Hitachi Dynamic Provisioning

Unofficial Best Practices / ROT / Interoperability

V1.1

Contents

1. Introduction

It is intended that this document will help you in your deployment of HDP/ThP, and give you a broad understanding of the technology as well as help you ask the right questions of your HDS and HP representatives.

This document is written to correspond as much as possible with ucode V03+1 on a HDS USP-V and HP XP24000. It will be updated accordingly, by the author, on a best effort basis.

Cross-reference of important technology names between HDS and HP

HDS name	HP name
HDP	ThP
USPV	XP24000
DP-VOL	V-VOL
ShadowImage	Business Copy
TrueCopy	Continuous Access

The above are registered trademarks of Hitachi Ltd, HDS and HP.

Throughout this document the author will use the HDS terms.

> **DISCLAIMER:** At the time of writing this document, I do not work for Hitachi, HDS or HP. The information contained within this document is from my personal knowledge and experience with Hitachi Dynamic Provisioning software, and is in many instances my own opinion. As such it should not be considered authoritative. Your HDS or HP representatives should hopefully have the latest and greatest Best Practice advice.
>
> Also, the information in this document of course subject to change and being rendered out of date due to new releases of microcode and the likes. However, all effort is made to keep it up to date and useful to those who work with HDP and help us all implement and use HDP appropriately.

For more information from the author regarding HDP, see –
http://blogs.rupturedmonkey.com/?cat=18

For more information from the author concerning storage technologies, especially relating to Hitachi (HDS, HP) storage, see –
http://blogs.rupturedmonkey.com

To contact the author directly – nigelATrupturedmonkeyDOTcom

2. POOL Pre-requisites and Restrictions

- The required Shared Memory DIMMs **must** be installed to the DKC to enable HDP to work. HDP has dedicated DIMMs for the storing of HDP constructs such as the Dynamic Mapping Table (DMT) etc during HDP operations. If they are not physically installed it will not work.
- All LDEVs used to create a Pool must be OPEN-V and between 8GB and 4TB in size.
- Each Pool can be comprised of a maximum of 1024 LDEVs (Pool-VOLs).
- A maximum of 128 Pools can be created per subsystem. This is the combined total of HDP and COW Pools.
- A Pool can be between 8GB and 4096TB.
- All Array Groups used to build a Pool must be mapped to the same CLPR.
- An LDEV cannot be added to a Pool if it has a path definition (that is, presented to a front end port).
- Once an LDEV is added to a Pool, as a Pool-VOL, it cannot be removed without first deleting the **entire** Pool. **Double check before adding Pool-VOLs!**
- You cannot delete a Pool that has DP-VOLs associated with it.
- 8192 DP-VOLs can be associated with each Pool
- Only OPEN-V volumes can be created
- Once a DP-VOL is created it cannot be increased in size without a **Volume to Space** operation and then recreating the DP-VOL. *As of microcode v03+1 the ability to dynamically expand volumes may be available for Windows 2008. This functionality will likely be extended to more Operating Systems in the future*. *This also requires there to be free space in the V-VOL Group immediately after the DP-VOL*. *See DP-VOL Size Recommendations section for further details.*
- It is recommended that Windows servers be placed in Host Storage Domains with a mode setting of 2C (Windows Extension). This is more future proof than 0C and will be a pre-requisite for future enhancements to HDP and other HDS software. No additional overhead is incurred by using mode 2C.

3. POOL Performance

HDP Pools can be configured in many ways, and the optimum configuration for one environment may not be the optimum configuration for another. This section outlines 2 areas of best practice in relation to Pool performance –

3.1 **Global Best Practices**, which should apply to most, if not all, configurations.
3.2 **Local Best Practices**, which should be tuned according to specific requirements

3.1 Global Best Practices for Pool Configuration

To achieve the most *consistent* and *predictable* performance –

- The more physical spindles that sit behind a POOL, the better the performance of that Pool. This translates to Array Groups. The more the better. Recommended quantities of Array Groups in a Pool are in increments of 4. E.g. 4, 8, 12, 16, 32.....
- Use and balance Array Groups that are from different DKA/BED pairs or different external controllers where possible.
- Do not intermix internal and external LDEVs in the same Pool, even if in the same CLPR.
- Each Array Group should comprise a single large LDEV occupying the entire space of the Array Group*.
- A single Array Group should never contain both normal LDEVs and Pool-VOLs.
- When adding additional space to a Pool, it is recommended that you add additional space in the same quantity and configuration as the original allocation (same number of underlying AG's, RAID type, spindle speed and capacity….). This will keep striding, striping and performance consistent.
- The first time that a Pool is created, and then each time a full new set of LDEVs are added, the "Optimise" button should be used. This ensures striping over all of the Array Groups, including the newly added space.
- Assign DP-VOLs to the same CLPR as the Array Groups that are used to build the associated Pool.
- Assign Pools to the same CLPR as the Array Groups that were used to create the Pool.

3.2 Local Best Practices for Pool Configuration

This section will attempt to explain some of the theory behind **why** you might configure a Pool in a certain way. Three possible Pool configurations are presented and addressed –

3.2.1 Homogeneous Pools

Homogeneous Pools are defined (by the author) as Pools in which all Array Groups/LDEVs comprising the Pool are configured identically. E.g. –

- Common RAID level
- Common spindle type – FC/SAS/SATA/SSD....
- Common spindle capacity
- Common spindle rotational speed (rpm)

Advantages - Homogeneous Pools allow for the most predictable and consistent performance. All spindles in the Pool will have the same performance characteristics and overhead. No single disk is guaranteed to become a bottle-neck or slow-spot due its inherent characteristics.

Disadvantages - ?

3.2.2 Heterogeneous Pools

Heterogeneous Pools are defined (by the author) as Pools which contain Array Groups/LDEVs of differing configurations. E.g. –

- Differing RAID levels
- Differing spindle types – FC/SAS/SATA/SSD
- Differing spindle capacities
- Differing spindle rotational speeds (rpm)

Advantages – If you have subsystem with a mixture of spindle and Array Group configurations, you *may* get improved Pool performance by assigning as many LDEVs/Array Groups to the Pool as possible. For example, you may have the following Array Groups free in your subsystem –

- 4 x RAID1 (4+4) on 146GB 15K FC
- 4 x RAID5 (7+1) on 300GB 10K FC

By adding all of the above 8 Array Groups into a single Pool you will have the IOPs performance of 44 data spindles. Whereas if you created a Homogeneous Pool using only the RAID1 (4+4) Array Groups, you would only have the IOPs performance of 16 data spindles. Under certain circumstances, spindles win prizes, so the more the better.

Disadvantages - Performance will be less predictable and consistent. Intermixing different spindle types and RAID configurations within a single Pool introduces varying performance characteristics and levels of overhead to that Pool. As a result,

performance is more likely to rise or fall depending on which disks are handling an IO. Larger capacity disks will also have more Pool Pages and as a result have a higher access density making them more likely to become hot-spots/slow-spots.

Authors Opinion: It is the authors own opinion that creating Homogeneous Pools is the best practice.

3.2.3 Multiple Pools

Coming Shortly.....

Other notes

- A single Pool can usually handle random and sequential workloads, such as database file and log file activities, without a noticeable performance impact. It is not a de facto requirement to have separate Pools to isolate such workloads. However, a single DP-VOL should not be assigned both random and sequential workloads.
    - It may, however, be beneficial to have separate Pools for heavily conflicting applications. An example being - putting an application which saturates disk overnight, during defrags etc, on the same Pool as servers that run critical overnight batch jobs or staging backup to disk would not be recommended.
    - You may also want to place ShadowImage P-VOLs and S-VOLs in separate Pools.

* If the capacity of the Array Group is larger than the maximum size of a single LDEV, multiple large LDEVs should be created to occupy the entire space of the Array Group. There is supposedly no more requirement to create multiple LDEVs per Array Group to assist *Striding*.

4. DP-VOL size recommendations

- As of ucode V03+1 it will be possible, under certain circumstances, to dynamically expand the size of a DP-VOL. To allow for this feature, it is necessary to have only a single DP-VOL within and V-VOL Group. *
- If creating a very large number of small DP-VOLs it may be worth sizing them in multiples of 42MB in alignment with HDP Page size.
- It is often good practice to stick with any existing fixed LUN sizes as this allows for array based migration jobs through UVM.

* Initially it is expected that to be able to dynamically expand a DP-VOL will only be available for the last DP-VOL in a V-VOL Group. This may change with future versions of ucode. For detailed explanation of this, and the theory behind it, see the authors following article – http://blogs.rupturedmonkey.com/?p=190

V-VOL groups will be 4TB in size in future releases of ucode.

5. Other Considerations

Monitoring

- Currently the only practical level of alerting is at the Pool level. DP-VOL alerting is not considered useful at the time of writing this document. Future releases of ucode and possibly HTnM version 6.0 may resolve this.

The following items should be considered if there is a requirement to utilise the oversubscription feature of HDP -

- NTFS partitions should be formatted with the Windows **Quick Format** option. A Windows **Full Format** on Windows Server 2008 beta version has been seen to consume all possible pages for a DP-VOL.

- Filesystem defrags should be avoided wherever possible (***online*** application defrags, as seen in Exchange, SQL Server, Active Directory and Oracle are fine as they do not walk all over the filesystem).

- Veritas Volume Manager is the recommended tool to be used when migrating old systems from traditional LDEVs to DP-VOLs as this is friendly to the oversubscription feature of HDP.

- It is recommended to use appropriate application tools to shrink disk space usage as much as possible before using tools to migrate old systems to DP-VOLs

The Windows NTFS file system is not HDP “Thin” friendly. This is because the NTFS file system uses a form of Least Recently Used (LRU) algorithm when choosing clusters/blocks to allocate to writes. Basically, even after deleting data from an NTFS file system, subsequent writes will be written to previously unallocated filesystem blocks/clusters in preference to the space released by delete operation. This behaviour assists the Microsoft Undelete functionality of NTFS and is at the time of writing not configurable.

6. Further Information

- The Dynamic Mapping Table (DMT), V-VOL Management area and other HDP constructs are held in dedicated Shared Memory DIMMs that must be installed for HDP to work. Therefore, the impact of HDP on the Shared Memory and Cache Memory systems is minimized. This is the same as with COW. If the dedicated DIMMs are not installed, the product will not work.
- In the event of a PS-OFF, the DMT, stored in Shared Memory is de-staged to the SVP hard disks (assuming Mode 460 is set) and also to the reserved area of the first HDP Pool created in the system.
- Initially DP-VOLs do not consume any space.
- After performing a Windows **Quick Format** (default cluster size, NTFS) on a 5GB HDP volume, it registered as consuming 6% of its 5GB space. After the same Windows **Quick Format** operation on a 25GB HDP volume, the Storage Navigator GUI reported as consuming 2% of the available space.
- Read operations directed to an area of a DP-VOL that is not already allocated space do not cause space to be consumed.
- On Windows Server 2003 a full format consumes the same amount of space as a quick format. However, Windows 2008 has been seen to consume all pages of a DP-VOL when a full format is performed. This was on a pre-release version of Windows 2008!

For more information regarding the 42MB Page size see - http://blogs.rupturedmonkey.com/?p=182

For more general information from the author regarding HDP see – http://blogs.rupturedmonkey.com/?cat=18

For more general information re storage, and in particular HDS and HP storage, see - http://blogs.rupturedmonkey.com

To contact the author directly - nigelATrupturedmonkeyDOTcom

Feedback welcome!