

Hitachi NAS Platform

NAS 12.1 HDP Best Practices Guide

This document describes the best practices for configuring and using Hitachi Dynamic Provisioning (HDP), predicated on the use of thin provisioned Dynamic Provisioning (DP)-Volumes.

MK-92HNAS063-01

May 2023

© 2021 Hitachi Vantara LLC. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., or Hitachi Vantara Corporation (collectively, "Hitachi"). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara Corporation at https://support.HitachiVantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara Corporation.

By using this software, you agree that you are responsible for:

- 1) Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals to access relevant data; and
- 2) Verifying that data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

EXPORT CONTROLS - Licensee will comply fully with all applicable export laws and regulations of the United States and other countries, and Licensee shall not export, or allow the export or re-export of, the Software, API, or Materials in violation of any such laws or regulations. By downloading or using the Software, API, or Materials, Licensee agrees to the foregoing and represents and warrants that Licensee is not located in, under the control of, or a national or resident of any embargoed or restricted country.

Hitachi is a registered trademark of Hitachi, Ltd., in the United States and other countries.

AIX, AS/400e, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, Flash Copy, IBM, Lotus, MVS, OS/390, PowerPC, RS6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z/VM, BCPI™ and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, the Microsoft Corporate Logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or web site are properties of their respective owners.

Table of Contents

| | |
|--|-----------|
| Preface | 5 |
| About this document..... | 5 |
| Document conventions..... | 5 |
| Intended audience..... | 5 |
| Accessing product downloads..... | 5 |
| Comments | 5 |
| Getting Help..... | 6 |
| Chapter 1: Introduction..... | 7 |
| Chapter 2: Traditional RAID Group versus HDP Pool Integration..... | 7 |
| Chapter 3: File system | 8 |
| Chapter 4: Span..... | 8 |
| Chapter 5: LU..... | 9 |
| Chapter 6: Calculating free space..... | 9 |
| Chapter 7: File system | 10 |
| Chapter 8: Parity groups (or pool volumes), DP Pools and DP-Volumes | 10 |
| Chapter 9: Span..... | 11 |
| Chapter 10: The LU | 14 |
| Chapter 11: Calculating free space..... | 14 |
| Chapter 12: Mandatory Guidelines and Restrictions | 16 |
| Hitachi HNAS v12.0 and earlier | 16 |
| Hitachi NAS 12.1 and later..... | 16 |
| Additional restrictions enforced in v12.1 or later..... | 17 |
| Chapter 13: Recommended Array Settings | 17 |
| Modular HUS 100, and AMS2000 class storage systems..... | 17 |
| Enterprise HUS-VM/VSP/VSP G1000 class storage systems | 18 |
| Chapter 14: RAID Recommendations..... | 20 |
| Chapter 15: Sharing HDP Pools Guidelines..... | 21 |
| Chapter 16: Pool Initialization and Formatting Recommendations | 21 |
| Chapter 17: Recommendation for Sizing Stripesets and Optimizing Q-Depth | 22 |
| Calculating IOPs potential: | 25 |
| Calculating IOPs potential: | 26 |
| Calculating IOPs potential: | 26 |

| | |
|---|-----------|
| Calculating IOPs potential (due to changing the capacity of the SD):..... | 26 |
| Chapter 18: Maximum DP-Volume size | 30 |
| Chapter 19: Upgrading from prior releases and enabling Thin Provisioning | 30 |
| Chapter 20: Considerations when deleting a span | 34 |
| Chapter 21: Considerations when expanding a span | 34 |
| Chapter 22: System Drive(SD) Group Considerations..... | 35 |
| Chapter 23: Understanding free space on a Span or HDP Pool..... | 36 |
| Chapter 24: Downgrade Considerations v12.1 to v12.0 | 37 |
| To downgrade to v12.0:..... | 37 |
| Chapter 25: Upgrade Considerations | 37 |
| Chapter 26: Rolling Upgrades/Downgrades | 38 |

Preface

About this document

This document describes the best practices for configuring and using Hitachi Dynamic Provisioning (HDP), predicated on the use of thin provisioned Dynamic Provisioning (DP)-Volumes.

Document conventions

This document uses the following typographic convention:

| Convention | Description |
|---------------|--|
| Bold | <ul style="list-style-type: none">Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click OK.Indicates emphasized words in list items. |
| <i>Italic</i> | Indicates a document title or emphasized words in text. |
| Monospace | Indicates text that is displayed on screen or entered by the user. Example: <code>pairdisplay -g oradb</code> |

Intended audience

This document is intended for users using Hitachi Dynamic Provisioning (HDP).

Accessing product downloads

Product software, drivers, and firmware downloads are available on Hitachi Vantara Support Connect: <https://support.hitachivantara.com/>.

Log in and select Product Downloads to access the most current downloads, including updates that may have been made after the release of the product.

Comments

Please send us your comments on this document to doc.comments@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara.

Getting Help

[Hitachi Vantara Support Connect](https://support.hitachivantara.com/en_us/contact-us.html) is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to Hitachi Vantara Support Connect for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

[Hitachi Vantara Community](https://community.hitachivantara.com) is a global online community for customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Chapter 1: Introduction

The Hitachi NAS Platform (HNAS) is a versatile, intelligent, and scalable multi-protocol solution. HNAS can be configured in many ways to meet a variety of requirements. Understanding the principles of how storage is used within HNAS will help you maximize the solutions capabilities and reduce the likelihood of improper configuration. This document details the best practices for configuring and using Hitachi Dynamic Provisioning (HDP), predicated on the use of thin provisioned Dynamic Provisioning (DP)-Volumes. The support for thin provisioned DP-Volumes was introduced in the HNAS v12.1 code release. The document assumes that you are familiar with HDP and HDS storage.

Chapter 2: Traditional RAID Group versus HDP Pool Integration

In the traditional HNAS implementation, RAID groups are created and then presented to the HNAS sever as Logical Units (LU) or Logical Unit Number (LUNS). HNAS views these LUNS as system drives (SD). Best practice mandates a one-to-one mapping of RAID group to LUN. When added to a span or storage pool, each of the system drives will have a corresponding System Drive Group (SDG). This effectively results in one LUN per RAID group, one SD per LUN, and one SDG per SD.

An HDP Pool is typically comprised of multiple RAID Groups. The HDP pool is divided into multiple DP-Volumes. Best practices generally suggests one or two DP-Volumes be created for each installed RAID group within the HDP Pool although the number of HDDs per RG and class of media (HAF, SSD, SAS, NL-SAS) can alter that. These DP-Volumes are viewed from HNAS as system drives. When a span or storage pool is created from an HDP Pool, a single parallel SDG is created that contains all of the SDs/DP-Volumes.

Simplistically speaking, when provisioning storage with HNAS using HDP (v12.1 or later) there are a few steps that should be followed:

- Create RAID Groups or Pool Volumes and the HDP pool with a total capacity as large as required at the time of provisioning. Account for the need of file systems to maintain file system utilization no higher than 85% (to prevent performance degradation).
- Create DP-Volumes no more than three times the capacity of the real storage (300% over-provisioned) in order to meet the anticipated growth over the next few years.
- Create a span on all the DP-Volumes, placing all system drives in a single stripeset.
- Create, format, and mount file systems on the span.
- Share and/or export the file systems.

Use this document in conjunction with the following man pages as they contain additional information on the topic of HDP:

- hdp
- span-create

- span-hdp-thickly-provisioned
- span-list
- span-vacated-chunks
- span-delete-all-filestystems
- filesystem-create

Free Space Allocation – HDP/Non-HDP Hitachi NAS v12.0 and earlier

Chapter 3: File system

Most blocks in a typical file system contain data or metadata. The HNAS file system performs most optimally when at least 15% of the blocks on a file system are unused, although it will continue to function when free space is much lower than that.

Creating or expanding a file or creating a directory uses some of the space on the file system, although sparse files and metadata sometimes reduce the amount of space needed.

If snapshots are in use, deleting a file or directory will free space if the file or directory was created since the latest snapshot was taken. Otherwise, the file or directory will move into a snapshot and continue to occupy space until the oldest snapshot that contains the file or directory is deleted.

If space is available, and if you have not confined the file system or its host span (storage pool), the file system will auto-expand when it becomes too full. Auto-expansion occurs when the file system becomes 85% full. When an auto-expansion occurs, the file system requests enough new space to make its space usage 80% of specified capacity.

To view the free space on a file system, use the `df` command.

Chapter 4: Span

Several LUs (logical units — system drives) make up one stripeset. One or more stripesets make up a span. A stripeset is divided into small regions of disk space called “chunks.” A chunk resides entirely on one stripeset and is either dedicated to one file system or completely unused.

| Code | Chunk Size |
|-------------------|--|
| V11.2 or later | 0.5 GiB to 18 GB |
| V11.1 and earlier | 50 GiB – 320 GiB generally used. In theory up to 1 TiB and as small as 5GiB by default or even smaller at the user’s explicit request. |

Creating a file system or expanding one, whether manually or through auto-expansion, allocates one or more chunks to the file system.

Unlike a file system, a span can see its free space fall to zero without loss of performance. From v12.0 onwards, deleting a file system frees up no space, because the file system goes into the *recycle bin*. Recycling a file system (either manually or automatically) frees up all the space it previously occupied.

In releases prior to v12.0, deleting a file system freed up all its space immediately. Expanding a span by adding a new stripeset adds new free chunks. Future allocations are likely to draw from the new stripeset, but existing data does not move (unless you use DRB, which is not available on HDP based spans).

The amount of space on a span is slightly smaller than the total capacity of all its LUs. A few MiB per LU are used for **COD***, other spanning-level metadata, and VLSI padding.

***COD** = Configuration on Disk which consists of metadata that describes the composition of the span, file system catalog and chunk allocation.

To see the amount of free space on a span, use the *span-list* command.

Chapter 5: LU

An LU, if used at all, is dedicated to a single stripeset of a single span.

Chapter 6: Calculating free space

This section is a complex topic and is included in this document to help you understand current free space calculations.

In a future release, HNAS plans to provide a command that will allow you to more readily see when a system needs more storage.

For example, to calculate the free space of 25 TiB Span S.

- It has 8 TiB on file system A, which is 80% full,
- 9 TiB on file system B, which is 85% full.
- It also has the recently-deleted 5 TiB on file system C in its recycle bin. The total free space outside the live file systems is:

| Location | Amount | Result |
|---------------|----------|--------------|
| Span S | 25-8-9-5 | 3 TiB |
| File system C | All | 5 TiB |
| Total | | 8 TiB |

The entire file system C's space is free because, as once the span runs out of chunks, it automatically recycles file system C to free up space.

You cannot use all 8 TiB because file systems must not get close to 100% full. Leaving 15% free space enables you to store $0.85 * 8\text{TiB} = 6.8\text{ TiB}$ in this space.

We should also look for free space in the live file systems, where we have decided to fill the file system to no than 85% of their capacity. The usable free space inside the live file systems is:

| Location | Amount | Result |
|---------------|--------|---------|
| File system A | 5% x 8 | 0.4 TiB |
| File system B | 0% x 9 | 0 |
| Total | | 0.4 TiB |

Five percent of the existing capacity of file system A is available because it is only 80% full, and we are willing to take it to 85%. File system B is already 85% full and we do not want to take it further.

Add these two figures together, and we get $6.8\text{TiB} + 0.4\text{TiB} = 7.2\text{TiB}$. That is the amount of data you can store without buying more storage or suffering reduced performance.

Free Space Allocation - HDP Hitachi NAS v12.1 and later

Chapter 7: File system

The maximum initial capacity of a file system is limited to 1 TiB. Free space on a file system works in the same way as the non-HDP case, with one exception: background auto-expansion operations are restricted to 64 GiB, give or take a chunk.

In v12.1 and later, manual expansions are limited to 1 TiB; foreground auto-expansions are not bound by a limit in order to satisfy a client write. If a client were to create a 2 TiB file, HNAS fulfills that request and pre-allocates the HDP pages.

Chapter 8: Parity groups (or pool volumes), DP Pools and DP-Volumes

Version 12.1 adds support for HDP thin provisioning, in which a given amount of physical disk capacity is oversubscribed when presented to the HNAS. The Span Manager component within HNAS avoids using more disk space than physically exists (in order to avoid an untimely and unexpected out of physical capacity condition).

This enables customers to add small amounts of space at the storage level in the form of pool volumes or parity groups.

The customer starts with 30 TiB of parity groups or pool volumes. The customer then creates:

A DP pool - the pool's free space is 30 TiB.

- On this DP pool, creates 16 each, 5 TiB DP-Volumes, totaling 80 TiB.
- On these 16 DP-Volumes, creates a single 80 TiB span, on which the customer places one or more file systems.

Each DP-Volume is divided into equal-sized pages, whose capacity, depending on the storage platform, will be either 32 MiB or 42 MiB. It is not necessary for a DP-Volume's capacity to be a multiple of the page size. Writing to any block in a page will cause the entire page to be mapped to real disk space in the parity groups or pool volumes and will reduce the free space on the DP pool by 32 MiB or 42 MiB. Writing anywhere else in the same page will not reduce the DP pool's free space.

HNAS can write anywhere on the 16 DP-Volumes. However, it cannot write everywhere because it cannot allocate more than the 30

TiB that exists on the underlying storage (HNAS can write anywhere, but cannot allocate space everywhere at once). If the storage uses 32 MiB pages, HNAS cannot write to more than $30 \text{ TiB} / 32 \text{ MiB} = 983,040$ different pages on this DP pool's DP-Volumes. The server must know about HDP page boundaries and be clever enough not to use more space than actually exists.

Free space on storage platforms differ slightly. Allocating the first page on each LU may reduce the free space on the pool by N pages. Writing to the next N-1 pages, regardless of whether they are adjacent to the original page, does not reduce the DP pool's free space. On HUS, for example, it has been observed that $N=32$. As a result, on some storage systems, creating a span and writing COD to 16 DP-Volumes can reduce the free space on the DP pool by 16 GiB, but further allocations require no more capacity until the server has used 1 GiB on each DP-Vol.

Chapter 9: Span

Everything described about non-HDP spans above applies to HDP spans.

Continuing with our example of an 80 TiB span that resides on 30 TiB of real storage. Over time, file systems are created, deleted, and recycled, and new file systems take their place. The server can see through the 80 TiB capacity of the DP-Volumes and discern the amount of space that underpins them. HNAS avoids trying to store more than 30 TiB of file systems on the span. However, it must do more than that: it must ensure that new file systems use the same HNAS chunks (and thus the same HDP pages) as recycled file systems. Otherwise, the storage would continue to map the recycled file systems chunks to real disk space, and space-allocations at other locations for the new file systems would fail.

Span Manager does not return recycled file systems chunks to the DP pool. Instead, it keeps track of these chunks in a vacated-chunks list (VC list) that resides in COD, and that introduces a backwards incompatibility with v12.0 at 0 whenever the VC list is not empty. The VC list's free space is considered part of the span's free space: a span might have 78 TiB of free space, of which 3 TiB is in its VC list and the remaining 75 TiB is elsewhere. When the storage is thinly provisioned, the presence of 78 TiB of free space on the span is no guarantee that 78TiB of new data can be stored on the span.

Span Manager controls the allocation of HDP pages by allocating the right HNAS chunks. Allocating one HNAS chunk is equivalent to allocating all the HDP pages underneath it. However, HNAS chunk boundaries do not always coincide with HDP chunk boundaries. When they do not, a single HDP page will be shared between two HNAS chunks. Depending on where the boundaries lie and whether adjacent chunks are already allocated, allocating a chunk

on a stripeset of 16 DP-Volumes may allocate 16 pages more or 16 pages fewer than you might expect. Combine this with the behavior described in the previous section, and you can see that free-space accounting is a little approximate. Predictions are accurate only to the nearest GiB or so per DP-Volume— therefore, if there are 16 DP-Volumes in your span, free-space predictions could easily be 16 GiB out.

This approximation is one reason why the server cannot use every byte of available space on a DP pool. It must leave a few GiBs to ensure that it does not exhaust space and cause I/O errors. Another reason is that chunk size are not aligned with the amount of free space on the span. If your DP pool has 15 GiB of free space but your chunks are 18 GiB each, the server cannot allocate another chunk, and the last 15 GiB goes to waste. Keeping chunks small will minimize the impact of this.

When new space is needed for other file systems, Span Manager looks in the VC list first. Only if the VC list is empty does Span Manager allocate virgin chunks at new locations. One consequence is that recycling a file system increases the amount of free space on the span, but not on the underlying DP pool: GUIs such as Storage Navigator will not report an increase in free space. Likewise, neither will *span-list -s*. Conversely, creating or expanding a file system does not reduce the free space on the DP pool if the span has enough vacated chunks to satisfy the allocation. This behavior becomes much clearer once you are familiar with the VC list and how it is used.

The space in each span's VC list is displayed (accurately) by *span-vacated-chunks*. In addition, from v12.3 onwards, *span-list -s* shows how much of the free space on each stripeset is in the VC list. This information helps you predict which stripesets will be used for future allocations. For example, if Stripeset 0 has vacated chunks and Stripeset 1 does not, all new allocations will come from Stripeset 0 until its vacated chunks have been exhausted.

From v12.1 onwards, *span-list -s* shows how much free space is in the DP pool that hosts the DP-Volumes that make up each stripeset.

- If the same DP pool contributes space to two or more stripesets, the same free-space figure will be displayed twice, but you obviously cannot add the numbers together to produce an inflated estimate of free space. You can add them together if the stripesets are all hosted on different DP-Volumes.
- If (against best practice) a single DP pool contributes DP-Volumes to two or more spans, you cannot count its free space against every one of those spans. If Span A allocates space from a DP pool, that space will not be available for reuse by Span B.

The server offers a *span-unmap-vacated-chunks* command, which pulls chunks from the VC list and uses SCSI commands to hand them back to the DP pool. The pages they occupied can then be remapped to other DP-Volume offsets when the server allocates new chunks. However, this process has a number of disadvantages, including long execution times, reduced I/O performance, temporary unavailability of storage, and the need for manual management and monitoring. It is recommended that customers avoid all these disadvantages by relying on the VC list, which is completely automated.

In the following example, we use the same figures as described earlier: 30 TiB of disk space presented as 80 TiB of DP-Volumes, all used in a single span.

| Operation | Span free space | Including VC list space | DP pool free space |
|---|--|--------------------------------|---------------------------|
| New span | 80 TiB | 0 | 30 TiB |
| Create 3 TiB file system Alpha | 77 TiB | 0 | 27 TiB |
| Create 2 TiB file system Bravo | 75 TiB | 0 | 25 TiB |
| Delete Alpha | 75 TiB | 0 | 25 TiB |
| Recycle Alpha | 78TiB | 3TiB | 25TiB |
| Create 1 TiB file system Charlie | 77TiB | 2TiB | 25TiB |
| Expand Charlie to 2 TiB | 76TiB | 1TiB | 25TiB |
| Expand Charlie to 3 TiB | 75 TiB | 0 | 25TiB |
| Expand Charlie to 4 TiB | 74 TiB | 0 | 24 TiB |
| Delete & recycle Charlie | 78 TiB | 4 TiB | 24 TiB |
| span-unmap-vacated-chunks (followed by a long wait) | 78 TiB | 0 | 28 TiB |
| Expand Bravo to 30 TiB | 50 TiB | 0 | 0 |
| Expand Bravo to 31 TiB | Expansion fails non-disruptively: too little real disk space | | |
| Add 5 TiB of pool volumes | 50 TiB | 0 | 5 TiB |

| Operation | Span free space | Including VC list space | DP pool free space |
|----------------------------|-----------------|-------------------------|--------------------|
| Expand Bravo to 31 TiB | 49 TiB | 0 | 4 TiB |
| Add 45 TiB of pool volumes | 49 TiB | 0 | 49 TiB |

At this point, the DP pool is no longer thinly provisioned. The total capacity of its DP-Volumes equals the total capacity of its parity groups or pool volumes. If you want to store more than 80 TiB of data on the span, it will be necessary to create more DP-Volumes and add them to the span. In this example, we create 16 additional 5 TiB DP-Volumes and add them to the span by issuing the *span-expand* command:

| | | | |
|-----------------------------------|---------------|----------|---------------|
| Add 10 TiB of pool volumes | 49 TiB | 0 | 59 TiB |
| Create 16 new 5 TiB DP-Volumes | 49 TiB | 0 | 59 TiB |
| `span-expand' | 129 TiB | 0 | 59 TiB |

The system can now store up to 90 TiB of data on the span. Adding further parity groups or DP-Volumes, up to a total of 160 TiB, allows the system to store 160 TiB on the span without the need for a further *span-expand* operation. Beyond that point, a third stripeset of 16 DP-Volumes would be needed.

Chapter 10: The LU

GUIs such as Storage Navigator can report the amount of real storage mapped to each LU, but it is not a very useful number for HNAS purposes.

Chapter 11: Calculating free space

Let us assume the simple and strongly recommended case in which each span resides on its own DP pool.

Span S is created on a thinly provisioned HDP pool. Clearly, the span's headline capacity (total capacity of its DP-Volumes) is an arbitrary number that does not enter into the calculation. Imagine that the DP pool has 4.3 TiB of free space, and the span has 1.5TiB in its VC list. It has the same three file systems (A and B live, and C recently deleted) as in the non-HDP example. The amount of free space outside the live file systems is as follows:

| Location | Amount | Which is |
|---------------|--------|----------|
| The VC list | | 1.5 TiB |
| The DP pool | | 4.3 TiB |
| File system C | All | 5 TiB |
| Total | | 10.8 TiB |

(The fact that this number is larger than for the non-HDP case is not due to the use of HDP. In this example, a little more disk space has been provisioned. HDP neither increases nor reduces the amount of data you can store on a given consignment of disks, whether or not you use thin provisioning.)

All of deleted file system C's space is considered to still be available. If the span runs out of space, the server will immediately recycle file system C to free up chunks.

As before, 15% of the capacity should be reserved to avoid allowing the file systems to fill up completely. That reduces the value to 9.18 TiB.

If the span had DP-Volumes on two or more DP pools, and if those DP pools were both dedicated to the span, one could add the free space on all of the DP pools. Conversely, if, in contravention of best practice, there was a single DP pool and it was shared with other spans or foreign hosts, one would have to come up with 2 numbers: the 6.5 TiB in the VC list and deleted file system C would be guaranteed to be available, and the remaining 4.3 TiB on the DP pool would be only conditionally available, in that it was shared with other spans and might go away at any moment. Allocating space on one span or cluster would make that much space unavailable to all other users of the DP pool. Ease of accounting and management is another reason for using a separate DP pool for each span.

span-list -s will warn if a DP pool is shared with other spans that are loaded on this cluster. It does not warn if a DP pool is shared with other clusters or foreign hosts through DP-Volumes that the local cluster cannot see.

The usable free space inside our span's live file systems is exactly the same as in the non-HDP case:

| Location | Amount | Which is |
|---------------|------------|----------|
| File system A | 5% x 8 TiB | 0.4 TiB |
| File system B | 0% x 9 TiB | 0 |
| Total | | 0.4 TiB |

We add this figure to 9.18 TiB to arrive at a total of 9.58 TiB.

If there is free space in the VC list, on the DP pool, and on a recently-deleted file system in the recycle bin, the server allocates space from those three sources in that order. As a result, it will not recycle a file system (making un-deletion impossible) until it has no other source of chunks. However, if the file system was deleted more than 7 days ago, any chunk-allocation will cause the deleted file system to be recycled and its chunks to be added to the VC list before any allocation takes place, even if space is available elsewhere. As a result, file systems deleted more than 7 days ago do not cause the server to allocate capacity from the DP pool unnecessarily.

HDP with Hitachi NAS Best Practices

Chapter 12: Mandatory Guidelines and Restrictions

HDP is a feature of Hitachi Unified Storage (HUS), Adaptable Modular Storage (AMS2000), and Virtual Storage Platform (VSP) class storage arrays that HNAS can leverage. HDP is supported for use with HNAS with a few mandatory guidelines:

- DP-Volumes (or ANY logical unit presented to HNAS) must *never* be re-sized.
- HNAS volumes cannot be re-sized, either online or offline.
- To add storage to HNAS, create more SDs, after adding additional RAID groups to the HDP Pool

Hitachi HNAS v12.0 and earlier

- The HDP Pool must not be oversubscribed. Ideally use a dedicated HDP pool for HNAS.
- For VSP and VSP class storage subsystems, set oversubscription to 95%.
- For HUS100 or AMS2000 series, use full capacity mode volumes.

Hitachi NAS 12.1 and later

- The HDP Pool may be oversubscribed. The modest use of HDP thin provisioning, by a factor of two or three to one at most (see [Recommendation for Sizing Stripesets and Optimizing Q-Depth](#) on page 22) is recommended.
- For HUS100 or AMS2000 series, DO NOT use full capacity mode.
- DO NOT use Zero Page Reclaim (ZPR) with HNAS

Additional restrictions enforced in v12.1 or later

- No span can contain a mixture of DP-Volumes and plain parity groups (though, on a single cluster, some spans can use DP-Volumes while others use parity groups).
- No stripeset can contain DP-Volumes from more than one DP pool. (Every new span contains one stripeset; a further stripeset is added every time the span is expanded.) However, different stripesets within the same span can optionally reside on different DP pools.
- This includes tiered file systems where one tier is comprised of SDs from regular parity groups and the other is comprised of SDs that reside on an HDP Pool.

If planning an upgrade to v12.1 or v12.2, and your configuration violates the above rules, functionality is severely hampered – file systems will still mount and be read/write accessible; however, you will be unable to create, expand, or delete file systems on the affected span.

V12.3 introduces *span-hdp-thickly-provisioned* command that will disable thin provisioning support and waive the above two restrictions. This command only releases the restriction - the pool should still be reconfigured to meet best practices. Therefore, it may be advisable to withhold upgrades if your configuration violates the above rules.

As an example, consider the following configuration created in an earlier release:

- SD 8 resides in HDP pool 0
- SD 9 resides in HDP pool 1
- SD 10 resides in HDP pool 2
- SD 11 resides in HDP pool 3

Upon upgrade, you will find the following error:

```
** This span breaks HDP rules, making some management functions impossible:  
** The span violates HDP rules by containing DP-Volumes from more than one HDP  
pool in a single stripeset
```

Note that issuing the *span-hdp-thickly-provisioned* command in V12.3 means it is no longer safe to thin-provision.

Chapter 13: Recommended Array Settings

Modular HUS 100, and AMS2000 class storage systems

System Settings

- Use Advanced Wide Striping Mode (AWS) (except when using HDT).
- DO NOT use full capacity mode.

Host Group and Port Options

While not directly related to HDP, enable the following:

- On each FC target port being used by HNAS, enable Command Queue Expansion mode (Array Name > [Settings](#) > [Port Settings](#) > *Port Options*)
- On each Host Group containing an HNAS WWPN, enable “HNAS Option Mode” (*Groups > Host Groups > 000:G000 > Options*)

Enterprise HUS-VM/VSP/VSP G1000 class storage systems

SOM and HMO Settings

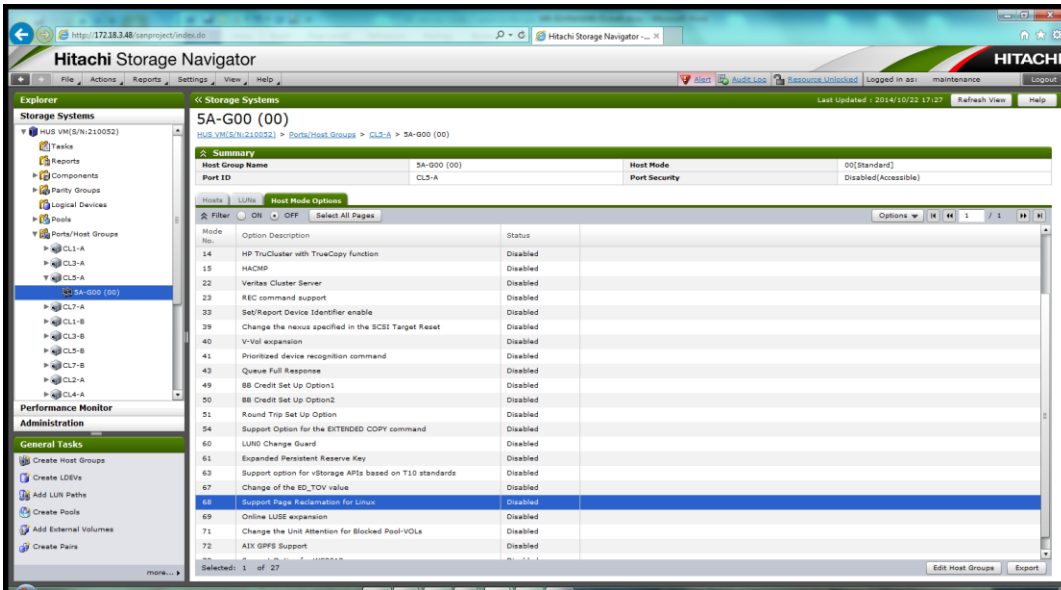
SOM (System Option Mode) and HMO (Host Mode Option) apply to Enterprise RAID - for simplicity, the examples will use VSP, VSP G1000, HUS VM and the forthcoming VSP Gx00 series. Unless otherwise noted, it is assumed the options are available on all arrays.

HMO 68 - Enabled on each host group being accessed by HNAS. This will allow the server to access information about the device that is required to support HDP page reclaim functionality using the UNMAP and WriteSame SCSI commands in order to determine the amount of vacated chunks to process as part of the `span-unmap-vacated-chunks` command. While the command to vacate chunks is not expected to be widely used, enabling HMO 68 is still required.

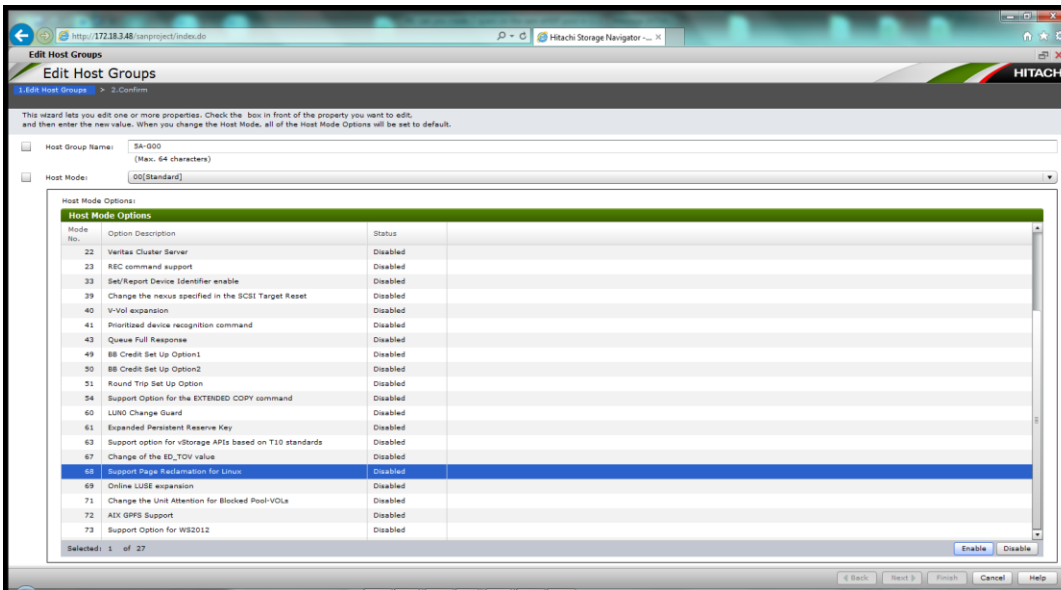
On HUS 1x0 Modular Storage there is no associated Host Mode Option because the array FW implicitly supports the underlying SCSI command structures that enable HNAS to interrogate the used chunks. Vacated chunks are physical regions of disk space on an HDP pool that were used as part of a file system that has been deleted or recycled on a span – they are not HDP pages. These chunks range in size from .5 GiB to 18 GiB (when the storage pool is created with the SMU)

For each Host Group containing HNAS FC initiator port WWN's, do the following:

1. On Storage Navigator, select the **Ports/Host Groups** tab, select the Target Port that HNAS will use for access; and then select the desired Host Group on the Target Port.
2. Once selected, click the **Host Mode Options Menu**



3. Click the Edit Host Groups icon



4. Select Host Mode option 68 Support Page Reclamation for Linux and then click the Enable icon.

Follow the remaining steps in Storage Navigator to complete the task.

Repeat the process for each Host Group containing an HNAS FC port WWN.

Set SOM 917 to ON - This mode supports rebalancing to average page usage rate among parity groups or external volume groups in which pool volumes are defined to reduce drive workload of the parity groups or external volume groups.

SOM 896 ensures that pages are formatted in the background rather than at allocation. Setting this incorrectly will cause pages to be formatted upon allocation, incurring a performance penalty.

HUS-VM, verify SOM 896 is OFF (this is the default) - note, the desired HUS VM functionality occurs as default; setting this bit to **ON** will disable the functionality.

- VSP, set SOM 896 to ON.
- VSP G1000/Gx00, verify SOM 896 is ON (this is the default) - note, the desired VSP G1000/Gx00 functionality occurs as default; setting this bit to **OFF** will disable the functionality.

SOM 901 to ON. By setting this mode to ON while the drive type of DT tier1 is SSD, page allocation of Tier Level ALL will not go to the next tier until the capacity reaches the limit without consideration of performance potential. This ensures that the pages are not demoted when the performance utilization on an SSD tier exceeds 60%. As the response times from the SSD are relatively good, the performance utilization on the SSD tier can exceed the 60% recommended limit. This SOM is not available for HUS-VM.

SOM 904 to ON (but not by default, be guided by GSC analysis). This mode is used to reduce the load of drive so as to improve host I/O performance by reducing the amount of page migration per second (42MB/s) at tier relocation. This option should be explored if an HNAS system experiences storage response time degradation in the environments with frequent relocation and rebalancing. This SOM is not available for HUS-VM.

LDEV Sharing

On the HUS100 and AMS2000 series storage subsystems, whole RAID groups are assigned to an HDP Pool. In Enterprise class storage subsystems, LDEVs are assigned to DP-Volumes and referred to as "pool volumes."

The best practice is to allocate all defined LDEVs in a RAID group to an HDP Pool. On Enterprise RAID (HUS VM, VSP, VSP G1000), NEVER share LDEVs from a parity group between HDP pools, although it is possible to do so. This can result in unpredictable performance. For example, if Parity Group 1-1 is comprised of LDEVs 0 and 1, do not place LDEV 0 into HDP pool 0 and LDEV 1 into HDP pool 1.

Chapter 14: RAID Recommendations

Use of 6+2 DP RAID Groups across all storage product lines is recommended, regardless of media type. This RAID type was chosen because it is very common and available across both modular and enterprise, although 14+2 DP RAID Groups may also be used on Enterprise Class Arrays. See [Recommendation for Sizing Stripesets and Optimizing Q-Depth](#) on page 22 for more information.

RAID5/RAID10 are permitted and valid, However, RAID6 offers superior protection against failure, especially in larger pools.

Mixing different capacity RAID groups in an HDP pool is permitted. For example, the original HDP pool comprised 6 x (6+2) DP RAID groups using 600 GB drives. After 2 years, the smallest capacity available was 900 GB. The HDP pool is expanded with 6 x (6+2) using 900 GB drives. All of the new capacity is available to use. The newly added drives are *not* truncated to the smaller drive capacity.

However, there may be performance implications by doing this as the larger disks will carry more data and could therefore be busier than the small drives. Short and uneven spurts of access to these drives may very well increase response times to HNAS. Proportionally speaking, the smaller disks will carry less load and thus, the overall workload on the pool may be less predictable. Although less busy, the smaller DP RAID Groups will receive as much IO as possible, given that are less IOPs per TB than there were previously.

Chapter 15: Sharing HDP Pools Guidelines

- Do not share an HDP pool between multiple spans (enforced through FW/SMU up to v12.2, but available as an override in the CLI. In v12.3, the restriction is waived and SMU will allow the user to share an HDP pool between multiple spans, but the CLI will still require an override).
- Do not share HDP Pools between multiple clusters
- Do not share HDP Pools between HNAS and other servers (Linux/Windows, VMWare, and so forth.)
- It is not recommended that an HDP pool be shared across spans, clusters, or with foreign hosts because capacity cannot easily be re-assigned within the HDP pool from an HNAS span to another span, foreign host, or HNAS cluster.

For example, if capacity is allocated from an HDP pool to a file system belonging to a span and that file system is then deleted, the associated capacity cannot be allocated to a different span or another host or cluster that shares that same HDP Pool unless it is first unmapped from the original span.

The un-mapping (of blocks) process can be both resource-intensive and take a significant amount of time to complete. For this reason, it is not recommended that an HDP pool be shared across spans, clusters, or foreign hosts.

Combining workloads on the same HDP pool (from foreign hosts or clusters) could lead to unpredictable performance.

Chapter 16: Pool Initialization and Formatting Recommendations

ALWAYS wait for an HDP pool initialization to sufficiently complete, before creating a file system. Failing to allow sufficient capacity to complete initialization will result in extremely slow IO, with the high likelihood of temporary file system failure. For example, if the initialized capacity of the HDP pool is 5 TiB (out of 50 TiB), you can create a 1 TiB file system (default maximum as of v12.1) and manually expand this to 5 TiB (4 expansions). Of course, those additional expansions are not necessary, unless there is a burgeoning need to have a 5 TiB file system immediately available (simple auto-expansions as data is loaded onto the file system would occur).

To help speed up the format process during initial deployment of the HDP pool on the Modular (HUS 1x0 or AMS2000) arrays, set the Format Priority mode as follows:

- HUS130_XXXX > Groups > Volumes > Format Mode

Format Priority Mode = Format (for fastest format, but at the expense of IO from HNAS; this mode may complete format within a day or two).

Format Data = Zero Data (this mode is the underlying default for HDP on the HUS 1x0, regardless of the actual setting).

- Set the Format Priority Mode = Host, in the following cases:
When expanding an HDP pool by adding physical disk (for slowest format, with HNAS IO taking priority).
- When vacating chunks for ANY reason (for slowest zero initialization as pages are zeroed and returned to the HDP pool).

Note: During hours where client IO to HNAS may lessen, you may want to set the Format Priority Mode = Normal (or even Format if IO to HNAS is minimal).

Chapter 17: Recommendation for Sizing Stripesets and Optimizing Q-Depth

When attempting to size the stripe sets for an HNAS implementation that is utilizing HDP and HDP oversubscription, there are a few considerations to use in order to create the appropriate configuration.

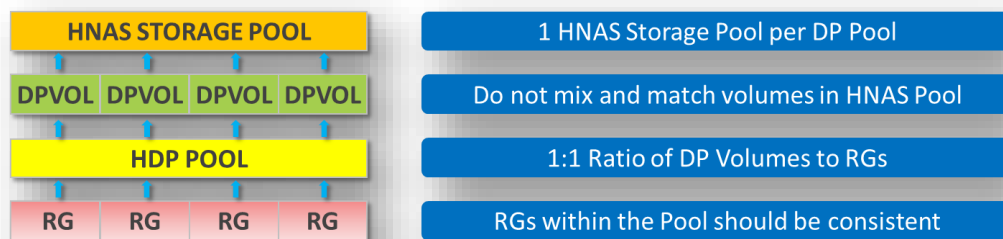
When creating the span and the initial stripeset, it is generally recommended to place all SDs in the initial stripeset (until stripeset expansion is needed). However since the maximum number of SDs in a single stripeset is limited to 32, you might consider breaking into smaller stripesets given the following restriction that begin in v12.3:

Every new span must contain at least four SDs (DP-Volumes).

- Every subsequent stripeset must contain at least four SDs (DP-Volumes) **AND** at least as many as the first stripeset in the span. Because DP pools will be thinly provisioned, this requirement does not commit the customer to large expansions. For example, if the initial stripeset contained 16 SDs, any expansion must contain 16 SDs.

Forty-eight drives is the recommended starting number of HDD (24 is the absolute minimum) per HDP pool. This statement applies to SAS/NL-SAS, as it is understood that for SSD/FMD the minimum amount of 24 modules may not be available in the array or may not be practical in terms of cost.

The ratio of DP-Volumes to physical RAID groups is important. Undersubscribing can starve the storage and potentially reduce performance. Oversubscribing can overload the storage, increase response times and thereby reduce the overall performance. The basic ratio of 1:1 (see the Figure below) can be adopted; however 2:1 (DP-Volume/RG) may be more appropriate when using SAS 10K/15k drives – particularly when there are a larger number of drives per PG.



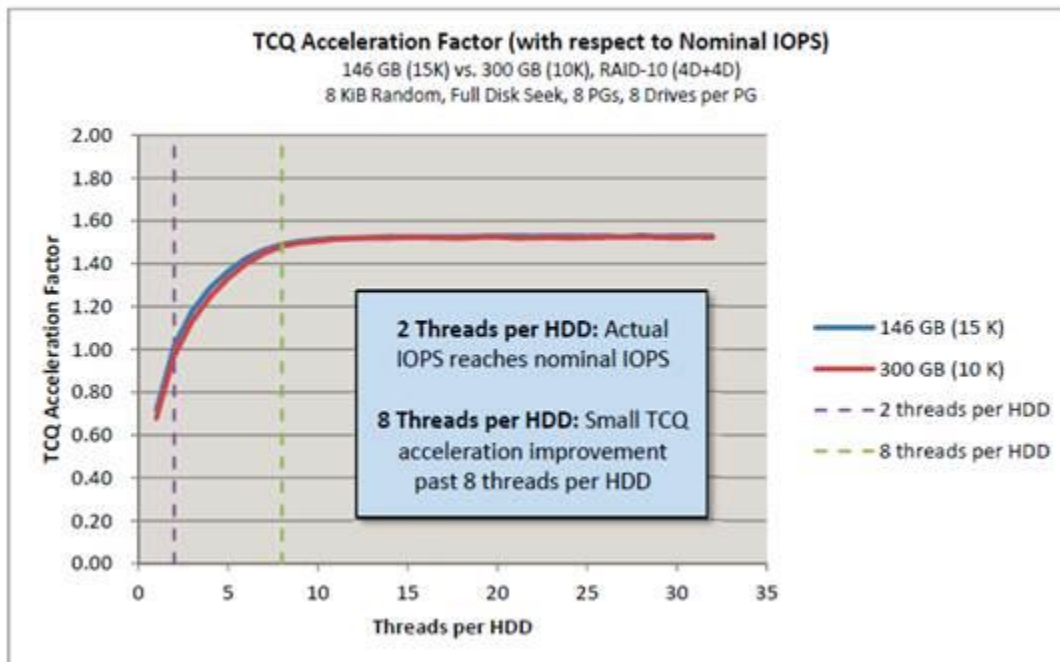
HNAS supports a maximum of 512 LUNs per cluster. While creating many smaller DP-Volumes may increase capacity granularity, it may also significantly reduce an HNAS cluster's ability to scale to maximum storage capacity, if the LUN size is limited.

In HNAS v12.3 and below, the HNAS Q-Depth per SD is set to 32 (at an array level). Therefore, a larger ratio may be required for certain use cases, including the use of HAF (Hitachi Accelerated Flash) or SSD. The use of Array Program Products may also mandate a different ratio due to their LDEV size restrictions. HNAS v12.4 will provide additional flexibility by allowing Q-Depth to be set/modified on a per SD basis (also known as, Per SD Queue Depth).

Ultimately this enhancement may provide additional flexibility in attempting to size and adjust queue depth over time to the appropriate level in order to provide optimal performance.

Internal testing demonstrates that the optimal performance for a drive type/device depends greatly on the number of concurrent threads.

Figure 8. Tagged command queuing acceleration scalability



The following table provides guidance on the recommended maximum tags per device.

| DRIVE TYPE | TAGS |
|-------------|------|
| 10k/15k SAS | 8 |
| NL-SAS | 4 |
| SSD | 16 |
| HAF | 64 |

Based on the information above, you can calculate the **maximum** number of DP-Volumes per PG as follows (assuming 6+2, R6 PG), where 32 is the System Drive Q-Depth per LUN.

| DRIVE TYPE | DP VOLS/PG |
|-------------|------------------|
| 10k/15k SAS | $(8*8)/32 = 2$ |
| NL-SAS | $(8*4)/32 = 1$ |
| SSD | $(8*16)/32 = 4$ |
| HAF | $(8*64)/32 = 16$ |

Consider also, the following examples below, where a “Rule of 8TB SD” has been created in order to come up with a methodology for sizing SAS configurations.

The “Rule of 8TB” is a simplistic set of rules designed to provide a balance between Pool scalability, Queue Depth, and Overall Pool size. It is not a hard and fast rule for sizing, but designed to illustrate the thought process involved in arriving at an appropriate configuration. It may not apply to all configurations, and may in some cases, limit the overall scalability of the HNAS cluster since 8TB SDs would limit the cluster scalability to $8TB \times 512 = 4PB$.

The following is a generic example for calculation of SD capacity and the number of SD's:

1. Use 200% x physical capacity, a maximum of 300%
2. Calculate the physical disk IOPS (PDQ)
3. Use a standard SD size of 8TB or multiples of 8TB (16, 24, 32)
4. Calculate the SD capacity and then select an optimal number of SDs in multiples of 8 (8, 16, 24, 32 etc).
5. Calculate the SD Queue QePTH (SDQD)
6. Aim to have the PQD \geq SDQD, and adjust the results as necessary to increase the number of SDs so the SDQD is close to the PQD.

10k SAS – 600 GB HDD

- 26 Configured Parity Groups * (6 HDD * 600GB/HDD) = 81.6 TB
- 81.6 TB (*200%) = 163.2 TB (200% is desired overcommit ratio)
- 163.2 TB/8 TB per SD = 20.4 SD (*using the rule of 8 TB SDs*)

Therefore, create 32 System Drives (this recalculated overcommit ratio of approx. 314%)

Note: The choice of 32 SD's is not a strict requirement and is chosen merely as a power of 2 reference. One could easily choose 20 SD's or 24 SD's, with the goal to keep an even number balanced against HNAS FC ports and controller target ports.

Calculating IOPs potential:

- 26 Parity Groups * 8 HDDs per Parity Group = 208 HDD's
- From above, 8 tags per 10k HDD * 208 HDDs = 1664 tags (capability)
- Scsi queue depth default on HNAS = 32 tags
- 32 SDs * 32 tags/SD = 1024 tags
- 32 SDs provides enough outstanding tags and falls within the 2:1 DP-Volume to PG ratio.

10k SAS – 1.2 TB HDD

- 26 Configured Parity Groups * (6 HDD * 1200 GB) = 166.4 TB
- 166.4 TB (*200%) = 332.8 TB (*200% is desired overcommit ratio*)
- 332.8 TB/8 TB per SD = 42 SD (*using the rule of 8TiB SDs*)

Therefore, create 32 System Drives (recalculated overcommit ratio of approximately 154%)

The same IOPs potential applies as the above example as we are using the same number of SDs (32).

10k SAS – 900 GB HDD using 14+2 (**note the change in RAID configuration)

- 6 Configured Parity Groups * (14 HDD * 900GB/HDD) = 67.2 TB
- 67.2 TB (*200%) = 134.4 TB (*200% is desired overcommit ratio*)
- 134.4 TB/8 TB per SD = 16.8 SD (*using the rule of 8 TB SDs*)

Therefore, create 16 System Drives (recalculated overcommit ratio of approximately 190%)

Calculating IOPs potential:

- 6 Parity Groups * 8 HDDs per Parity Group = 96 HDDs
- From above, 8 tags per 10k HDD * 96 HDDs = 768 tags (capability)
- Scsi queue depth default on HNAS = 32 tags
- 16 SDs * 32 tags/SD = 512 tags

16 SDs provides enough outstanding tags and falls in the 2:1 DP-Volume to PG ratio.

7.2k NL-SAS 4 TB

- 5 Configured Parity Groups * (6 HDD * 4000 GB/HDD) = 106.8 TB
- 106.8 TB (*200%) = 213.6 TB (*200% is desired overcommit ratio*)
- 213.6 TB/8 TB per SD = 27 SD (*using the rule of 8 TB SDs*)

Calculating IOPs potential:

- 5 Parity Groups * 8 HDDs per Parity Group = 40 HDDs
- From above, 4 tags per NL-SAS HDD * 40 HDDs = 160 tags (capability)
- Scsi queue depth default on HNAS = 32 tags
- 27 SDs * 32 tags/SD = 864 tags
- ** 864 tags far exceeds the capability of the NL-SAS disk, so we must choose a different methodology to provide a reasonable number of SDs.
- Because 8 TB per SD causes a large concern with queuing, choose a larger capacity SD.
- 213.6 TB/ 32TB per SD = 8 SD (*using a rule of 32 TB SDs*)
- Therefore, create 8 System Drives (recalculated overcommit ratio of approximately. 225%).

Calculating IOPs potential (due to changing the capacity of the SD):

- 5 Parity Groups * 8 HDDs per Parity Group = 40 HDD's
- From above, 4 tags per NL-SAS HDD * 40 HDDs = 160 tags (capability)
- SCSI queue depth default on HNAS = 32 tags

- 8 SDs * 32 tags/SD = 256 tags

Eight SDs may exceed the IOPs capability and fall outside the 1:1 DP-Volume to PG ratio for the NL-SAS, but this is not expected to cause performance issues. If HNAS queuing becomes a concern, adding additional Parity Group(s) could help to redistribute load and in v12.4, per SD queue depth could also be invoked.

Alternatively, the following table may provide a handy guideline in attempting to size an appropriate number of SDs based on the drive type and spindle count.

| HDP System Drive Count Per Storage Pool | | | | |
|--|---------------|------------|------------|------------|
| Spindle Count | NL-SAS | SAS | SSD | HAF |
| 4 | | | 2 | 8 |
| 6 | | | 3 | 12 |
| 8 | | | 4 | 16 |
| 10 | | | 5 | 20 |
| 12 | | | 6 | 24 |
| 14 | | | 7 | 28 |
| 16 | | | 8 | 32 |
| 18 | | | 9 | 36 |
| 20 | | | 10 | 40 |
| 22 | | | 11 | 44 |
| 24 | 8 | 8 | 12 | 48 |
| 26 | 8 | 8 | 13 | 52 |
| 28 | 8 | 8 | 14 | 56 |
| 30 | 8 | 8 | 15 | 60 |
| 32 | 8 | 8 | 16 | 64 |
| 34 | 8 | 8 | 17 | 68 |
| 36 | 8 | 8 | 18 | 72 |
| 38 | 8 | 8 | 19 | 76 |

| HDP System Drive Count Per Storage Pool | | | | |
|---|--------|-----|-----|-----|
| Spindle Count | NL-SAS | SAS | SSD | HAF |
| 40 | 8 | 8 | 20 | 80 |
| 42 | 8 | 8 | 21 | 84 |
| 44 | 8 | 8 | 22 | 88 |
| 46 | 8 | 8 | 23 | 92 |
| 48 | 8 | 16 | 24 | 96 |
| 50 | 8 | 16 | 25 | 100 |
| 52 | 8 | 16 | 26 | 104 |
| 54 | 8 | 16 | 27 | 108 |
| 56 | 8 | 16 | 28 | 112 |
| 58 | 8 | 16 | 29 | 116 |
| 60 | 8 | 16 | 30 | 120 |
| 62 | 8 | 16 | 31 | 124 |
| 64 | 8 | 16 | 32 | 128 |
| 66 | 16 | 16 | 33 | 132 |
| 68 | 16 | 16 | 34 | 136 |
| 70 | 16 | 16 | 35 | 140 |
| 72 | 16 | 16 | 36 | 144 |
| 74 | 16 | 16 | 37 | 148 |
| 76 | 16 | 16 | 38 | 152 |
| 78 | 16 | 16 | 39 | 156 |
| 80 | 16 | 16 | 40 | 160 |
| 82 | 16 | 24 | 41 | 164 |
| 84 | 16 | 24 | 42 | 168 |
| 86 | 16 | 24 | 43 | 172 |

| HDP System Drive Count Per Storage Pool | | | | |
|--|---------------|------------|------------|------------|
| Spindle Count | NL-SAS | SAS | SSD | HAF |
| 88 | 16 | 24 | 44 | 176 |
| 90 | 16 | 24 | 45 | 180 |
| 92 | 16 | 24 | 46 | 184 |
| 94 | 16 | 24 | 47 | 188 |
| 96 | 16 | 24 | 48 | 192 |
| 98 | 16 | 24 | 49 | 196 |
| 100 | 16 | 24 | 50 | 200 |
| 102 | 16 | 24 | 51 | 204 |
| 104 | 16 | 24 | 52 | 208 |
| 106 | 16 | 24 | 53 | 212 |
| 108 | 16 | 24 | 54 | 216 |
| 110 | 16 | 24 | 55 | 220 |
| 112 | 16 | 24 | 56 | 224 |
| 114 | 16 | 32 | 57 | 228 |
| 116 | 16 | 32 | 58 | 232 |
| 118 | 16 | 32 | 59 | 236 |
| 120 | 16 | 32 | 60 | 240 |
| 122 | 16 | 32 | 61 | 244 |
| 124 | 16 | 32 | 62 | 248 |
| 126 | 16 | 32 | 63 | 252 |
| 128 | 16 | 32 | 64 | 256 |

Chapter 18: Maximum DP-Volume size

The maximum supported volume size within HNAS is 64 TiB. Do not create a DP-Volume in excess of 64 TiB using any SW configurator (technically, a thin provisioned DP-Volume of that size is only capable on Modular Storage as enterprise RAID limits the maximum DP-Volume to 60 TB)

Note: In the future, VSP G1000 and VSP Gx00 program products, such as Hitachi Universal Replicator and True Copy may permit upwards of 256TB DP-Volumes to be used. 68719476735

For example, when selecting the capacity of the DP-Volume, do not enter 64 TB, as the array will create a DP-Volume that is in excess of the maximum allowed SD capacity (64 TiB). In turn, HNAS will truncate the extra capacity and provide a warning message to that effect.

```
3960 Warning 2014-11-06 11:23:34-08:00 The capacity reported by
System Drive 2 exceeds the maximum supported by the server. The
server will restrict the SD capacity to 64TiB.
```

If you need to create the largest possible DP-Volume that can be used by HNAS, enter 137438953471 blocks, which is 64 TiB.

For a 32 TiB SD, choose 68719476735 blocks.

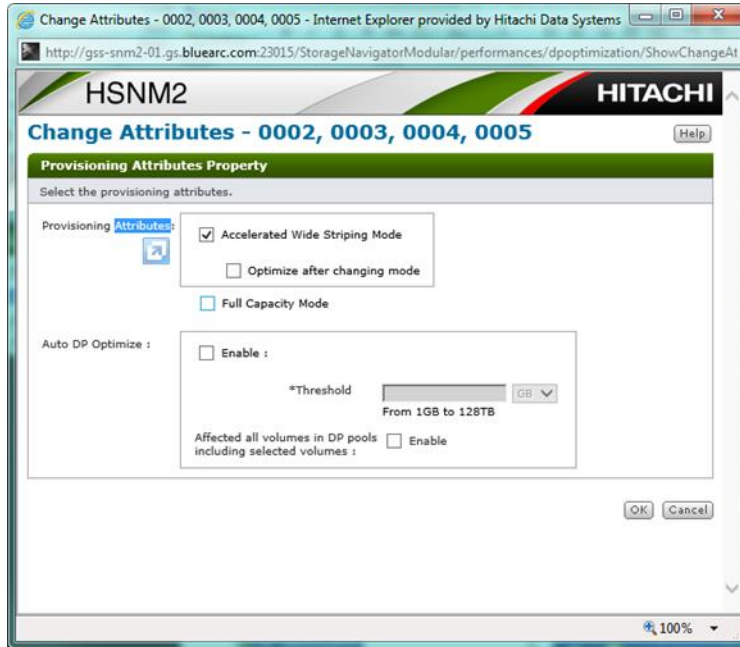
Chapter 19: Upgrading from prior releases and enabling Thin Provisioning

The following example covers a generic process of expanding an HDP pool that was previously thick provisioned to one that is thin provisioned. The process used here covers the HUS 1x0 series storage but is adaptable to the Enterprise arrays.

Upon upgrade to v12.1, complete the following steps:

1. Within SNM2, go to **Groups > Volumes > Format Mode** and click **Edit Format Mode**.
2. Change the Format Priority Mode to **Host Access**.
3. Lowering the priority will allow HNAS IO priority on the array. Note that zero initialization will take longer.

4. Uncheck **Full Capacity Mode** if in use (see figure below)



5. On the HNAS CLI type the following command

- a) `span-unmap-vacated-chunks --exhaustive `
- b) The array will begin the process of zero-initializing the unused pages within the HDP pool. This process will take time depending on the amount of space that can be freed.
- c) There can be a performance impact to the array (and on HNAS and any existing clients. You cannot quantify the impact of the operation.

On the CLI of HNAS, you can check the status of the vacated chunks:

```
$ span-vacated-chunks test
```

This span has no vacated chunks

An unmapper process has unmapped 3024 chunks and a total of 26433GiB

To monitor the progress, use the `span-list -s` command and observe the HDP free space grow.

“span-vacated-chunks” monitors the HNAS unmapping progress and “span-list -s” monitors the storage array’s zero initialization progress, which are two separate items, and may finish long after the HNAS unmapping process.

```
span-list -s
```

| Span instance name | OK? | Free | Cap/GiB | Chunks | Con |
|--------------------|-----|------|---------|--------|----------------|
| test | Yes | 99% | 32768 | 3750 x | 9382494749 90% |

On HDP pool 0 with 359GiB free

Set 0: 4 x 8192GiB = 32768GiB

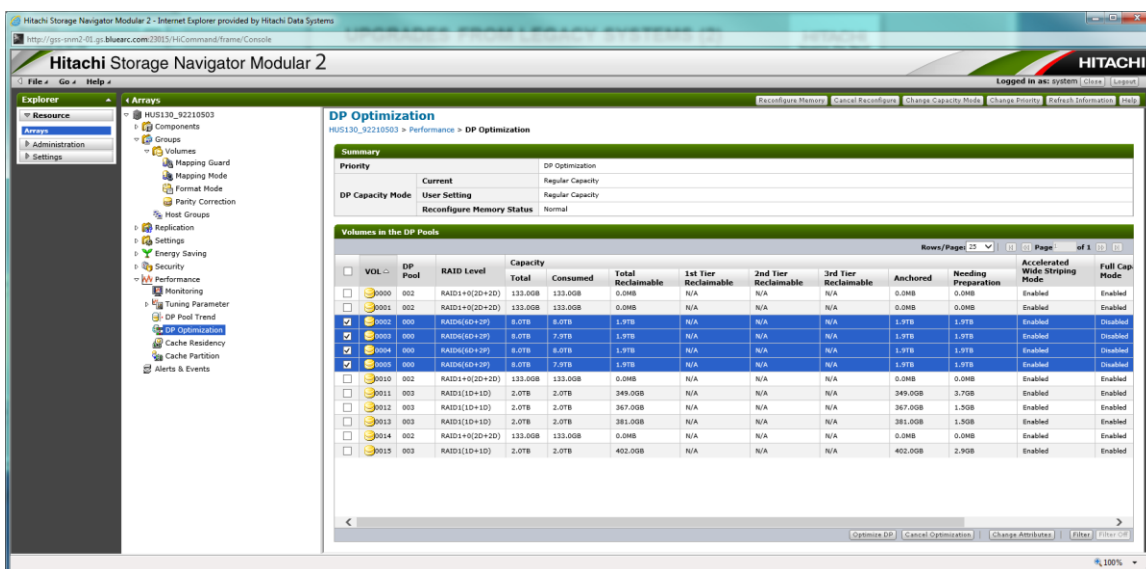
SD 0 (rack '92210503', SD '0002')

SD 1 (rack '92210503', SD '0003')

SD 2 (rack '92210503', SD '0004')

SD 3 (rack '92210503', SD '0005')

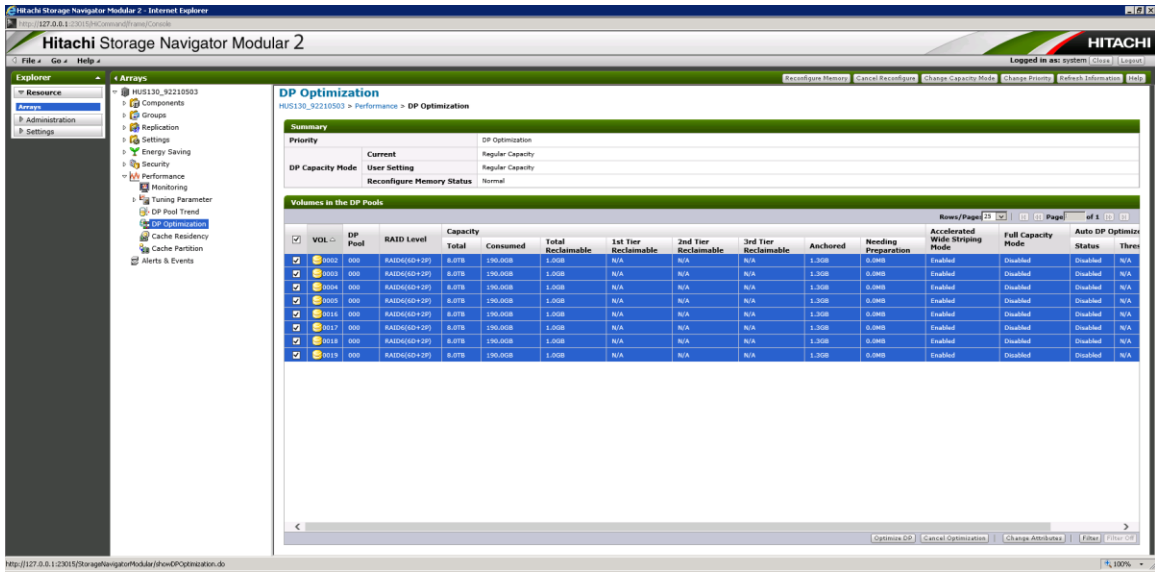
Monitor the “Total Reclaimable” and “Anchored” columns as the process takes place. When the “Total Reclaimable” space reaches 0 for each DP-Volume, the process has completed.



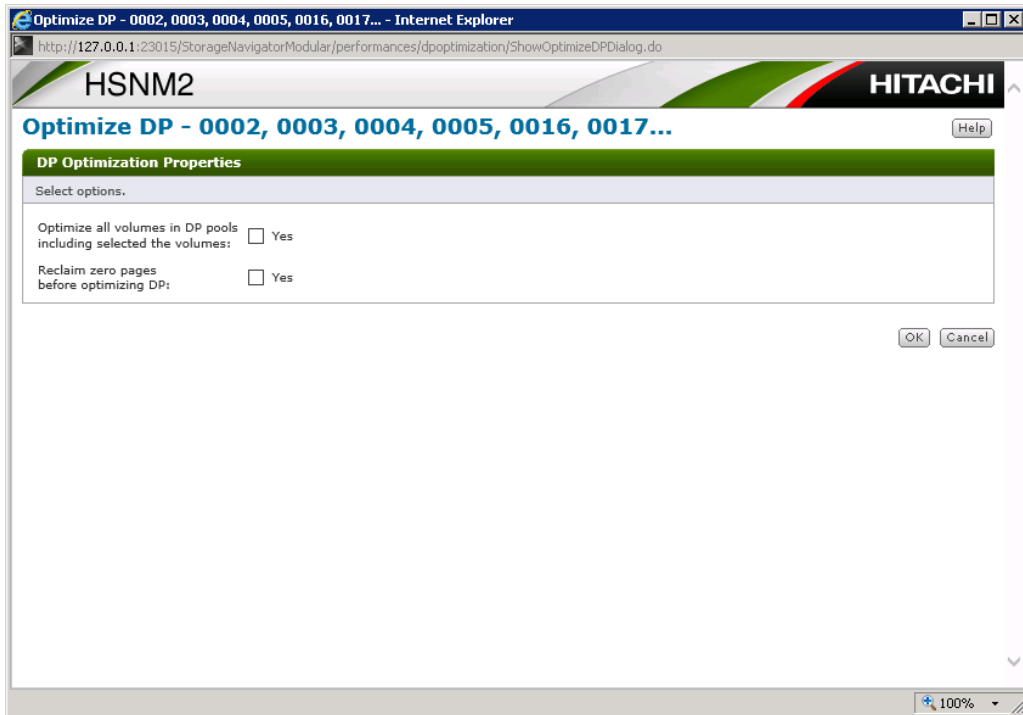
Using this test system as an example, the array (IO inactive) has reclaimed 15435 GiB in 17.5 hours for an approximate rate of .245 GiB/s.

Once the zero initialization completes on the HDP pool, the total capacity used on the HDP pool approximately equal the total size of the file systems on the span.

6. Select the desired DP-Volumes and click the “Optimize DP”.



7. Do not make any selections on the following page, simply optimize the DP-Volumes in the desired HDP pool:



Optimization of the pool will allow the pages to be rebalanced across the underlying raid groups and provide optimal performance. During this period, the performance of the storage array may be impacted.

Chapter 20: Considerations when deleting a span

The `span-delete` command allows you to delete DP-Volumes when deleting a span. Otherwise, some HDP pages will stay mapped to real disk space.

If you plan to delete the DP-Volumes on which the span resides, no special measures need be taken.

```
span-delete --will-delete-dp-vols <span-base-or-instance-name>
```

If you plan to reuse the DP-Volumes in a different span or with a foreign server, before you delete the span, it is important to ensure that no real disk space underlies the DP-Volumes. Otherwise, the disk space used by the original span will remain allocated and, unless exactly the same offsets are used by the new span or the other server, the DP-Volumes will run out of space prematurely due to the previously mapped pages.

See the for `span-delete` man page with `--reuse-dp-vols` for the appropriate process to reuse DP-Volumes. No matter which method is chosen, the process is interactive within the CLI and provides the exact steps necessary.

Chapter 21: Considerations when expanding a span

If the HDP pool is thinly provisioned, it is better to back-fill it by adding new pool volumes than to add DP-Vols and expand the span on to them. Otherwise, you must create new DP-Vols and add them to the span.

If you plan to re-use DP-Vols from a span that was previously deleted, do not forget to validate that the proper process has been performed in accordance with the `span-delete --reuse-dp-vols` sequence as referenced in the man page.

In case of uncertainty, you can determine whether a pool is thinly provisioned by running `span-list -s`, which shows both the free space on the pool and the free space on the stripesets that reside on the pool. If there is more free space on the stripesets than on the pool that hosts them, the pool is thinly provisioned.

1. Use `span-confine` to prevent file system auto-expansion on this span.
2. Add new pool volumes to the HDP pool. If available, enable the DP Optimize (as described earlier) option to have the data restriped over old and new media. The new pool volumes should be identical in performance to the existing ones: choose the same RAID level, disk type and capacity, and number of disks per pool volume.
3. If (and only if) the total capacity of the pool volumes now exceeds that of the DP-Vols on the pool, add new DP-Vols to the pool. There should be as many as were set up when the span was first created, and they should have the same capacity. You can check these figures by looking at the first stripeset in `span-list -s <span-instance-name>`. Once the DP-Vols have started formatting (which may not happen at once if you create or expand several HDP

pools within a short period of time), license them with `sd-allow-access` and add them to the span using `span-expand`.

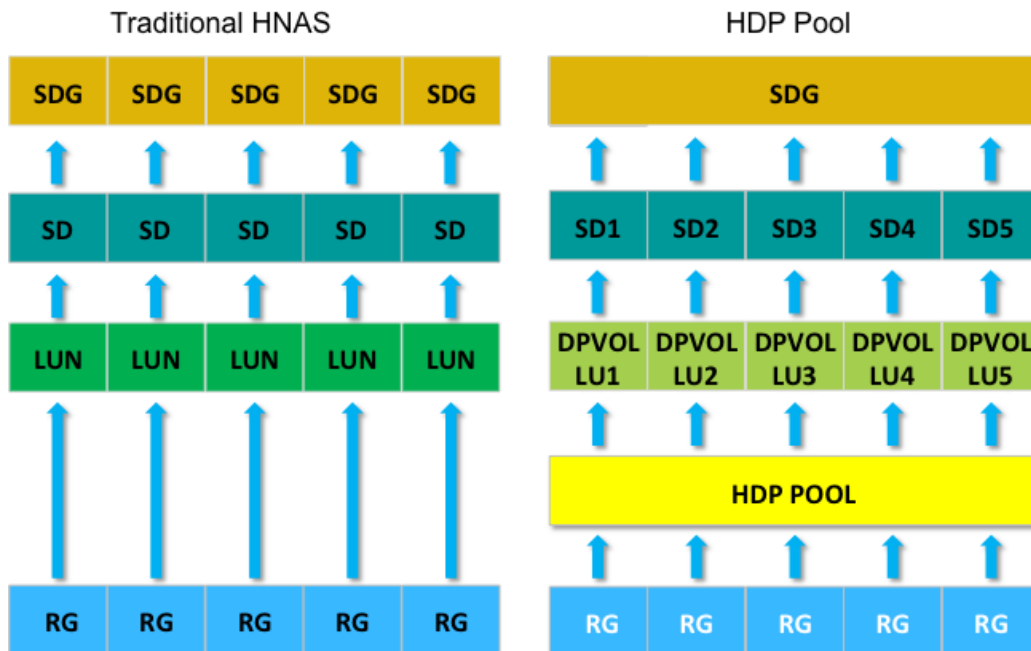
4. Wait until formatting is complete.
5. If desired, use `span-release` to permit file system auto-expansion. Alternatively, create or expand file systems by hand.

The amount of space you add can be as small as you want: even a single pool volume will make the pool faster and not introduce slow spots.

If using Modular storage, set the format priority mode to **host access** (rather than 'normal'). This will increase the time taken for formatting to complete but will reduce its performance impact. See [Pool Initialization and Formatting Recommendations](#) on page Chapter 16: Pool Initialization and Formatting Recommendations

Chapter 22: System Drive(SD) Group Considerations

HNAS release 11.2 auto-groups DP-Volumes from the same HDP pool into a single parallel SDG.



Prior to HNAS release 11.2, the best practice was to group each SD in its own SDG (which is no longer applicable in HNAS release 11.2 and later). As a result, changing the SD grouping method requires un-mounting of file systems in order to correct a system that was set up under the previous practice.

Note: SDs grouped into a parallel SDG can only belong to a single tier.

Because of this change, a user cannot “pin” a DP-Volume page to tier 1 of an HDT pool and attempt to create a tiered file system on this pool. Effectively, SDs from this singular HDT pool can be allocated on the array to separate tiers of storage, which could then be placed into an HNAS tiered file system. In v12.3, HNAS will support “pinning” of DP-Volume pages to Tier 1 storage.

In HNAS v12.3 and later, System Drive groups are no longer automatically created whatsoever. System Drive groups are considered optional except for when the configuration is a standard parity group that is divided into multiple logical units, residing on magnetic media. Obviously this type of poor configuration is not often deployed, but if one desires to configure the array in this manner, System Drive Groups are essential.

Chapter 23: Understanding free space on a Span or HDP Pool

The available free space on an HDP Pool = (Total Pool Capacity) – (Used space).

The act of creating a span consumes one HDP page per LU (DP-Volume) to store the COD.

The act of deleting a file system does not immediately free up capacity in the span because the file system will remain in the recycle bin. In order to immediately recover the HDP pool pages and reuse the capacity, issue the CLI command *file system-delete --no-undeletion-information <fs_name>*.

Once the file system is recycled, no space is released into the HDP pool or span because the chunks previously allocated to the file system go into the Vacated Chunk (VC) list.

The act of creating or expanding a file system draws space from the VC list first, if any is available, costing no space. Further space required is drawn from the HDP Pool.

While unmapping vacated chunks frees space in the HDP pool, this should never be done just to make the numbers line up.

Refer to the span-list –s, df, and span-vacated-chunks commands to measure span free space, file system free space, and capacity and number of vacated chunks.

```
<span-list -sv for pnode 1>
Span instance name OK? Free Cap/GiB Chunks Con Snap Cod fullness Cluster
Engrav Permanent Id DWB HDP free
-----
-----
SP1 Yes 68% 57836 3213 x 19327352832 90% 2%, 2%, 0% This
Dual 589812083FED61D9 Yes 873
On HDP Pool 1 with 873Gib free
Set 0: 4X12259GiB = 57836Gib, of which 39386GiB is free, 0Gib is vacated
SD 0 (rack '123173', SD '0010')
SD 1 (rack '123173', SD '0011')
SD 2 (rack '123173', SD '0012')
SD 3 (rack '123173', SD '0013')
fs fs1 Mount, EVS 1, cap 9225, con 0 , perm 589812F145258F0C, dev 1024, 32KiB, WFS-
2, DSBs: 128, DWB: Yes , Thin: No, AX: OK
fs fs2 Mount, EVS 1, cap 9225, con 0 , perm 589812EE051B61AE, dev 1024, 32KiB, WFS-
2, DSBs: 128, DWB: Yes , Thin: No, AX: OK
```

- Total amount of free space on all the HDP pools that comprise the Span
- Total amount of free space on the individual HDP pool that comprises this stripeset

- Total of unused chunks in the stripeset
- Total capacity of vacated chunks

Chapter 24: Downgrade Considerations v12.1 to v12.0

The vacated-chunks (VC) list is stored in the chunk table, a region of COD. It needs a new COD schema, because v12.0 does not support a VC list.

To downgrade to v12.0:

Use *span-unmap-vacated-chunks* and wait until the VC list is empty

- You do not need to wait for HDP's `zero-init` process to complete

Chapter 25: Upgrade Considerations

As part of an upgrade to v12.1, check that the current configuration does not violate the stripeset configuration previously mentioned in this document.

No span can contain a mixture of DP-Volumes and plain parity groups (though, on a single cluster, some spans can use DP-Volumes while others use parity groups).

No stripeset can contain DP-Volumes from more than one DP pool. (Every new span contains one stripeset; a further stripeset is added every time the span is expanded.) However, different stripesets within the same span can optionally reside on different DP pools. This includes Tiered File Systems where a Tier X is comprised of SDs from Parity Groups and Tier Y is comprised of SD from and HDP Pool(s)

Failure to address these prior to upgrade will render an upgraded system unable to create, expand, or delete a file system on a span in violation.

HNAS v12.3, improves the usability of spans that violate the prior restrictions. The *span-hdp-thickly-provisioned* command allows the restriction to be relaxed at the expense of not being able to thin provision the span. This command only releases the restriction - the pool should ultimately be reconfigured to meet best practices.

Review the [Recommended Array Settings](#) to ensure that host mode and appropriate SOMs are configured.

Note: If Full Capacity mode on DP-Volumes is enabled with v12.1, HNAS will see that all capacity has been allocated and will not (cannot) create or expand a file system.

Attempting to do so will result in the following error:

```
file system-create -b 32 test t4 100
```

```
Creating the file system...
```

```
Failed: Can't create or expand the file system: the host span has too few free chunks
```

Upon upgrade to v12.1, the server will notice there is no free space left on the HDP pool. Therefore, after the upgrade, do the following:

1. In SNM2, go to **Groups > Volumes > Format Mode** and select **Edit Format Mode**.
2. Change the Format Priority Mode to “Host Access”. Lowering the priority will allow HNAS IO priority on the array, however zero initialization will take longer.
3. Select the desired DP Vols and disable Full Capacity Mode.
4. On the HNAS CLI, run the command:

```
span-unmap-vacated-chunks -exhaustive <span>
```

This process will vacate any chunks from the span, thus returning any free space to the HDP pool. If there is no space to return that means the span is already full and the only way to expand the underlying file systems would be to add more storage.

- It is at this point that the array will begin the process of zero-initializing the unused pages within the HDP pool. This process will take time depending on the amount of space that can be freed.
- To monitor the progress use the command `span-list -s` and watch the “HDP “free” space grow.

Chapter 26: Rolling Upgrades/Downgrades

Provisioning of storage is not permitted during a rolling upgrade.

During a rolling downgrade the system will not boot into v12.0 or earlier while any span has a non-empty VC (vacated-chunks) list. However, one could shut down the entire cluster and reboot into 12.0 if a span had a non-empty vacated chunks list. Upon power on, the HNAS event log will warn about the filesystem using too recent a COD schema, additionally “span-list” would warn that the `filesystemd` had not loaded from COD. See Chapter 24: Downgrade Considerations v12.1 to v12.0

Hitachi Vantara

Corporate Headquarters 2535 Augustine Drive
Santa Clara, CA 95054 USA www.HitachiVantara.com community.HitachiVantara.com

Regional Contact Information

Americas: +1 866 374 5822 or info@hitachivantara.com

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

Asia Pacific: +852 3189 7900 or info.marketing.apac@hitachivantara.com

