (cache miss or cold cache accesses) and for some workloads where Tier 0 takes pressure off Tier 1 by processing write IOs that would have normally gone to Tier 1 storage if TFS was not used. In other words, without TFS, all the disks in the Storage Pool would have to service metadata IO, with TFS, that IO load is no longer present and allows those disks to potentially do more work and/or improve their response time because they are doing less work.

Note: The cache used for metadata in the server (the HNAS node) will in many cases be more than sufficient for normal NAS workloads and deployments. TFS is only recommended when the level of metadata workload begins to have a routine and high level of metadata cache misses that require more and more disk access, and therefore becoming an ideal candidate for TFS.

Listed below are some of the application and workload profiles that are expected to benefit from TFS:

- Heavy metadata workloads with a lot of metadata cache miss.
- Any workloads that are metadata read intensive.
- Write workloads that are also metadata write intensive.
- Workloads that scan large sections of file system.
- Concurrent File Access to Large files (applications that use very large files).
- Hitachi NAS Platform iSCSI logical Units.
- Databases.
- VMware Virtual Machines over NFSv3 and iSCSI.
- Directory listings with cold cache (this will be more apparent on directories with many files).
- Data loading / Data migrations to TFS may run faster (i.e. writes).
- HNAS File Replication, Tape Backup (NDMP), HNAS Data Migrator.

Below is an example of the type of benefit TFS can provide. This was a test done across several Linux clients mounting a single NFS share with 2.8 million files, with each client doing a cold cache “ls” on separate directories containing 700K files each. The chart shows various configurations to provide baseline comparisons.

Note: NLSAS was configured as Tier 1 in the TFS configurations.

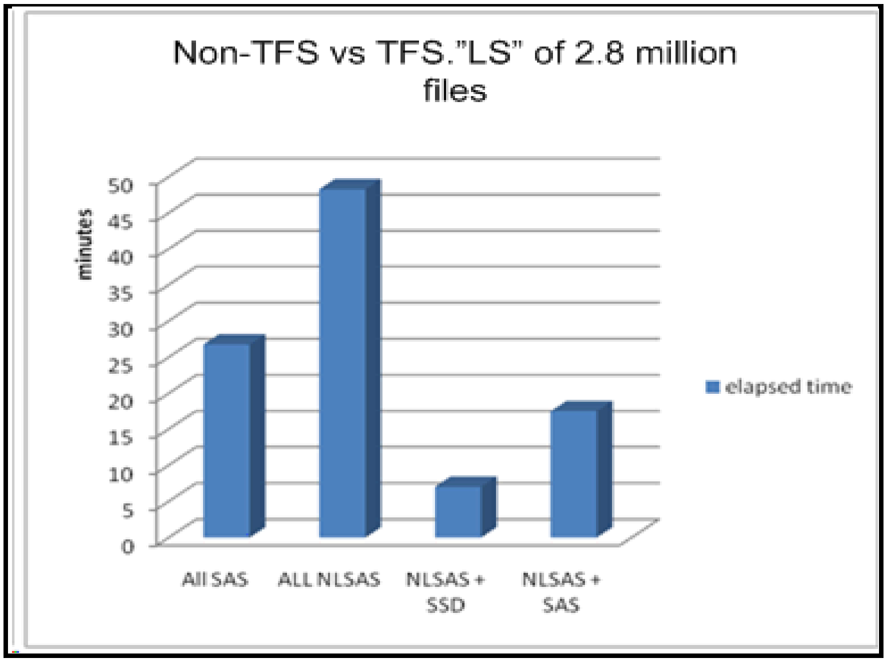

Figure 2 - TFS Performance vs. Non TFS

Misconceptions about TFS

It is important to understand that TFS will not increase the maximum performance capabilities of the server. If the CPU's, FPGA's or other architecture components have reached their limit, then TFS will not help.

If the server is very busy, the performance benefit of TFS may be reduced. Remember, TFS helps get metadata into the server quickly and improves checkpoint times. Outside of performing these functions, there may be other areas of the server that can become bottlenecks and slow down operations.

If the workload has a cache friendly dataset, then apart from the initial cache misses, TFS will not improve metadata read performance because the metadata is already in cache of the server and/or the storage. It may improve write performance if a lot of writing is going on.

If the environment frequently accesses (read) the same files, TFS is not likely to provide a performance increase. Look at HNAS Local or Cluster Read Caching instead.

TFS will not solve other bottlenecks. For example, if the Tier 1 storage is saturated, and the customer is trying to read data from the storage or write data to the storage, while the metadata aspects of the reads or writes will complete faster, it still make take a long time for the Tier 1 storage to service the requests thereby reducing or eliminating the benefits of TFS.

Though we have observed an average of 5% metadata space consumption on a file system, in some situations, where there are very large numbers of small files <8kB, metadata can consume a significant amount of the file system.

Note: Having lots of metadata does *not* imply a metadata workload intensive file system that requires TFS. The two are separate but important design criteria.

Upgrading to TFS

Upgrading Storage Pool File Systems to TFS

It is possible to convert existing file systems to TFS. The process involves adding a Tier 0 of new storage capacity to an existing Storage Pool, then all the file systems on that new Tiered Storage Pool will become Tiered File Systems. This process requires downtime and only new or updated metadata will reside on Tier 0. Existing metadata is eventually moved to Tier 0 as that data is a modified.

For a File System that is very active, this approach should work well. For a File System that is inactive, i.e. an archive, apart from migrating the data to a new Tiered Storage Pool, there is no existing method to move all metadata to Tier 0.

Using DRB to help redistribute metadata to Tier 0 after upgrading a Storage Pool to TFS

Dynamic Read Balancing (DRB) may be used to help move some metadata to Tier 0 after upgrading a Storage Pool to TFS. To use DRB, additional SDs must be added to the Storage Pool and then the File System must be expanded onto the new SDGs. For more information on DRB, see the man page of `fs-read-balancer`.

Note: DRB will not rebalance metadata across Tier 0 System Drive Groups (SDG).

Sizing Capacity for Tier 0

Metadata occupies space on a WFS File System, and it is important to know how much space is needed to size it and to make the most efficient use of Tier 0 resources. The general rule of thumb is to size Tier 0 for 5% of overall file system size

Sizing the amount of capacity needed for Tier 0 involves looking at several factors:

1. **Average file size** - This is the biggest factor contributing to the amount of metadata needed.
2. **File system block size** - This can greatly affect the amount of space metadata occupies based on the average file size.
3. **Number of File System checkpoints (WFS-2**) and **snapshots.**

File Size and File System Block Size

HNAS offers two file system block sizes: 4KB and 32KB. The average file size typically determines which is selected.

Every file on HNAS consumes at least 1 file system block for metadata. So if the average file size is small, it is more efficient to use 4KB file system block size.

For example, if the average file size is less than 32KB, then 4KB may be more appropriate. On the other hand, if the average file size is greater than 32KB, then 32KB is more appropriate.

Determining the amount of metadata in an existing File System

Starting in HNAS OS 7.0, there is a utility called `fs-analyze-data-usage` that provides a detailed breakdown of how space is allocated in an existing File System.

Note: `fs-analyze-data-usage`, until 8.2.2374 or 10.1.x, should be used on read-only file systems. This requires downtime to unmount the targeted file systems and remount them read-only and then reverse the process.

For some application types, i.e. VMware, iSCSI or databases over NFS that make use of large files which are concurrently accessed by many threads, it can be beneficial to store only the leaf onodes on Tier 0 instead of all types of onodes. This is possible by using the `--multi-tier-leaf-onodes-on-tier-one` option when creating a Tiered Storage Pool.

SSD Sizing Practices for Tier 0

Tests with SSDs have shown that one drive can provide from 8000 - 1000 random read IOPS at 4KB request sizes and 1000-1200 random write IOPS at 4KB request sizes. A handful of SSDs can easily provide enough IOPS capability to satisfy the metadata requirements for a very busy server. There are a few considerations when designing a solution for Tier 0:

- RAID level
- Number of System drives per Volume\RAID Group
- Dedicated storage array or shared storage array

The recommendation is to use RAID 5 with SSD. RAID 5 lowers the overall cost of the solution.

Reference the table below to understand how many System Drives can be created for each Volume\RAID Group.

Number of System Drives per Volume\RAID\Storage Pools(DDN)	*Up to 4 per 3+1 SSD/FMD RAID Group* *Up to 8 per 6+2 or 7+1 SSD/FMD RAID Group*

Ideally, to maintain high levels of performance it would be best to have separate storage controllers for Tier 0 and Tier 1. It is expected that the majority of configurations will share the same controller (storage array) for Tier 0 and Tier 1. The following section offers guidance on sharing.

Intermixing SSD and SAS/NLSAS in the same storage system

Intermixing SSD and SAS/NLSAS can provide a cost effective solution for TFS, especially with the bigger HDS and DDN storage systems. When intermixing, one should consider the upper limit of IOPS the storage system can handle and the ratio of performance between SSD and the other drive type installed to ensure SSD performance is not limited by the storage system. Below is a table showing the ratios for SAS.

Operation Type	***SAS to SSD Ratio***	***NLSAS to SSD Ratio***
Write	9 to 1	18 to 1
Read	24 to 1	36 to 1

Figure 2 - SSD to SAS/NLSAS Ratio

The ratios can be relaxed a bit if the configuration is not expected to need the full IOPS capability of the SSD. The table below lists the observed number of IOPS for various storage systems.

System	***Max observed IOPS***
VSP[1]	825K
HUS VM	190K
HUS 150	90K+
AMS 2500	90K
AMS 2300	50K
AMS 2100	25K

[1] - VSP with Flash Acceleration

SAS Sizing and Configuration for Tier 0

Testing has shown that SAS drives can provide up 300-350 IOPS per drive. This is significantly less than SSD and is bound by the normal physics of spinning media. Therefore, when using SAS as Tier 0, a reasonable number of drives should be configured. Tier 0 needs to be able to cope with very large numbers of very small read and write IOPS. As the File System ages, dealing with a large number small random writes becomes important. The best practice recommendation is to configure RAID 1+0 when using SAS as Tier 0.

Note: Not using RAID1+0 can lead to severe performance degradation.

The effect of Snapshots and File System checkpoints on Tier 0 space

HNAS snapshots are based on metadata. When a snapshot is created, metadata is frozen to preserve pointers to the data. When data on the live File System is altered, new metadata is created to account for new or updated blocks. Overtime, as the snapshot ages, the amount of metadata (Tier 0) and data (Tier 1) space increases. It is important to track snapshot utilization on Tier 0 so it does not become overrun.

Also, take into account that snapshots take time to delete, so, tracking space utilization on Tier 0 is important. If Tier 0 is overrun, then metadata will be written to Tier 1. Once metadata is written to Tier 1, it will stay there until the metadata is updated even if there is free space on Tier 0.

Similarly, File System checkpoints will also consume additional capacity in Tier 0. The amount of space will vary depending on the amount of changes preserved in the checkpoints and the activity on the File System. Unfortunately, there isn't a clear way to determine this ahead of time. One way to find out how much space is consumed in Tier 0 is to run **`fs-analyze-data-usage`** and then subtract the total metadata value from the output of the `df` command on the server.

Frequently Asked Questions

- What happens when Tier 0 capacity is exhausted?
  - Metadata will spill onto Tier 1.
- What kind of workload profile will there be on Tier 0?
  - Tier 0 will see a very large number of small random read and write IOPS.
- When I add SDs to Tier 0, will metadata be distributed to the new SDs?
  - Only new metadata or metadata that is updated will be moved to the new SDs via Dynamic Write Balancing. Remember to expand your file system to make use of the new Tier 0 capacity.
- What can I do if I put too few SDs in Tier 0 and I need to add more to improve Tier 0 performance?
  - The best thing to do is not to run into this situation. If you are forced to add SDs to Tier 0, it could take some time before an improvement is seen for metadata cache misses, except for metadata writes during checkpoints. If you also add SDs to Tier 0, then DRB can help distribute metadata in Tier 0, but there is no guarantee that the distribution will equal.

- Can I upgrade to TFS?

  – Yes, a development level command called `span-Tier` is required and the procedure is offline. Metadata will be moved as it is updated due to user IO over time. New metadata is written to Tier 0. After using the span-Tier command, use the `span-expand -t` to assign Tier 0 SDs to the Storage Pool. Please contact support for assistance.

- How can I expand the capacity on Tier 0?

  - This can be done via the HNAS CLI using `filesystem-expand`. See the man page for more details.

- Why can't I mount my File System or create a new File System after running span-Tier?

  – You need to assign Tier 0 capacity to the Storage Pool, as shown below:

```
                       HNAS:$ spans
Span label             OK?  Free  Cap(GB)  System drives                  Con
---------------------  ---  ----  -------  -----------------------------  ---
sp1                    Yes  100%     2537  1,3;0,2,4,5                    90%
   Tier 0: capacity    2537GB; free:    2537GB (100%)
   Tier 1: empty: File Systems can't be created or mounted
```

  – This is a fairly fixed amount of overhead needed by the File System. The table below shows the percentage of a File System this overhead will utilize.

Capacity	***Percentage of fixed overhead***
100GB	2%
500GB	1%
>1TB	.5% or less

Figure 9 – Percentage of File Systemutilized

Hitachi Data Systems

Corporate Headquarters
2845 Lafayette Street
Santa Clara, California 95050-2639
U.S.A.
www.hds.com

Regional Contact Information

Americas
+1 408 970 1000
info@hds.com

Europe, Middle East, and Africa
+44 (0) 1753 618000
info.emea@hds.com

Asia Pacific
+852 3189 7900
hds.marketing.apac@hds.com