VERITAS

Veritas NetBackup Flex Appliances Best Practices

This document provides best practice recommendations to achieve optimized performance for backup, restore, duplication, and replication workloads on NetBackup™ Flex Appliances.

White Paper | June 2023

Contents

Executive Summary

Veritas NetBackup™ Flex Appliances allow you to configure multiple NetBackup primary, media, MSDP-Cloud, and WORM storage server instances on a single hardware platform. Each NetBackup instance runs in a container and shares the hardware resources such as CPU, memory, I/O, and network.

Evaluation, planning, deployment and post deployment are the four important phases to optimize Flex Appliance performance.

Evaluate	Planning	Deployment	Post Deployment
▪ Workloads and compute requirements ▪ Assess existing or new infrastructure ▪ HA vs stand-alone options ▪ Assess migration needs	▪ Number of instances ▪ Type of instances ▪ Instances storage layout	▪ Configure application instances ▪ Apply add-ons as needed ▪ Tuning such as MSDPCache	▪ Performance monitoring ▪ Tuning if needed ▪ LUN sharing impact during re-size

Pre-Design → Design → Solution → Optimize

Tool Kits

- Sizing Tools—USE (internal sales engineers)
- Migration—NetBackup Primary Migrator
- Flex Deployment Best Practices Guide
- Utility for NetBackup "touch" files management
- Tools for LUN Migration and Instance Topology Performance Data Analyzer

This paper will provide guidance and best practices for sizing, tools, and strategies for maximizing performance and ROI for NetBackup Flex Appliances.

Evaluation

Evaluation is the critical step to ensure proper sizing and achieve optimized performance for data protection. Veritas Sales engineers can help you choose the right appliance platform and configurations based on workload requirements, high availability, and migration needs.

The following workload information is needed for the sizing tool:

- Workload types and sizes
- Data characteristics (third-party encryption, millions of small files)
- Backup methodology and retentions
- Data lifecycle and secondary operations (optimized duplications, auto image replication [AIR], or tape-out)
- Backup windows
- Recovery time objective (RTO), recovery point objective (RPO), and service level agreements (SLAs)
- Finite system resources (CPU, memory, network)
- Storage (the number of physical disks and associated LUNs)

Planning

Sizing Flex Container Instances

All Flex Appliance instances share hardware resources, IO, disk space, CPU, and memory. When determining the number of Flex container instances to deploy per Flex node, we need to plan the resource, making sure not to oversubscribe.

Primary Server Instances

A NetBackup Primary Server instance tends to be I/O-intensive. We recommend 40 TB storage LUNs with 4 TB disks or 80 TB storage LUNs with 8 TB disks for Flex appliance 5350. We recommend creating Primary Server LUN first, and group Primary server instances on the same LUN.

MSDP and WORM Server Instances

When sizing MSDP or WORM server instances, review the factors below to plan out the LUNs allocated for each instance:

- Workload types
- Front-end terabytes (FETB)
- Data growth and change rate
- Storage lifecycle policy (SLP) backlogs for secondary operations
- Maintenance requirements

Below are some best practices on LUN allocation:

Grouping workload types: Optimizes dedupe performance as well as simplifies the solution to make predictive analysis and trending easier

Use a conservative dedupe rate—80 percent: The deduplication rate depends on the data change rate and characteristics

Secondary operations: Duplication, replication via AIR, and duplication to tape use memory, CPU, and I/O resources. Duplications to tape are particularly resource-intensive because images on dedupe storage must be rehydrated as part of the duplication process to tape. This process results in intensive read operations. Additionally, Flex instances that have Advanced Disk and MSDP pools will generate intensive read and write operations when specific workloads such as database transaction logs are written to an Advanced Disk pool prior to being duplicated to an MSDP pool. These types of intensive read and write operations will use significant I/O and CPU resources.

LUN Size Recommendations

- 4 TB drives shelf presents 40 TB LUNS, 8 TB drives shelf presents 80 TB LUNS
- Create/grow instances in 40 TB/80 TB increments respectively to avoid risk of LUN sharing
- 4 TB drive shelf for smaller instances can improve storage utilization and reduce need for LUN sharing
- 8 TB drive shelf for larger instances preserves overall footprint and maximum capacity

MSDP Memory Calculation

MSDP is memory intensive. To ensure MSDP instances don't run into a memory contention issue, plan for the MSDP cache to use 1 GB of memory for every 1 TB of disk storage. For example, an MSDP pool that is 480 TB in size should have at least 480 GB of memory resources to allow for optimal MSDP cache operations. Flex instances that are consistently busy with backup and secondary operations should also have enough memory above the MSDP cache requirements to accommodate the resources required to process the workloads.

Please note: MSDP changes memory lookup scheme that significantly reduce the memory usage beginning in 10.1, please refer to this section for more detail: Tuning MSDP-Direct Cloud Tier.

NetBackup Appliance to Flex Consolidation Considerations

NetBackup Appliance to Flex consolidation initiatives are becoming more common as companies realize the efficiency and security benefits of the Flex Appliance. As they plan for such consolidation efforts as part of a hardware refresh, correct sizing of the target Flex environment is paramount to support existing and future workloads successfully. We recommend engaging Veritas Sales Engineers to help you with the migration.

Unlike a typical sizing effort for a new Flex environment, an existing NetBackup Appliance footprint that will be consolidated with new hardware on the Flex platform requires additional information and planning. Because there are existing workloads that are being processed by the legacy NetBackup Appliance solution, it is critical to gather the following key information about the existing environment prior to sizing the new target Flex environment.

The number of NetBackup Appliances being consolidated, including:

- Model
- Disk pool size(s)—MSDP and/or Advanced Disk v % of capacity in use per disk pool
- Memory in GB
- Type and number of CPU cores
- Network info (speed and number of interfaces)
- Workload types and FETB
- Current dedupe rates
- Average rate of change per day
- Expected growth per year
- Current backup window
- Current SLAs, RPO, and RTO
- All secondary operations (optimized duplications, AIR, duplication to tape)
- Special data characteristics (third-party encryption, millions of little files)

Flex 5250 Appliance Cloud Tiering Sizing

The Flex 5250 Appliance is a small configuration appliance that supports cloud tiering and immutable storage. The smallest Flex 5250 configuration consists of internal storage (no external shelves) with 9 TB of usable disk capacity and 64 GB of memory. This configuration can support the following configurations

- A single NetBackup Primary Server container and a single NetBackup Media Server container with a standard MSDP pool
- MSDP cloud-tiering, which should not run concurrently with backups
- WORM Storage server

When cloud tiering is in use, change the MaxCacheSize NetBackup parameter to 30 percent; otherwise, leave it at its default value.

The backup configuration using server-side deduplication has been tested with up to 90 concurrent backup streams before seeing errors. Maximum throughput was achieved with only 16 concurrent streams, however, so we recommend not exceeding 16 concurrent backup streams using server-side deduplication. Using client-side deduplication reduces the resources required on the appliance and

can likely support more concurrent backup streams. Organizations should always use the lowest number of backup streams required to achieve maximum throughput. In this particular case, if 16 concurrent streams can meet your throughput requirement, then there is no additional performance gain by running more than 16 concurrent streams.

When cloud tiering is in use we recommend changing the MaxCacheSize to 30 percent to allow the cloud upload to go through memory cache. Our performance testing showed that MaxCacheSize greater than 30 percent resulted in insufficient memory cache left for cloud upload. As a result, cloud upload will go through disk cache, which could cut down cloud upload performance by 50 percent.

Deduplication Consideration

The backup configuration with server-side deduplication has been tested with up to 90 concurrent backup streams. Maximum throughput was achieved with 16 concurrent streams. We recommend using up to 16 concurrent backup streams when you use server-side deduplication.

Client-side deduplication reduces the resource requirements on the appliance and can support more concurrent backup streams. Organizations should always use the lowest number of backup streams required to achieve maximum throughput.

Concurrent streams: 16 for peak throughput, tested up to 90 without job errors

- Client-side dedupe allows more concurrent streams
- 1 TB free space required per cloud tier
- MaxFileSize (MSDP data container size): 64 MB (default value)
- MaxCacheSize: 30 percent (only change if using cloud-tiering)

Recommended Configuration of a Flex 5250 Appliance with MSDP-C and No External Storage

- 36 TB capacity
- 64 GB memory
- One Primary Server
- One Media Server
- Standard MSDP pool (or immutable)
- Cloud tier to S3 target(s)

Please note: MSDP changes memory lookup scheme that significantly reduce the memory usage in 10.1, please refer to this section for more detail: Tuning MSDP-Direct Cloud Tier.

Flex Appliance Tuning

MaxCacheSize is an MSDP parameter that controls MSDP instance fingerprint cache size to limit the maximum amount of memory a media instance can use for fingerprint caching, and reserves enough RAM for the operating system and application processes.

Without this limitation, MSDP can consume an excessive amount of RAM for fingerprint caching and cause memory starvation for other processes running on the system. Excessive swapping can happen and slow down overall system performance.

Default MSDP Setting

Because all instances share hardware resources on Flex Appliances, proper tuning can help optimize performance and leave enough memory for cloud upload memory cache, OS, and other NetBackup processes also running on the system.

Before we discuss how to customize MaxCacheSize, Table 1 shows the default manufacture MaxCacheSize setting.

Version Date:	Before NetBackup 8.2	NetBackup 8.2 and later
Each Media MSDP container fingerprint cache size (MaxCacheSize)	10 percent of system memory	50 percent of system memory

Table 1. Default MaxCacheSize setting

For example, the Flex 53x0 Appliance is shipped with 768 GB of memory by default. If you apply the memory upgrade kit that doubles the physical memory to 1.5 TB and vmstat command output shows large amount of free memory, you can relax the 50 percent limitation, increase the MaxCacheSize to the instances that have lower than expected dedup ratio. Make sure to leave enough memory for other processes also running on the system, especially if there are MSDP-C instances configured, and if hundreds of concurrent jobs may run on the system. Reserve at least 150 MB RAM for each concurrent job running. For memory requirements with MSDP-C instances, refer to the section: Tuning MSDP-Cloud direct Tier.

Media MSDP Instance Memory Tuning

The MaxCacheSize setting determines how many fingerprint indexes can be cached in memory per instance, which can potentially influence the deduplication ratio. Any generated index would be identified as a new index if there is no matching index found in the cache. When the system lacks cache space, the least recently used index will be deleted for the new index. When MaxCacheSize is set too low, this scenario will happen more often. Fingerprint cache miss increases the write I/O to the storage pool, decreases the deduplication ratio, and slows down the overall backup performance.

MaxCacheSize

The following example shows how to calculate MaxCacheSize based on storage size.

The total amount of RAM required for a media MSDP intance on a Flex Appliance is based on 1GB of RAM for each TB of storage. We recommned using no more than 50 percent of the RAM for fingerprint caching.

Example:

If the storage allocated for a media/MSDP instance is 80 TB, the total RAM required to run the instance would be 80 GB. Of the 80 GB RAM, we recommend using no more than 50 percent for fingerprint caching. The fingerprint cache should be set at 40 GB RAM. A Flex 53xx Appliance by default is configured with 768 GB of RAM; 40 GB is roughly 5 percent of the 768 GB system RAM. Therefore, the MaxCacheSize should be set to 5 percent. The 5 percent is derived and rounded based on the formula (40GB/768GB) * 100) %.

Note: The 50 percent MaxCacheSize setting is a best practice, not a firm requirement. The ratio will hold only if the size of RAM on the appliance is exactly 1 GB of RAM per TB of storage. In actual customer deployments, the RAM size could be either greater or less than 1 GB per TB of storage. Therefore, the aggregate MaxCacheSize for all media instances will be either less than or greater than 50 percent, respectively. It is normal for the aggregate MaxCacheSize to be slightly higher than 50 percent of the RAM, especially in a newly deployed Flex Appliance, because the deduplication engine allocates fingerprint cache as needed. In a fresh deployment, the number of fingerprints will be small and only a small portion of the MaxCacheSize will be used for caching. As the storage pool is filled up, more RAM will be consumed to cache the increasing fingerprints up to the MaxCacheSize. When the aggregate MaxCacheSize actually consumed is much higher than 50 percent, the appliance will start to show signs of memory starvation, such as increased swapping activities and a slowdown of overall system performance.

MaxCacheSize Tuning Examples

This section gives two examples of MaxCacheSize configurations for a Flex Appliance with 4 TB and 8 TB storage shelves. Both examples have the RAM size much lower than the recommended 1 GB of RAM per TB of storage, therefore the aggregate MaxCacheSize is approximately 60 percent. For the first example, we recommend adding additional memory. For the second example, we recommend deploying a Flex HA Appliance and separating the workload into two containers, each running on a separate head node.

Please note: MSDP changes memory lookup scheme that significantly reduce the memory usage in 10.1, please refer to this section for more detail: Tuning MSDP-Direct Cloud Tier

Parameter	1 MSDP	2 MSDPs	4 MSDPs	6 MSDPs
MSDP Pool Size	MSDP 1: 960 TB, 24 LUNs MaxCacheSize calculation: 960*0.5=480GB (480/768GB) * 100 = 62.5	MSDP 1: 480 TB, 12 LUNs MSDP 2: 480 TB, 12 LUNs	MSDP 1: 240 TB, 6 LUNs MSDP 2: 240 TB, 6 LUNs MSDP 3: 240 TB, 6 LUNs MSDP 4: 240 TB, 6 LUNs	MSDP 1: 160 TB, 4 LUNs MSDP 2: 160 TB, 4 LUNs MSDP 3: 160 TB, 4 LUNs MSDP 4: 160 TB, 4 LUNs MSDP 5: 160 TB, 4 LUNs MSDP 6: 160 TB, 4 LUNs
MaxCacheSize per MSDP Instance	60% (rounded from 62.5)	30%	15%	10%

Example 1. 4 TB disk drives with 768 GB of RAM

Flex Appliances support 1.9 PB storage; each media server instance supports no more than 960 TB . You can create two media server instances, each with 960 TB.

Parameter	1 MSDP	2 MSDPs	4 MSDPs	6 MSDPs
MSDP Pool Size	MSDP 1: 1920 TB, 24 LUNs 1920/2=960GB (960/2*768 GB) * 100 = 62.5	MSDP 1: 960 TB, 12 LUNs MSDP 2: 960 TB, 12 LUNs	MSDP 1: 480 TB, 6 LUNs MSDP 2: 480 TB, 6 LUNs MSDP 3: 480 TB, 6 LUNs MSDP 4: 480 TB, 6 LUNs	MSDP 1: 320 TB, 4 LUNs MSDP 2: 320 TB, 4 LUNs MSDP 3: 320 TB, 4 LUNs MSDP 4: 320 TB, 4 LUNs MSDP 5: 320 TB, 4 LUNs MSDP 6: 320 TB, 4 LUNs
MaxCacheSize per MSDP Instance	60% (rounded from 62.5)	30%	15%	10%

The Deduplication Multi-Threaded Agent (mtstrmd) is designed to improve MSDP backup performance for systems with a low number of concurrent jobs. The low number of concurrent jobs is different for different appliance models. Roughly, the number of concurrent jobs that is lower than a quarter of the number of CPU threads in the appliance is considered low. For example, a 5340 appliance has 64 CPU threads, so the number of concurrent jobs below 16 is considered low. Each media server instance has its own mtstrmd that is enabled by default. The default mtstrmd session number is automatically calculated at the time of installation and is captured in the mtstrmd.conf, which is under the parameter MaxConcurrentSessions. The mtstrmd processes are CPU intensive. Too many mtstrmd sessions can cause a CPU bottleneck and degrade the overall backup performance.

The default session number works well for a single media instance on a Flex Appliance. However, when deploying multiple media instances on a single Flex Appliance head node, you should reduce the MaxConcurrentSessions for each instance. The aggregate number of sessions should not exceed the MaxConcurrentSessions listed in Table 2.

Flex Appliance Model	MaxConcurrentSessions
Flex 5150	3
Flex 5250	9
Flex 5340	15
Flex 5350	19

Table 2. Aggregate MaxConcurrentSessions limits

For example, if you deploy three media server instances with similar workload and performance requirements on a single Flex 5340 Appliance, we recommend you set the MaxConcurrentSessions to five for each media instance. If one of the media instances has a high-performance requirement, set the MaxConcurrentSessions to 15 on that media instance and disable the mtstrmd on the other two media server instances. Here are instructions for changing the MaxConcurrentSesions parameter:

To change the MaxConcurrentSessions number, log into the media container, edit the value inside /usr/openv/lib/ost-plugins/mtstrm.conf and then kill the mtstrmd process inside the container when there's no job using mtstrmd. It will start automatically when the new job kicks in, and the new MaxConcurrentSessions setting will take effect.

Media MSDP Instance Storage Allocation

The storage allocation plays an important role in media MSDP performance. Simply allocating enough storage to meet the size requirement of each instance is not sufficient for optimal performance.

When multiple MSDP containers share the same LUN(s), multiple backup jobs from the containers can write to the same LUN concurrently and result in I/O contention. The I/O contention degrades performance, especially for the I/O-intensive workload.

Avoiding or reducing multiple instances sharing the same LUN is critical to reduce I/O contention and achieve optimal I/O performance.

LUN Creation Sequence

A fully populated Flex 53xx storage shelf such as RBOD/EBOD is configured with 6 x RAID6 LUNs. Each RAID6 LUN is built with 13 disk drives or 11 data disks and two parity disks called (11D+2P) LUN. A half-populated shelf is configured with 3 x RAID6 LUNs. Figure 2 displays Flex 53xx storage options.

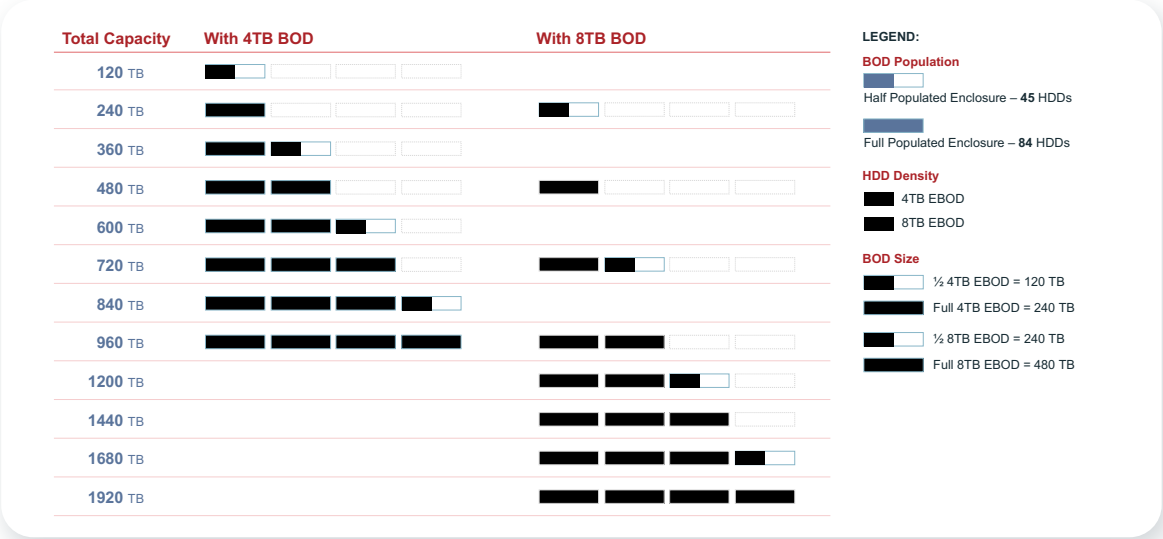

Figure 2. Flex 53xx storage capacity options

The correct container creation sequence can prevent multiple media MSDP containers from sharing the same LUN(s).

LUN creation uses the round-robin method. Let's say we want to provision six Primary containers and six Media MSDP containers on a single, fully populated storage shelf with 4 TB drives. There are six 40 TB LUNs. Six Primary containers with 5 TB storage each are created first. Using the round-robin method, these containers are created on a different LUN. Then we create six 34 TB media MSDP containers on six different LUNs as shown in the diagram below. The disk I/O is isolated by running NetBackup jobs concurrently on six media MSDP instances on six different LUNs.

```
[root@nbapp842 ~]#  df -h | grep msdpdata
/dev/vx/dsk/vxosdg/vWWXA_msdpdata-0                  34T  9.2T   25T  28% /var/lib/docker-veritas-plugin/vWWXA_msdpdata-0_vxosdg
/dev/vx/dsk/vxosdg/v1JTY_msdpdata-0                  34T   11T   24T  30% /var/lib/docker-veritas-plugin/v1JTY_msdpdata-0_vxosdg
/dev/vx/dsk/vxosdg/vFTpX_msdpdata-0                  34T   10T   24T  30% /var/lib/docker-veritas-plugin/vFTpX_msdpdata-0_vxosdg
/dev/vx/dsk/vxosdg/vSzVE_msdpdata-0                  34T   11T   24T  31% /var/lib/docker-veritas-plugin/vSzVE_msdpdata-0_vxosdg
/dev/vx/dsk/vxosdg/vMoeR_msdpdata-0                  34T  9.2T   25T  28% /var/lib/docker-veritas-plugin/vMoeR_msdpdata-0_vxosdg
/dev/vx/dsk/vxosdg/vA8E1_msdpdata-0                  34T   11T   24T  31% /var/lib/docker-veritas-plugin/vA8E1_msdpdata-0_vxosdg
[root@nbapp842 ~]#  df -h | grep catalog
/dev/vx/dsk/vxosdg/vXoMo_catalog                    5.0T  3.2G  5.0T   1% /var/lib/docker-veritas-plugin/vXoMo_catalog_vxosdg
/dev/vx/dsk/vxosdg/vg27z_catalog                    5.0T  3.2G  5.0T   1% /var/lib/docker-veritas-plugin/vg27z_catalog_vxosdg
/dev/vx/dsk/vxosdg/vFUOz_catalog                    5.0T  3.3G  5.0T   1% /var/lib/docker-veritas-plugin/vFUOz_catalog_vxosdg
/dev/vx/dsk/vxosdg/vVDj3_catalog                    5.0T  3.2G  5.0T   1% /var/lib/docker-veritas-plugin/vVDj3_catalog_vxosdg
/dev/vx/dsk/vxosdg/vgjQa_catalog                    5.0T  3.3G  5.0T   1% /var/lib/docker-veritas-plugin/vgjQa_catalog_vxosdg
/dev/vx/dsk/vxosdg/vzhnu_catalog                    5.0T  3.2G  5.0T   1% /var/lib/docker-veritas-plugin/vzhnu_catalog_vxosdg
```

```
[root@nbapp842 ~]#  vxprint -ht | egrep "sd" | egrep -v "^v|^pl"| egrep "catalog"
sd F000285A3EB979055B01000000-01 vFUOz_catalog-01 F000285A3EB979055B01000000 0 10737418240 0 vrts_0_4_data ENA
sd F00028E783CF79055B01000000-01 vVDj3_catalog-01 F00028E783CF79055B01000000 0 10737418240 0 vrts_0_5_data ENA
sd F000285A3E9379055B01000000-01 vXoMo_catalog-01 F000285A3E9379055B01000000 0 10737418240 0 vrts_0_2_data ENA
sd F000285A3EE179055B01000000-01 vgjQa_catalog-01 F000285A3EE179055B01000000 0 10737418240 0 vrts_0_6_data ENA
sd F00028E783A579055B01000000-01 vg27z_catalog-01 F00028E783A579055B01000000 0 10737418240 0 vrts_0_3_data ENA
sd F00028E783F279055B01000000-01 vzhnu_catalog-01 F00028E783F279055B01000000 0 10737418240 0 vrts_0_7_data ENA
[root@nbapp842 ~]#  vxprint -ht | egrep "sd" | egrep -v "^v|^pl"| egrep "msdp"
sd F00028E783F279055B01000000-02 vA8E1_msdpdata-0-01 F00028E783F279055B01000000 10737418240 73014444032 0 vrts_0_7_data ENA
sd F000285A3EB979055B01000000-02 vFTpX_msdpdata-0-01 F000285A3EB979055B01000000 10737418240 73014444032 0 vrts_0_4_data ENA
sd F000285A3EE179055B01000000-02 vMoeR_msdpdata-0-01 F000285A3EE179055B01000000 10737418240 73014444032 0 vrts_0_6_data ENA
sd F00028E783CF79055B01000000-02 vSzVE_msdpdata-0-01 F00028E783CF79055B01000000 10737418240 73014444032 0 vrts_0_5_data ENA
sd F000285A3E9379055B01000000-02 vWWXA_msdpdata-0-01 F000285A3E9379055B01000000 10737418240 73014444032 0 vrts_0_2_data ENA
sd F00028E783A579055B01000000-02 v1JTY_msdpdata-0-01 F00028E783A579055B01000000 10737418240 73014444032 0 vrts_0_3_data ENA
```

Best Practices for LUN Sharing

You cannot avoid LUN sharing when the instance's storage requirement is much smaller than the LUN size. Two containers sharing the same LUN is acceptable, although the I/O performance would be degraded when the two containers run I/O workloads concurrently. Staggering the workload schedules will help mitigate the I/O contention.

> The best practice is to have no more than two media MSDP containers share a LUN. Staggering the workload schedules will help mitigate the I/O contention.

With careful planning to minimize possible I/O contention, it is still possible to achieve good I/O performance even with some LUN sharing. Here are the best practices to follow if you cannot avoid LUN sharing:

1. Limit to no more than two instances sharing the same LUN, if possible.
2. Choose 4 TB or 8 TB shelves based on the MSDP storage pool size profile. Choose 4 TB shelves if the storage pool of multiple instances is 20 TB or less. This option can reduce the need for more than two instances sharing the same LUN.
3. Create the instance with the highest I/O performance requirement first. The storage pool of the first created instance will occupy the outer layer of the LUN, which can outperform the storage pool located on the inner layer of the LUN.
4. Use the backup schedule and Storage Lifecycle Policy (SLP) to reduce I/O contention and achieve the best I/O performance.
    - Multiple instances sharing the same LUN will not affect performance unless the instances sharing the LUN are active at the same time. If the workload of small instances can be finished in a few hours, then stagger the activation of each instance to avoid job overlap and thus the I/O contention.
5. Instances with a high dedupe ratio, such as 80 percent or above, do not generate a lot of write I/Os. If multiple instances (two or more) must share the same LUN and you cannot stagger the backup workload, then choose instances with high dedupe ratios to share the LUN. You should stagger the SLP jobs that generate read I/Os for each instance, especially when more than two instances share the same LUN.

6. In some cases, the leftover space (we will call it a subdisk) from any LUN on the storage shelf may not be enough to meet an instance's storage pool requirement. An alternative is to concatenate two or more subdisks, each from a different LUN, to form the required volume size. In this case, the order in which you choose the subdisks can impact performance. Carefully choosing the subdisks can avoid unnecessary I/O contention. The following example demonstrates how to choose subdisks to avoid I/O contention:

   Assume LUN 1 has two subdisks—SD1 and SD2, and LUN 2 has two subdisks—SD3 and SD4. The storage pool of instance A requires SD1 and SD3, and instance B ends up with SD2 and SD4. By default, the subdisks are concatenated together to form a larger volume and write I/O will fill up the first subdisk before going to the second subdisk. So if instance A configures SD1 as the first subdisk, then instance B should configure SD4 as the first subdisk. Doing so will ensure that instances A and B will not write to the same LUN some of the time, thus avoiding I/O contention.

   - Note: Sometimes you discover LUN sharing after the media/MSDP instances are deployed and backup data already resides on the subdisks. To reduce I/O contention in this case, you may need to change the storage allocation by moving subdisks around. You will need to contact Veritas Tech Support for help.

Multiple LUNs for One Media MSDP Container

If you provision fewer than six media MSDP containers, you can allocate more than one LUN for some containers.

The storage shelves of the Flex 53xx Appliance can be populated with either 4 TB drives or 8 TB drives. The LUN size of 4 TB storage shelves is 40 TB, and the LUN size for 8 TB storage shelves is 80 TB. If the volume size is more than 40 TB with 4 TB drive storage shelves, two LUNs will be concatenated to form the desired volume size.

For Flex Appliances, the default maximum volume size is 80 TB. More than one VxVM will be created if the storage pool size specified for an instance is greater than 80TB.

This default size will affect how many VxVM volumes/VxFS filesystems arMore than one VxVM will be created if the storage pool size specified for an nstance is greater than 80TB.TB. For example, to create a storage pool of 120 TB, two VxVM volumes will be created. How the two volumes are provisioned, however, will depend on the drive size of the storage shelf.

1. For a storage shelf populated with 4 TB drives, two LUNs will be concatenated to form the first 80 TB VxVM volume and the second VxVM volume would be 40 TB sitting on a single LUN. We introduced the Best Fit storage algorithm in Flex 2.0.1 that would create two file systems for 80 TB ask in a Flex 53xx 4 TB disk drive storage shelf as opposed to one file system as previously. The algorithm also takes care of choosing an appropriate LUN based on the storage size you specify.

2. For a storage shelf populated with 8 TB drives, the first 80 TB VxVM volume will be created on a single LUN and the remaining 40 TB VxVM volume will be provisioned out of another LUN.

The above two storage allocations are the default behavior, and the I/O performance should be comparable.

Tuning MSDP-Direct Cloud Tier

Beginning with version 8.3, NetBackup added a significant enhancement to the cloud tier offering, MSDP-C. The new enhancement simplifies cloud tier management and improves cloud tier performance. You no longer need to create a separate CloudCatalyst instance as the gateway for uploading data to cloud storage. This enhancement enables a single media server instance to configure multiple logical storage units (LSUs), including one local LSU and zero or more cloud LSUs. The LSU can be regular or WORM storage. With NetBackup 8.3.0.1 and 9.0, WORM is only supported in local LSUs. Support for WORM storage in cloud LSUs with AWS Object Lock is available starting with NetBackup 9.1, and support for WORM storage in cloud LSUs with Azure Object Lock is available starting with NetBackup 10.0.

Several tuning parameters have been introduced to let you customize the MSDP to meet the performance requirement of different workloads. These parameters are configured in the cloud.json file available at: <Storage>/etc/puredisk/cloud.json, where <Storage> should be similar to the following path directory: /var/lib/docker-veritas-plugin/vemLY_msdpdata-0_vxosdg

One of the most important tuning parameters that can significantly impact uploading performance is UseMemForUpload. If the parameter is set to true, the upload cache directory is mounted in memory as tmpfs. The parameter is set to true by default when adding a cloud LSU. If there is enough memory available, then uploading to the cloud will go through memory; otherwise, cloud uploading will use disk cache, which is significantly slower. The rest of this section will cover how to tune the new parameters to ensure memory uploading occurs.

Beginning in NetBackup 10.1, to support a large MSDP storage pool, a new fingerprint data lookup scheme is introduced to reduce memory requirements. In addition, there are some significant changes to the cloud tier memory configuration. The guideline in the remainder of this section is applicable to Flex releases that are configured with NetBackup releases prior to NetBackup 10.1. For the new memory cloud tier tuning NetBackup 10.1 and beyond, please follow the guideline in the section Tuning MSDP-Direct Cloud Tier for NetBackup 10.1 and beyond.

For NetBackup releases prior to 10.1, the following are memory-related tuning parameters. You can change the defaults, if needed.

1. UsableMemoryLimit
2. MaxCacheSize
3. UploadCacheGB
4. MaxCloudCacheSize

UsableMemoryLimit: This parameter specifies the maximum amount of memory usable for the fingerprint (FP) cache and cloud upload cache. The parameter is expressed as the percentage of physical RAM on the appliance. The default is 80 percent, which ensures at least 20 percent of the physical RAM is reserved for other processes and instances running on the system, such as NetBackup and kernel processes. The following relationship must be held to ensure memory is used for cloud uploads:

MaxCacheSize + MaxCloudCacheSize + UploadCacheGB/ (system RAM size) <= UsableMemoryLimit

By default, the MaxCacheSize and MaxCloudCacheSize in a media server instance are set at 50 percent and 20 percent respectively, which leaves 10 percent of RAM for tpmfs. The default setting should work well if there is only one local and one remote cloud storage configured on the Flex Appliance.

There may be one or more MSDP instances on a Flex Appliance, and each instance may have one local and zero or more cloud LSUs. Each local LSU has its own MaxCacheSize, and each remote LSU has its own MaxCloudCacheSize and UploadCacheGB (also called Cloud in Memory upload cache size). All the media server instances configured on the appliance need to share the 80 percent UsableMemoryLimit. To ensure memory is used for cloud upload, the above formula must be held for each individual instance. The sum of all MaxCacheSize, MaxCloudCacheSize and UploadCacheGB configured in all MSDP instances should not exceed the 80 percent of the system RAM to ensure memory upload and avoid potential memory starvation.

MaxCacheSize: The recommended MaxCacheSize for a media server instance changed to 50 percent in Flex 2.0.1 (down from 60 percent). This change ensures that the Flex Appliance performance with a single MSDP instance with or without MSDP-C is on par with a NetBackup Appliance. MaxCacheSize tuning as described in the section, Media MSDP Instance Memory Tuning is still applicable and must be followed with or without the remote LSU configuration.

UploadCacheGB: This parameter is set in the file <STORAGE>/etc/puredisk/cloud.json, and the default is 12 GB. The default value for this parameter is set in the contentrouter.cfg parameter, CloudUploadCacheSize. The location of this upload cache can be in tmpfs or on disk. If the parameter UseMemForUpload is false, the writes will go to the local disk. If the parameter is true, a tmpfs will be created in memory and tmpfs will be used for cloud upload if there is enough memory available for tmpfs. For each cloud LSU, this parameter should be set to larger than the following:

(Max concurrent upload stream number) * MaxFileSizeMB *2

The MaxFileSizeMB is set in the cloud.json file and the default is 64 MB.

MaxCloudCacheSize: This parameter is used to configure the amount of cache that can be used for temporarily caching the FPs needed for uploading to cloud storage. The purpose of the cache is similar to the FP cache configured for traditional client-side deduplication. It is used for cloud disk pool fingerprint lookup cache. The majority of the cloud fingerprint lookup cache is used to cache the FPs of the last backup. Cache entries are not permanent, and the content will be replaced by the next set of jobs once the previous jobs are completed. The size is specified as the percentage of the total physical RAM on the system and the default value is 20 percent. All cloud LSUs need to share this 20 percent of RAM.

Considerations when More than One Media Instance has at least One Cloud LSU Configured

As mentioned earlier, the following formula must be held for each individual media/MSDP instance: MaxCacheSize + MaxCloudCacheSize + (UploadCacheGB/system RAM size) <= UsableMemoryLimit

Most importantly, the 80 percent UsableMemoryLimit needs to be shared among all the media server instances configured on the appliance. Similarly, the 20 percent MaxCloudCacheSize needs to be shared by all cloud LSUs configured on the appliance.

If the upload image sizes among the cloud instances are different, you can fine-tune the MaxCloudCacheSize setting following these steps:

1. Maximum concurrent upload image size in KB = (Average stream size in KB * maximum concurrent upload streams)
2. Number of total fingerprints = (result from step 1 above)/128KB
   Where 128 KB is the default data segment size (DefaultSegmentSize) is defined in contentrouter.cfg
3. The size of MaxCloudCacheSize in Bytes = (result from step 2 above * 48(Bytes)
   Where 48 Bytes is the size of RAM required to cache a FP in memory
4. The MaxCloudCacheSize = (convert the result from step 3 above to GB)/Total RAM in GB * 100

If the number of concurrent upload streams is different between the cloud LSU instances, you can use the following formula to calculate the UploadCacheGB for each instance:

(Max concurrent upload stream number) * MaxFileSizeMB *2

If the aggregate MaxCacheSize is lower than 50 percent of the RAM size, then the aggregate MaxCloudCacheSize can be larger than 20 percent, and the aggregate UploadCacheGB can be greater than 12 GB as long as the 80 percent of aggregate UsableMemoryLimit is maintained.

For example, assume the RAM size of a Flex 53x0 Appliance is 1.5 TB and the total storage capacity is 960 TB. Then according to the best practice guide of allocating MaxCacheSize, we would allocate, in aggregate, approximately 480 GB (or 32 percent) of the RAM

for the MaxCacheSize. In other words, MaxCloudCacheSize and UploadCacheGB get an additional 18 percent of RAM allocated for improving performance, if necessary, without violating the 80 percent UsableMemoryLimit.

Tuning MSDP-Direct Cloud Tier for NetBackup 10.1 and Beyond

Beginning in NetBackup 10.1, a new fingerprint cache lookup data scheme is introduced. The new scheme splits the current memory lookup into two components, Predictive cache (P-cache) and Sampling cache (S-cache). The P-cache is used to cache the fingerprints that are most likely used in the immediate future, while the S-cache caches a percentage of the fingerprint from each backup and a subset of each sample fingerprint is inserted into the S-cache. P-cache is first used to find duplicates, and lookup misses reaching a threshold are searched in S-cache for possible matches; if found, the predicted relevant fingerprints are loaded from disk into the P-cache for deduplication.

With NetBackup 10.1, the P- and S-cache is the default FP lookup scheme for the cloud logical storage unit (LSU). The local LSU volume still uses the MaxCacheSize. Beginning in NetBackup 10.2, local and remote LSUs are defaulted to using P- and S-cache for new installations for all MSDP platforms, except MSDP BYO. The system upgraded to 10.2 keeps the existing configuration.

Configuration	Default Value
MaxCacheSize	512MB
Max P-cache size	40% in NetBackup 10.1 and 10% in NetBackup 10.1.1
Max S-cache size	20%
EnableLocalPredictivesSamplingCache in spa.cfg	True
EnableLocalPredictiveSamplingCache in contentrouter.cfg	True
MaxCloudCacheSize	Deprecated and replaced with Max P-cache size and Max S-cache size

Table 3. Configuration change and default value for P- and S-cache

With the above change, to ensure memory is used for upload, the formula prior to 10.1 is changed to:

MaxCacheSize + MaxPredictiveCacheSize + MaxSamplingCacheSize + Cloud in-memory upload cache size <= UsableMemoryLimit

The above P- and S-cache setting is the default for all MSDP-supported platforms, except BYO for new installations. The existing configuration is preserved for upgraded systems. For Flex Appliances configured with multiple instances of Media, Storage server, and local and cloud LSUs, the fingerprint cache setting needs to be set separately for each LSU.

With P- and S-cache in 10.2, local and all cloud LSUs share the same P- and S-cache, and the previous MaxCacheSize can be ignored. P- and S-cache setting needs to be done carefully, setting them too high will waste memory, while setting them too low will lead to a poor dedup ratio and impact backup performance.

In general, S-cache size should be proportional to the backend storage size, while P-cache size is determined by the maximum number of concurrent jobs. Use the following rule of thumb for P- and S-cache tuning:

1. For each 10 TB of backend storage, allocate 1 GB of RAM for S-cache
2. For each backup stream, allocate 250 MB of RAM for P-cache. So, the total P-cache allocated should be 250MB * maximum # of concurrent jobs

To ensure enough memory for other processes running on the system, P- and S-cache size together should not exceed the MaxUsableMemory. Other processes that also need memory include:

a. Basic OS/NetBackup if running as media server
b. NetBackup processes if NetBackup runs in the same node
c. Spad cache for opt-dup source
d. Mtstrd cache for backup source
e. Spooler cache

Disk cache for cloud upload and download

The NetBackup cloud-tier allows each media server to create one or more cloud LSUs. It is important to know that for each cloud LSU created, roughly 1 TB of MSDP storage pool is reserved for the LSU to be used as cloud disk cache. Starting with version 10.2, this preserved disk cache can be configured from the WebUI during LSU creation. The disk cache size for upload is 12 GB and is set by the parameter UploadCacheDB, while the default disk cache size for cloud download is 1 TB, which is set by the parameter DownloadDataCacheGB and DownloadMetaCacheGB. The default values for parameters are set in contentrouter.cfg with CloudUploadCacheSize, CloudDataCacheSize, and CloudMetaCacheSize respectively. As mentioned earlier, the disk caches occupy space in the MSDP pool. For an MSDP pool with limited storage size, the reserved disk cache can consume too much space, resulting in little usable space for regular backup jobs. If jobs are failing with error code 129 and 84, it may indicate that there is no space left on the device, even though the MSDP pool may still have plenty of space according to df –h and dsstat. For this kind of case, we recommend:

1. Limit the number of cloud LSUs created per media instance, especially if the storage pool is relatively small
2. Reduce the default CloudDataCacheSize and CloudMetaCacheSize

If there is enough memory for upload to go through memory cache, the UploadCacheGB can be set to maximum number of concurrent streams * MaxFileSizeMB * 2 and is set in the cloud.json file. If the maximum number of concurrent streams is 100, the UploadCacheGB can be set to 12 GB. The DownloadDataCacheGB and DownloadMetaCacheGB used for restore/opt-dup download cache can be as small as a few GB to function. A larger download disk cache size can improve restore/opt-dup performance because it can help avoid downloading the same data object more than once. Tuning the DownloadDataCacheGB and DownloadMetaCacheGB requires knowing the maximum number of concurrent download streams. In most cases, restoring from the cloud requires downloading the entire data container (64 MB). This is because the container created at backup time usually consists of data from a single client and MSDP-C will download the entire container so that the same container is only fetched once during a restore.

The default values of the parameters are set under <storage> /etc/puredisk/contentrouter.cfg and the default values are used for all future LSU. The parameters in cloud.json are used to set values used for each LSU already created . The file is at <storage>/etc/puredisk/cloud.json.

Appendix

Special note for Flex 3.0: If you rely on the iostat command to monitor Flex Appliance IO performance on the host, after upgrading to Flex 3.0 you may not be able to find VxVM volumes in the iostat output. This is because to bypass a known bug in RHEL8.6, VxVM changed the IO mode from request to BIO. Due to the high overhead of collection iostat with BIO mode, the iostat is disabled; instead we recommend using the InfoScale command vxstat for IO monitoring.

Flex Tuning Parameters

Default Tunings in Flex 2.0 and Flex 2.0.1

Parameter	Default Value	OS/VxFS/NetBackup
read_nstream	8	VxFS
write_pref_io	209152	VxFS
max_diskq	8388608	VxFS
dalloc_enable	0	VxFS

Table 4. VxFS tuning parameters added to improve I/O performance in Flex 2.0.x

The above four tunings should improve I/O subsystem read/write performance. For more information about the effect of the parameters, check the man page of the command `vxtunefs`.

To change the parameters dynamically:

```
vxtunefs -o read_nstream=8 <mount_point> vxtunefs –
o write_pref_io=2097152 <mount_point> vxtunefs -o
max_diskq=8388608 <mount_point> vxtunefs -o
dalloc_enable=0 <mount_point>
```

To make the changes persistent, modify `/etc/vx/tunefstab` and add the following line

```
>>>

system_default read_nstream=8
system_default write_pref_io=2097152
system_default max_diskq=8388608
system_default dalloc_enable=0
```

Manual Tunings Needed in Flex 1.2

This section lists the kernel, Veritas File System, and NetBackup tunings that are needed in the Flex 1.2 and 1.3 releases.

1. numa_balancing

Turn off numa_balancing on the Flex Appliance to avoid unnecessary cost in memory migration and memory swapping. You can do so in the Flex host as shown below:

```
# vi /etc/sysctl.conf
kernel.numa_balancing = 0
# sysctl -p /etc/sysctl.conf
```

2. overcommit_ratio

Increase the overcommit_ratio from default 90 to 100 to handle higher bursts in virtual memory usage. This can be changed in the Flex host as shown below:

```
# vi /etc/sysctl.conf
vm.overcommit_ratio = 100
# sysctl -p /etc/sysctl.conf
```

Manual Tunings Needed in Flex 2.0

1. Apply hotfix

Flex VxFS patch and MSDP EEB bundle for Flex2.0 running NetBackup 8.2 or NetBackup 8.3.0.1 container. The hotfix is needed to improve mixed (read/write) workload performance. For more details on applying the hotfix, see the technote:

2. Net.core.somaxconn

Starting from NetBackup 9.0.1, the net.core.somaxconn changed from 128 to 1024 by default.

This kernel parameter determines the maximum number of backlogged connections allowed for each TCP port. The default value is 128, but we recommend changing it to 1024 to avoid a connection refused error between clients and the appliance.

Prior to NetBackup 9.0.1, you can apply the tuning manually, as shown below. To tune the value for the NetBackup instance on the Flex Appliance host:

```
# cd /mnt/data/infra/profiles/instances
# vi izTAR_8-2.json
[...]
“sysctls”: [
“kernel.sem=400 307200 32 1024”,
“net.ipv4.tcp_keepalive_intvl=10”,
“net.ipv4.tcp_keepalive_probes=5”,
“net.ipv4.tcp_keepalive_time=900”,
 “net.core.somaxconn=1024”
],
[...]
```

Restart the NetBackup instance and verify the tuning change in the NetBackup instance:

```
# cat /proc/sys/net/core/somaxconn
1024
```

3. AllocationUnitSize

The default for this MSDP parameter changed to 8 MiB with a Media MSDP container running NetBackup 8.3.0.1; the default was 2 MiB. If you are running a NetBackup container prior to version 8.3.0.1, you need to set this parameter to 8 MiB manually, especially if the appliance has 1.5 TB of RAM installed.

Change the AllocationUnitSize in the MSDP container:

```
# ssh appadmin@<msdp_instance_name>
$ sudo /usr/openv/pdde/pdag/bin/pdcfg --write /mnt/msdp/vol0/etc/puredisk/contentrouter.cfg --section
CACHE --option AllocationUnitSize --value 8MiB
```

Check the changed AllocationUnitSize as shown below:

```
$ sudo /usr/openv/pdde/pdag/bin/pdcfg --read /mnt/msdp/vol0/etc/puredisk/contentrouter.cfg --section
CACHE --option AllocationUnitSize
```

Restart the pdde-storage process with the following commands:

```
$ sudo /etc/init.d/pdde-storage force-stop
$ sudo /etc/init.d/pdde-storage start
```

4. vm.extfrag_threshold

This kernel parameter impacts whether the kernel does memory compaction or memory reclaim to satisfy a high-order allocation; the default is 500. The default should work well for most installations. If the appliance performance is suffering frequently due to memory fragmentation, try increasing this value to 1,000 to free memory through reclaim.

You can do so in the Flex host as shown below:

```
# vi /etc/sysctl.conf
vm.extfrag_threshold = 1000
# sysctl -p /etc/sysctl.conf
```

5. read_nstream

This vxfs parameter impacts the file system read ahead performance. Increasing this value can improve rehydration performance. The default value was one for Flex releases before 2.0.1. The new recommendation is to set this parameter to eight to improve read performance.

To change it dynamically:

```
# vxtunefs -o read_nstream=8 <mountpoint_of_msdp_data_volume>
```

Versions

Parameter	Date	Author	Key Updates
1.0	Nov 2021	Su-jin Chan, Angela Ellingsen, Rachel Zhu	Original document
2.0	Mar 2023	Rachel Zhu, Su-jin Chan, WeiBao Wu	Removed Flex 1.2 and 1.3 content, updated MSDP-Configuration, added Flex 3.0 content.

VERITAS

2625 Augustine Drive
Santa Clara, CA 95054
+1 (866) 837 4827
veritas.com

For global contact information visit:
veritas.com/company/contact

V1452 06/23