VMAX ALL FLASH WITH THE ADAPTIVE COMPRESSION ENGINE

ABSTRACT

VMAX All Flash and the Adaptive Compression Engine offer more than simple capacity savings. Intelligent algorithms paired with hardware acceleration deliver enhanced performance and reduced compression overhead. This results in great overall system performance when inline compression is enabled and in use.

May, 2017

WHITE PAPER

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The VMAX All Flash's Adaptive Compression Engine (ACE) offers multiple benefits such as delivering expected performance and capacity savings. Space savings is commonly the first thought when compression is discussed. However, in all cases there is some cost, usually in performance, due to the overhead that comes with the actual function of compressing data. The Dell EMC Adaptive Compression Engine's design using intelligent algorithms, paired with hardware acceleration, minimizes this cost. This combination allows the system to maintain a balanced and optimal configuration. The result is a system that can deliver efficient capacity savings and deliver optimal performance.

AUDIENCE

This white paper is intended for:

- Customers who manage IT, storage architects and administrators involved in the everyday operations and decisions related to data storage management.
- Dell EMC staff and partners to assist in content development and understanding of the Adaptive Compression Engine.

COMPRESSION DESIGN OVERVIEW

The Adaptive Compression Engine (ACE) is available for VMAX All Flash storage arrays running the HYPERMAX OS 5977 Q3 2016 Service Release and later. All systems must be sized considering a compression ratio ranging between 1.3:1 and 3.0:1. When sizing a system where compression is included it is important to understand what data populates the system. It is equally important to understand how to determine the amount of capacity needed. There are multiple different definitions of capacity in reference to storage. One of the most important definitions is Host Addressable Capacity. This is how much capacity can be presented to the hosts and applied to applications that will deliver the workload to the system.

There are three key design factors which make the Dell EMC Adaptive Compression Engine unique in the industry. These design factors are:

1. Intelligent algorithms
2. Inline hardware compression
3. Dynamic backend storage

The Adaptive Compression Engine does more than save on capacity, its design delivers the performance expected from an all flash storage at the same time. A familiar and proven hardware component is used to compress the data with a secondary software option to continue compressing data in the event there is a fault with the hardware. Intelligent algorithms determine the best compression ratios to use as well as dynamically create a customized backend that caters to the incoming workload. The typical static backend becomes obsolete with the Adaptive Compression Engine. The ability to change the storage layout online as needed ensures that the system operates at optimal levels for both performance and space efficiency, making the backend truly dynamic. Using internal statistics, additional algorithms find the busiest data in the system. Doing this allows busy data to skip the compression process which means that data does not experience any compression overhead. This allows the system to efficiently focus on data that should be compressed and reduces the compression load on both the system and the compression I/O module. All this complexity is done within the system.

Managing compression is quite simple and can be enabled or disabled at the application level. The option is available when creating storage groups and is enabled by default. This can be changed when creating the storage group or by modifying the storage group at a later time. All incoming I/O is sent through the compression flow for storage groups that have compression enabled. Modifying the storage group compression setting does not start a compression operation that can be monitored. When the setting changed is to enable compression, the data is compressed over time by background functions in the system. When the setting is changed to disabled the data is uncompressed over time as long as there is adequate free capacity.

TERMINOLOGY

- **System compression** – The compression ratio that is relative to the system configuration. This value is set within the system and is the expected average of the compressed data in a given system.
- **Storage group compression ratio** – The compression ratio displayed for allocations related to a specific storage group. This value may be greater than the system compression ratio.
- **Space efficiency** – The range of values that describe the capacity space savings that a user may experience in regards to compression or other data services that offer capacity savings.
  - **VP Savings** – The savings achieved by not allocating unused space.
  - **Local Replication Savings** – The space savings achieved when using shared allocations to avoid allocating duplicate data.
  - **Compression Savings** – The amount of capacity savings achieved after compression has reduced the actual data size vs the amount of space taken without compression.
- **Compressibility** – The optimal compression ratio a device can achieve.
  - **Compressibility stamp** - A possible compression ratio assigned to a track as a hint. This may not be the actual ratio the data is compressed. This stamp is relative to its compressibility.
- **Compression candidate** – Data that has come into the system where the compression state of the related storage group is enabled and data is not marked as active data.
- **Compression ready** – The state of the system when the default Storage Resource Pool (SRP) is capable of storing compressed data. For a system to be able to compress data it must have one compression I/O module per director, have compression enabled and have a system compression ratio set.
- **Compression capable** – A system installed with at least the Q3 2016 HYPERMAX OS where the sized compression ratio applied to the system IMPL is 1.0:1.
- **Compression ratio** – A system setting describing the maximum backend potential capacity for the default SRP as an effect of compression.
- **Compression pool** – A collection of data devices configured within the drives where the track size is the same.
- **Petabytes usable (PBu)** – The usable storage capacity in the absence of compression referring to the amount of physical storage in the system.

  Example: 50 PBu is 50 Petabytes of usable physical storage.
- **Petabytes effective (PBe)** – The effective storage capacity in the presence of compression.

  Example: 50 PBu of physical storage that is compressible at a 2:1 ratio translates to 100 PBe capacity.

CONFIGURATION AND SIZING

SYSTEM CONFIGURATION

In previous VMAX3 systems, the physical storage media was preconfigured and static. All disk groups were comprised of physical disks with 16 hypers for each drive. The size of each device was determined by the maximum size of the drive with all the track sizes set as 128K. This was referred to as a 1/16th backend layout, allowing the system to have a one-to-one write-to-allocation ratio. Implementing compression for the systems changes this configuration. There is now a 1/64th backend layout. This means that each physical drive has 64 hypers for each physical drive where the track size is not a constant 128K. This is now considered a dynamic configuration. While data is allocated to the drives in the system, the backend configuration is dynamically modified to cater to the incoming workload. Each system starts with an initial configuration per drive consisting of hypers with 16K, 32K, 40K, 48K, 56K, and 72K track sizes for each, two hypers with a 64K track size, the remaining 56 hypers have a track size of 128K. This layout represents the compression pools where compressed data is allocated. The initial workload sent to the system starts to alter the backend configuration changing the count and variation of hypers to expand the pools when needed. This is performed by converting existing available space to the track size needed. The system determines which pool to convert space from using the compression pool with the largest track size where there is the most free space. When the system workload has reached a stable state the dynamic changes to the backend slow down.

SYSTEM SIZING

System sizing and configuration play a key role in how the compression engine functions. The sizing calculation is built from the standard sizing exercise used for all VMAX All Flash storage platforms. 1 terabyte of cache supports 235 terabytes of physical disk space representing the amount of capacity uncompressed data can be written to. This does not include the effective compression ratio or an overprovisioning (OP) factor. With compression applied to the calculation those numbers change: 1 terabyte of cache supports 117.5 terabytes of disk space, however with a compression ratio of 2.0:1 applied the 117.5 terabytes of disk space will accommodate 235 terabytes of written data. Adding the default OP factor of 1.3 equates to 152.25TB of host addressable capacity. The relationship between cache and effective capacity is vital to the configuration of every volume or object created within the system.

Information regarding the volume or object is captured as metadata. Metadata is a portion of cache used to build identifying tables of information about every object within the system. This information is critical to maintain and be available for recovery in the event of a disaster.

ADAPTIVE COMPRESSION ENGINE (ACE)

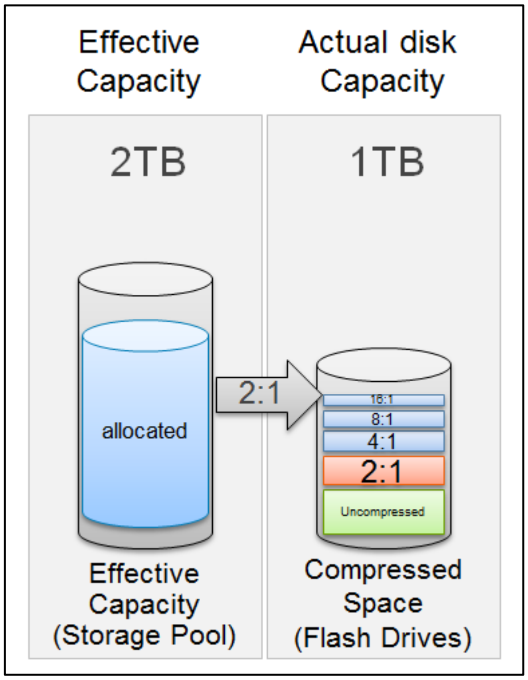
The Adaptive Compression Engine (ACE) is the combination of core compression functions and components. The core components are:

- **Hardware Acceleration** - Each system has multiple hardware compression modules that handle the actual compressing and decompressing of data. The system requirements state that each system will has compression module per director which equates to 2 modules per engine. The compression modules being used are tested and proven components as they are the same modules that have been in use for multiple years supporting remote replication compression (SRDF Compression). The use of the modules reduces compression overhead from the system resources. As a secondary function, there is built in software compression that is automatically applied in the event of a fault or failure with one or more of the compression modules.

- **Optimized Data Placement** – Optimized Data Placement is a function within the VMAX All Flash system that is always running and is responsible for dynamically changing the compression pools as needed. This function generates minimal overhead similar to the overhead experienced by FAST in a VMAX3 array. Optimized Data Placement dynamically alters the backend by creating a variation of compression pools that cater to the incoming data. Compression pools represent actual disk space on multiple solid state drives (SSD). Once compressed, data is allocated to these pools. There are multiple possible compression pools that may be created in order to build an optimal backend. The result is a suitable layout of compression pools that accommodate the data sent to the system. All data can be compressed however it does not all compress to the same degree. Some data may compress to one size and other data may compress to another size. In order to maximize compression efficiency, multiple compression ratios need to be available. The complete list of available track sizes for the data devices that will make up the compression pools are; 8K, 16K, 24K, 32K, 40K, 48K, 56K, 64K, 72K, 80K, 88K, 96K, 104K, 112K, and 128K. Each compression-enabled system may have a different variation of the compression pools where compressed data is allocated.

  Figure 1 to the right is a representation of the compression pool configuration illustrating multiple pools that accommodate various compression ratios.

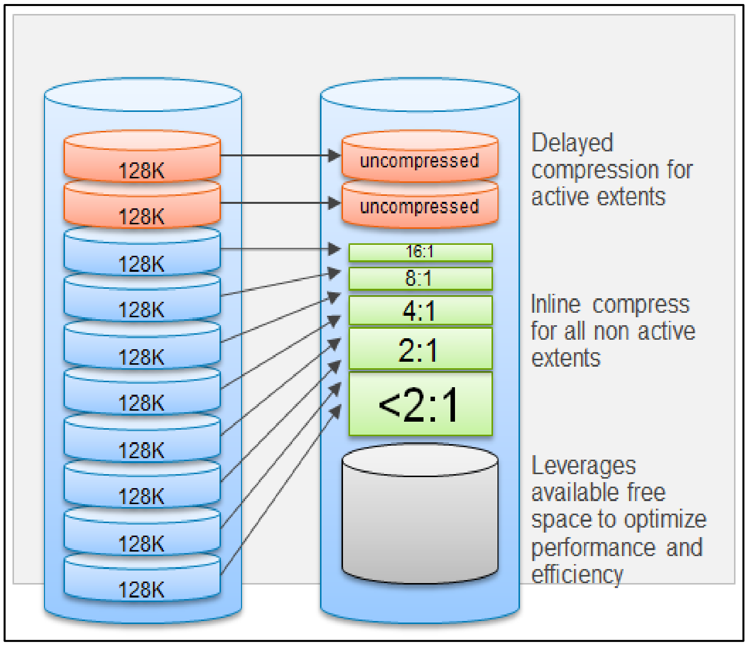
Figure 1: Optimized data placement example.



- **Activity Based Compression (ABC)** – ABC aims to prevent constant compression and decompression of data that is active or frequently accessed. The ABC function marks the busiest data in the SRP to skip the compression flow regardless of the related storage group compression setting. This function differentiates busy data from idle or less busy data and only accounts for up to 20% of the allocations in the SRP. Marking up to 20% of the busiest allocations to skip the compression action is a benefit to the system as well as the end users. This ensures optimal response time and reduced overhead that can result from constantly decompressing frequently accessed data. The mechanism used to determine the busiest data does not add additional CPU load on the system. This function is similar to the FAST code used for promoting data in previous code releases. ABC leverages the FAST statistics to determine what data sets are the best candidates for compression. It allows the system to maintain balance across the resources providing an optimal environment for both the best possible compression savings and the best performance. Effectively this avoids compress and decompress latency for the busiest data. In addition this reduces the system overhead of compression allowing the focus to be on the best candidate data to be compressed.

  Figure 2 shows active but compressible data as uncompressed. The remaining compression pools provide space for compressed data to be allocated.

Figure 2: Activity Based Compression example



- **Fine Grain Data Packing** – In VMAX All Flash using the Adaptive Compression Engine each 128K I/O is split into four 32K buffers. Each buffer is compressed individually in parallel maximizing the efficiency of the compression IO module. The total of the four buffers result in the final compressed size and determines where the data is allocated. Fine Grain Data Packing offers benefits with performance for both the compression function as well as the overall performance of the system. Included in this process is a zero reclaim function that prevents the allocation of buffers with all zeros or no actual data. Pairing the zero non-allocation function with the Fine Grain Data Packing allows the compression function to operate in a very efficient manner with minimal cost to performance. Compressing the 128K I/O in four buffers individually in parallel allows for each section to be handled independent even though they are still part of the initial 128K I/O. The main benefit comes in the case of partial write updates or read I/O. In the event that only one or two of the sections need to be updated or read only that data is uncompressed.

Figure 3: Fine Grain Data Packing example.

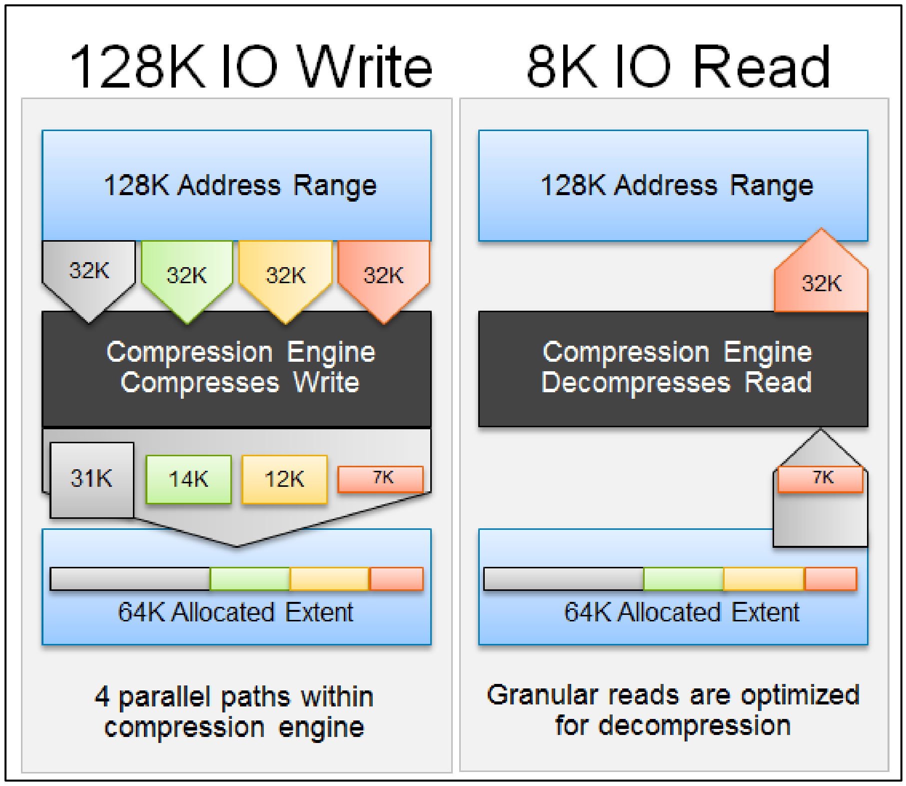


Figure 3 represents a 128K write I/O divided into four cache buffers. Each buffer starts as 32K (32*4=128) and is compressed individually. The sum of the four sections creates a 64K compressed track. The achieved savings with this example is 2:1 as the 128K I/O is compressed and allocated as a 64K track.

These core components combined with hardware acceleration are what make Dell EMCs compression adaptive.

INSTALLATIONS

There are three installation scenarios to consider. A system sized, built and installed with compression is compression ready. A system installed with compression not included is compression capable. All situations are valid and supported, however there are a few differences.

- **Compression Ready** - A compression-ready system includes a system compression ratio input to the IMPL and must be installed at 5977.945.890 or later. In addition, there is a default compression pool configuration with 8 compression pools to accommodate the allocation of compressed data. The default compression pools are: 16K, 32K, 40K, 48K, 56K, 64K, 72K and 128K. On a compression-ready system, the user can enable compression at the storage group level. The compression status is enabled by default when creating storage groups. It is also supported with cascaded storage groups and can be changed amongst the child groups; however, it is enabled by default. When provisioning with Solutions Enabler to create a storage group, compression is not enabled unless either a service level or an SRP are assigned.

- **Compression Enabled** - A compression-enabled system is installed with 5977.945.890 or later. A system in this state is very similar to a compression-ready system. The compression pool layout starts with the default as explained in the compression-ready description. The difference being a compression-enabled system has one or more storage groups with the compression option enabled. When the applications start sending compressible data to the system, it starts to consume the available space in the compression pools. In addition, as compressible data populates the system, the compression pool layout may change.

- **Compression Capable** - A compression-capable system is installed with 5977.945.890 or later. This means the system was installed without a system-level compression ratio applied to the IMPL. Compression-capable installations have only one pool to allocate written data. When a system is compression-capable, the options to enable compression at the storage group level are not available. All incoming data is not compressed and is allocated to the single pool. However, compression-capable systems still gather active and idle statistics regarding the data allocated. This is important should the system be changed to compression-ready. Changing the state of a compression-capable system to compression-ready is not as simple as just entering a compression ratio into the IMPL. It is critical that the system resources are reviewed and validated in order to confirm that compression can be supported. If the outcome of the review is favorable, a compression ratio can be added to the IMPL. In other cases it may be necessary to add components to the system, such as additional drive capacity, cache or both, in order to support compression.

CAPACITY USAGE

ACE is an inline compression feature intended to offer long-term space savings. The compression engine is intelligent enough to use the available system resources efficiently. Compressibility of the incoming workload is not known until the data is actually compressed. Because the compression engine is using statistics about the written data to determine what is active and what is idle, there must be some historical information to determine active or idle status. Therefore, the activity-based compression function does not apply to net new writes; they are compressed and allocated to a compression pool. Continued access to data allocated to the compression pools generates statistics. These statistics are used by the activity-based compression function to differentiate active data from idle data.

There are three basic phases of capacity usage that are important to recognize when running workloads on a system using ACE. Each of the phases can deliver a different level of achieved compression. The initial load to the system achieves a displayed compression ratio that changes over time. Throughout the three phases, the achieved compression ratio fluctuates while moving towards system compression. The three phases as related to used capacity are the initial load phase, climbing phase and optimal running phase.

- **Initial Load Phase** - Identified when capacity usage is 0% - 40%. During this time it is possible that the achieved compression ratio varies from system compression. In most cases the capacity usage growth is due to new data being written to the system. As new allocations compress, this is likely to change. Over time, if that data is accessed, it may become active data. In this case, it could be moved to the uncompressed pool (identified as the 128K pool). This alters both achieved system compression and storage group compression. This is a learning stage for the system as workload profiles and I/O patterns are identified. In addition, active and idle statistics are collected. The dynamic design of ACE uses the information gathered to change the compression tier layout online to one that best accommodates the workload.

- **Climbing Phase** - During the time the capacity usage is between 41% - 70% the system is in the climbing phase. The changing of the compression tier layout may be less active. Activity based compression may apply to more of the written data in the system. However remember ABC can only move up to 20% of the busiest data to the uncompressed tier. Moving the most active data to this tier is a benefit to both the system and the user. Leaving this busy data uncompressed can reduce the overhead of uncompressing data, allowing the system to deliver optimal performance. At the same time the system resources can be more balanced at handling internal system functions.

- **Optimal Phase** - Once the used capacity reaches 70% or greater the system is at the optimal operating state. The system achieved compression ratio starts to climb closer to the expected value. As more capacity is consumed, the system is more efficient at compressing data. In this phase the backend configuration is less dynamic, as the learning phase has led ACE to build a compression pool layout that caters to the workload. If or when the system reaches 100% full, the achieved system compression ratio is close to the expected ratio, however it may slightly vary. As an example, if the system compression is 2.0:1 the achieved ratio may be 1.9:1. Essentially, during this phase compression seems more active. As the capacity is consumed the system will react to achieve maximum expected compression.

ACHIEVED COMPRESSION

The achieved compression is displayed in two ways: system level and storage group level. These are two separate values that represent two different things. Achieved system compression is presenting the overall system compression ratio that includes all written allocations. Because this includes some internal written allocations the system ratio is almost always slightly lower than expected even when the system reaches capacity full status. The other display of achieved compression at the storage group level is only relative to the written allocations from the applications related to that storage group. These achieved compression ratios vary greatly, from 1.0:1 to 16:1 (8k). In addition it is critical to realize that taking an average of all the storage groups achieved ratios may not match the overall system achieved ratio. It is possible however not very likely.

MANAGING COMPRESSION

- **ENABLE COMPRESSION** - Compression is enabled by default when provisioning storage using Unisphere for VMAX, Solutions Enabler or REST API. The Unisphere provisioning wizard includes the compression option as a check box when the storage group is being created. The compression option is available for managed storage groups. Managed storage groups are when a storage resource pool (SRP) is assigned. If there is no SRP assigned the compression option is not available. When compression is enabled, the data sent to the system is sent through the compression path and compressed, when possible, to the best-case track size available in the array. The compressed data is allocated to the appropriate compression pool. When using the modify storage group option to enable compression the data is not immediately compressed. All incoming data is sent through the compression flow and existing data is compressed when accessed and over time. In parallel there is a code function that scans data sets looking for data to be compressed and does so when such data is encountered.

- **DISABLING COMPRESSION** - Disabling compression does not immediately start a decompression process. Just like enabling compression on an existing storage group the data is decompressed when accessed and over time. In parallel the code function finds data that should not be compressed and decompress it. When modifying storage groups if the assigned SRP is changed to none this automatically disables compression.

COMPRESSION DISPLAYS WITH UNISPHERE FOR VMAX

There are two compression reporting levels in regards to the savings achieved from ACE; overall system level and storage group level. Overall system compression can be viewed within the Unisphere capacity report. System achieved compression accounts for all data allocated to the system. Storage group achieved compression can be found in a few different views and accounts for allocations that relate only to it. In addition, there is a compressibility report that provides a possible achieved compression ratio for storage groups where compression is not enabled.

UNISPHERE CAPACITY REPORT

The capacity report presents a system's efficiency using a few sections. This view shows the system compression ratio as well as the capacity usage.

Figure 4 presents the array capacity usage in two factors, subscribed capacity and usable capacity. Subscribed capacity represents the total amount of requested front end host and eNAS capacity plus system-configured capacity such as Guest OS and RecoverPoint (RP) devices. The blue portion of the display includes logical allocated capacity based on a track size of 128K. The usable capacity represents the amount of total physical capacity available using the pool track size of all enabled data devices (TDATs). The blue portion represents the allocated physical capacity for all front end hosts, eNAS as well as internal devices such as Guest OS and RP devices.

Figure 4: Sample of the ARRAY USAGE from the Capacity report.

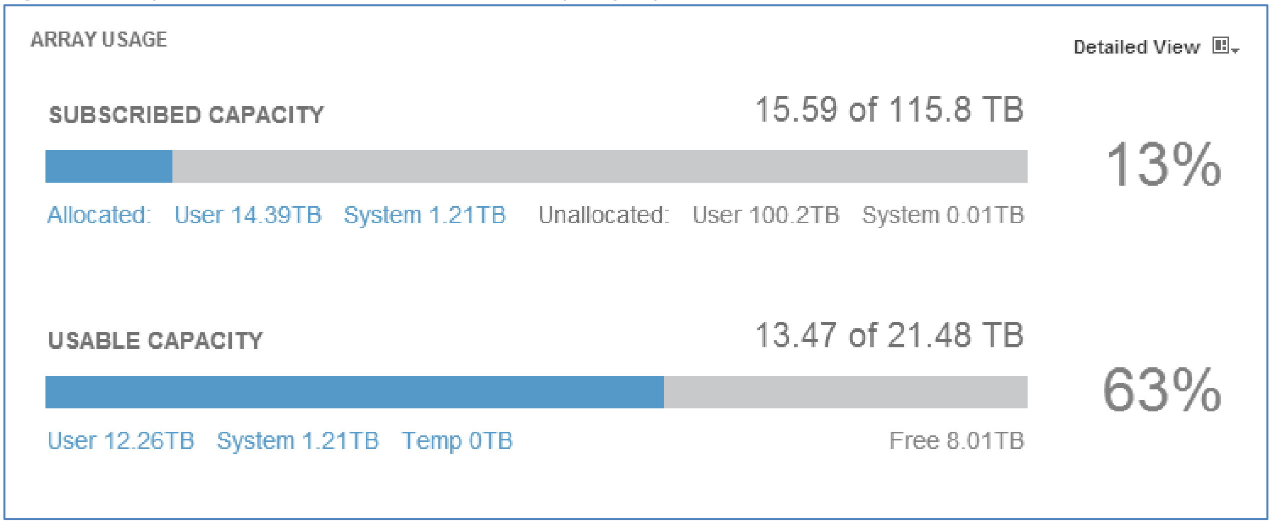


Figure 5 presents the system current overall system compression ratio. The COMPRESSION ENABLED STORAGE percentage represents the total amount of data populating the system where compression is enabled. System compression can also be displayed using Solutions Enabler as seen in the SYMMETRIX EFFICIECNY display (symcfg –sid xxx list –efficiency {xxx = last three digits from the system serial number}), see

Figure 6.

Figure 5: Compression ratio section from the Capacity Report

COMPRESSION RATIO

1.2:1

Virtual Provisioning Ratio **1.2:1**

Snapshot Ratio **4.0:1**

COMPRESSION ENABLED STORAGE 32%

Figure 6: SYMMETRIX EFFICIENCY display in Solutions Enabler

```
  S Y M M E T R I X    E F F I C I E N C Y

------------------------------------------------
          Overall System Efficiency

                    VP Snapshot   Comp  Overall
SymmID           Ratio    Ratio  Ratio    Ratio
------------ -------- -------- ------ --------
000197800130    1.6:1      N/A  1.8:1    2.9:1
```

COMPRESSIBILITY REPORT

The compressibility report provides a list of storage groups that do not have compression enabled. The list displays information specific to that storage group such as number of volumes, allocated capacity, used capacity and target compression ratio. The # of volumes is how many devices are in that storage group. The Allocated capacity and Used capacity reflect how much of the capacity has actually been written to the system. Target ratio presents the user with a potential compression ratio that could be achieved if compression was enabled.

Figure 7 depicts the compressibility report taken from Unisphere for VMAX. This can also be displayed in Solutions Enabler (symcfg –sid xxx list –sg_compression –by_compressibility {xxx = last three digits from the system serial number}) see Figure 8.

Figure 7: Compressibility report in Unisphere for VMAX

Storage Group	# of Volumes	Allocated(GB)	Used(GB)	Target Ratio
test_env	2	200.0	200.0	1.0:1
jfm	2	10240.0	10240.0	1.0:1
NOT_IN_SG	72	1418.8	1418.8	1.6:1

Figure 8: Compressibility report as seen with Solutions Enabler

```
Name                    : SRP_1

                        Number   Allocated        Used  Target
Storage Group Name     Devices        (GB)        (GB)  Ratio
-------------------------------------------------------------
<not_in_sg>                190      5692.5      5692.5   3.0:1
EMBEDDED_NAS_DM_SG          12       108.1       108.1   1.1:1
WM_T13                       4         0.0         0.0     N/A
```

FUNCTIONAL OPERATION

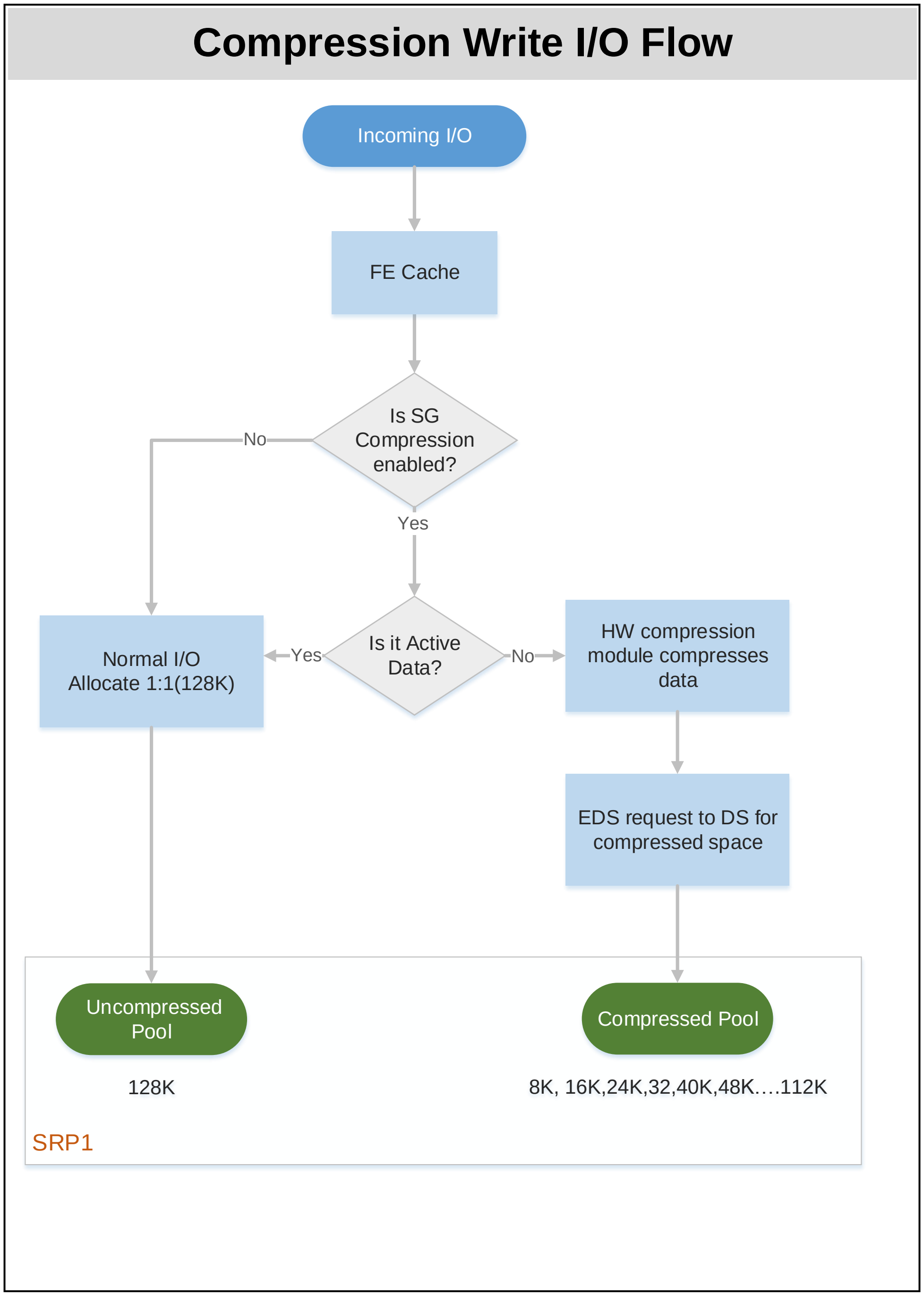
WRITE I/O DATA FLOW

The compression flow for a write I/O sent to the array is based on the following considerations which determine whether to compress or not.

- Whether compression is enabled for the data (related to the storage group compression setting).
- Active or idle data.
- Compressibility of the data, if at all.

Figure 9 shows incoming I/O entering Front End cache, memory space used with the system for incoming data to be processed. The I/O then enters the data flow path and is either allocated as is or compressed and allocated as compressed data.

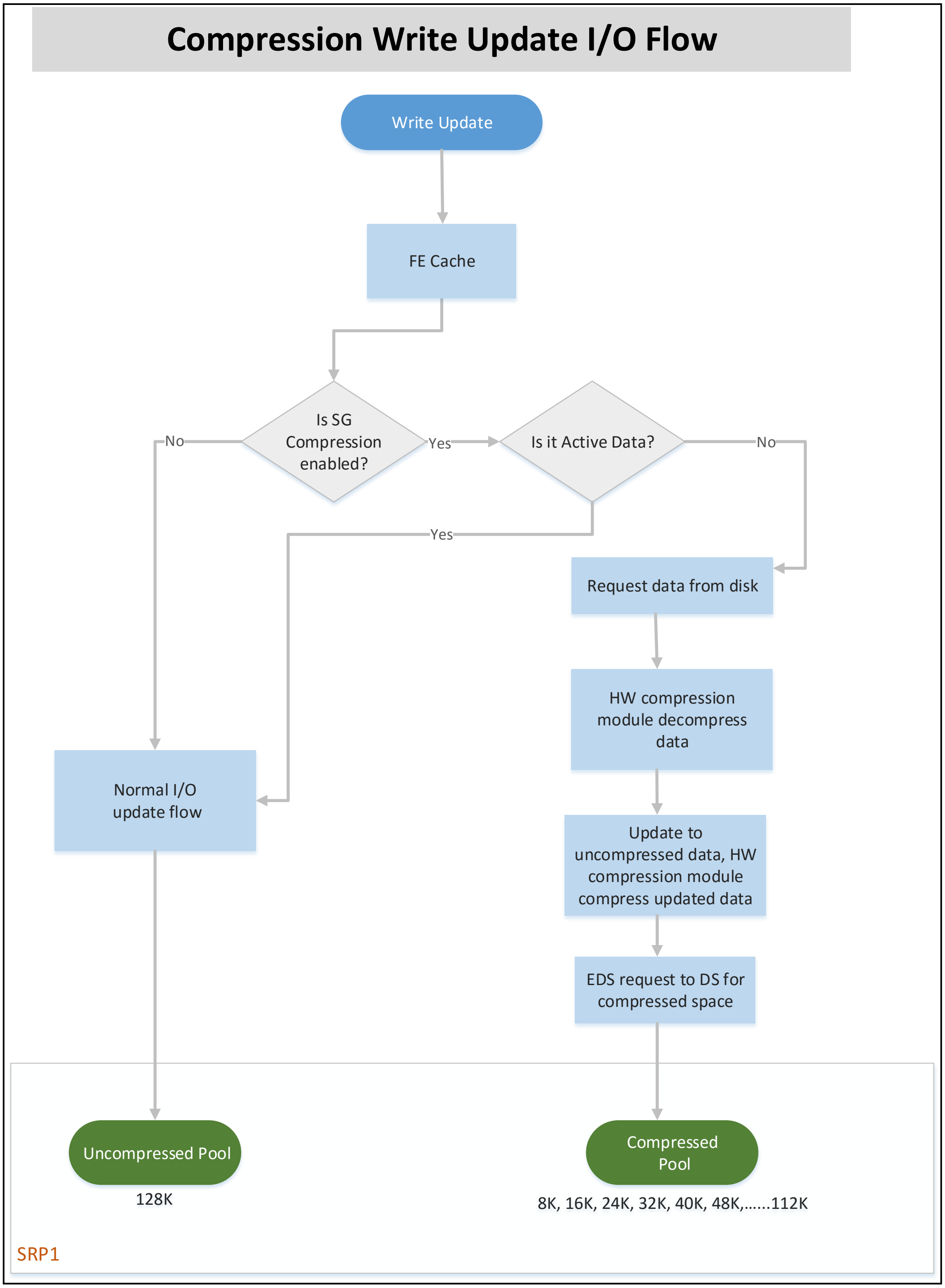
Figure 9: Compression Write Flow



WRITE UPDATE DATA FLOW

Figure 10 shows the compression flow for a write update I/O. An update to existing data that is currently compressed is sent to the array. The data flow checks whether compression is enabled. If so, and the data is compressed it is uncompressed by the compression module before the update can be applied. Once the update is applied the uncompressed data is then compressed again by the compression module. If the resulting compression size is the same, the data is allocated back to the same compression pool. If the update changes the compressed size, the data is allocated to the appropriate compression pool.

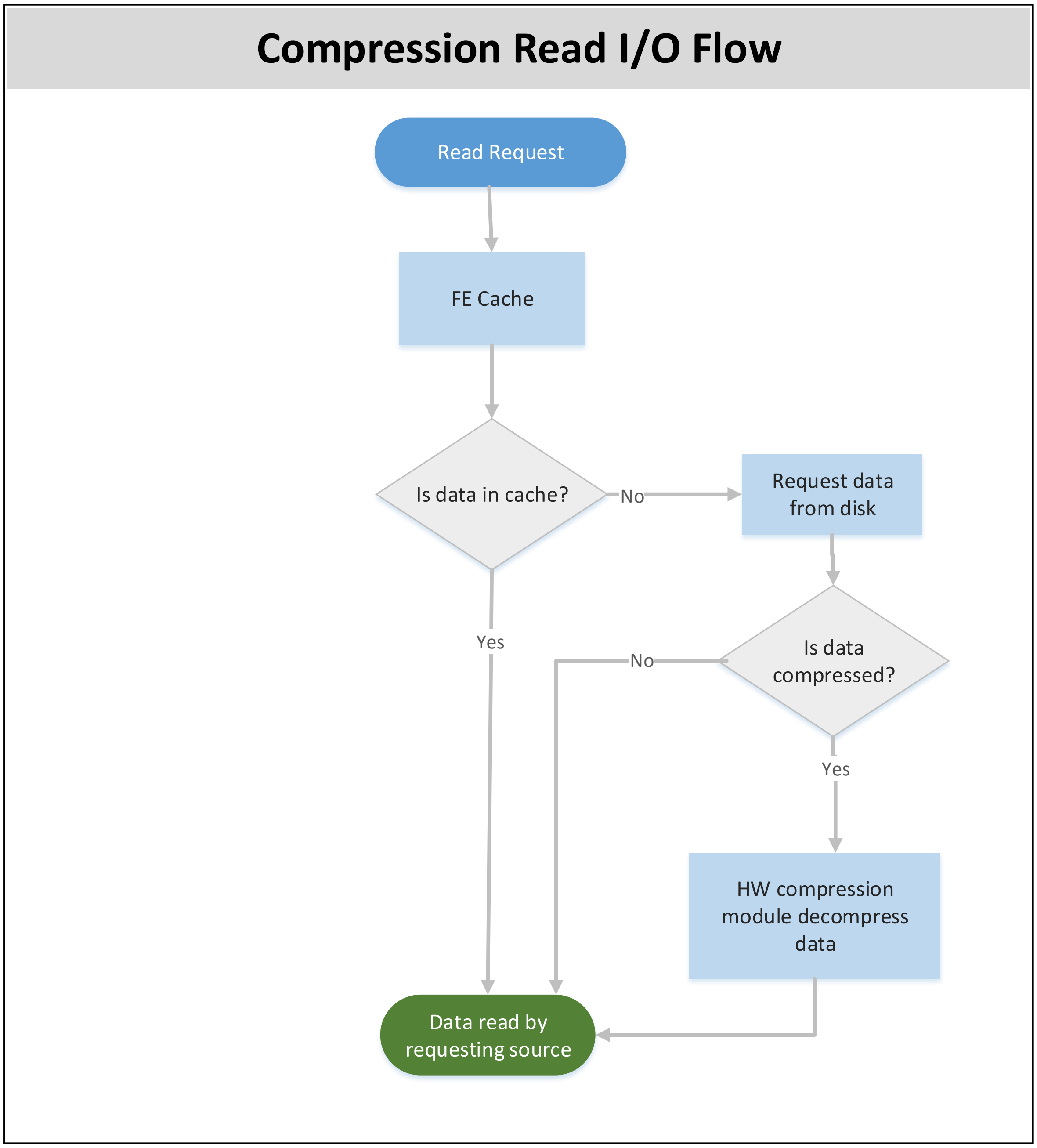
Figure 10: Compression Write Update I/O Flow



READ I/O DATA FLOW

Figure 11 shows the data flow for read requests. When a read request is sent and the data is in cache the data flow has no change. Data to be read that is in cache is a read hit and is presented to the requesting source. For the case of a read miss, the data needs to be requested from disk and then presented to the requesting source. When the requested data is compressed it is uncompressed and placed in cache for the requesting source to read it.

Figure 11: Compression Read I/O Flow



DATA SERVICES

All data services offered on the VMAX All Flash array with compression enabled are supported. This includes local replication (SnapVX), remote replication (SRDF), D@RE and VVols.

LOCAL REPLICATION (SNAPVX)

Compression is supported with the use of local replication features; there are multiple variations and use cases for local replication. Below are the details regarding the different local replications sessions that can exist. For more detail regarding Local replication and SnapVX see the TimeFinder and HYPERMAX OS Local Replication Technical Note available at dellemc.com.

NOCOPY SESSIONS (SnapVX, VP Snap)

Uncompressed source data remains uncompressed when becoming snapshot data, and may be compressed later as it becomes less active. However, activity to a snapshot through a linked target may prevent uncompressed data from being compressed. Similarly, compressed source data remains compressed when becoming snapshot data. Read activity to a snapshot through a linked target may cause the compressed data to be uncompressed.

The compression setting of a linked target only affects data written directly to the linked target and does not affect the snapshot data.

COPY SESSIONS (SnapVX Full Copy Linked Targets, Clone, Mirror)

The compression settings for both the source and target are taken into account for copy sessions.

When compression is enabled on the source the data is decompressed before copying to the target. When compression is enabled on the target the data is compressed before being allocated to the target. Likewise, when compression is enabled on both the source and target the data is decompressed before the copy and then compressed to allocate for the target.

Copy times may vary due to decompression and compression of the data. It is not recommended to change the compression settings in between differential operations (i.e. disabling compression before each differential operation and then again after the copy completes) as this causes data to go through needless compress/decompress cycles.

REMOTE REPLICATION (SRDF)

Compression for SRDF is already supported and known as SRDF compression. SRDF compression is a feature designed to reduce bandwidth consumption while sending data to and from systems connected using remote replication. SRDF compression and the Adaptive Compression Engine (ACE) both use the same compression module however exist for different purposes. Data that has been compressed using ACE is uncompressed before being sent across the SRDF link. In the event that SRDF compression and inline compression apply, the data is uncompressed by the module and then compressed using the SRDF compression function and then sent across to the remote site.

DATA AT REST ENCRYPTION (D@RE)

Compression is performed as part of the I/O flow process. Data at rest encryption is performed at the back-end I/O module. This means that the data being encrypted is already compressed.

VIRTUAL VOLUMES (VVols)

Compression is supported for the allocation of data to VVols and follows the same I/O path as all data as shown in figures 2 to 4. Compression as a feature is not included as a VVols resource to be used at the host.

CONCLUSION

The use of physical storage capacity is a common concern of storage administrators across the enterprise storage industry. The constant and ever growing amounts of data have created the need for more efficiency in the use of physical capacity. The VMAX All Flash system and the Adaptive Compression Engine do more than simply offer capacity savings. Enhanced technology included in the hardware allows the system to consume less data center real estate. Intelligent algorithms and hardware acceleration allow the system to offer exceptional performance as well as capacity savings. This delivers on capacity savings as well as other areas of concerns, less data center footprint and fewer components (less physical drives) lead to lower power and cooling costs.   In addition to the savings, compression is as simple as a single click to enable or disable. The system handles all the work.