INTRODUCTION TO XtremIO VIRTUAL COPIES

Abstract
This white paper introduces XtremIO Virtual Copies as a unique approach to writeable in-memory copies that are totally space-efficient. It describes best practices and key aspects of XtremIO’s Virtual Copies technology.

March 2016
Table of Contents

Executive Summary ........................................................................................................... 4
Conventional Snapshots .................................................................................................. 5
  Conventional Volume Management .............................................................................. 5
  Copy-on-Write Snapshots ......................................................................................... 5
  Redirect-on-Write Snapshots ..................................................................................... 7
  Other Copy Technologies ............................................................................................. 10
  Conventional Snapshots - Efficiency and Performance ................................................ 10
  Legacy Snapshots Use Cases ....................................................................................... 11
Introduction to XtremIO Virtual Copies ....................................................................... 12
  Creating XtremIO Virtual Copies .............................................................................. 12
  Capabilities ................................................................................................................ 13
Comparison ................................................................................................................... 14
Architecture Advantages ............................................................................................... 15
  Cross-Consistent Copies of Multiple Volumes ............................................................ 19
  Consistency Groups .................................................................................................. 20
  Snapshot Sets ............................................................................................................. 20
  Existence Bitmap ........................................................................................................ 21
  Virtual Copy Deletions ............................................................................................... 22
  No Garbage Collection ............................................................................................... 22
  Volume Snapshot Groups ......................................................................................... 23
  Remove Shadow Writes ............................................................................................. 24
  Refresh Capabilities .................................................................................................. 25
    Refresh Guidelines .................................................................................................. 27
  Tags .............................................................................................................................. 27
Use Cases ...................................................................................................................... 28
  Offloading Backup Operations .................................................................................... 28
    Key Benefits ............................................................................................................ 28
    Creating an XVC for Backup Use - Behind the Scenes ............................................. 29
  Testing and Development Use Cases ......................................................................... 31
    Key Benefits ............................................................................................................ 34
Logical Data Protection .................................................................................................. 34
Offload Processing and Data Analytics ......................................................................... 36
    Key Benefits ............................................................................................................ 37
Provision Bulk VMs by Using XVC on a Virtual Machine File System (VMFS) ................... 38
Conclusion .................................................................................................................... 39
Executive Summary

XtremIO arrays leverage the XtremIO Virtual Copy, or XVC technology. XVC technology abstracts the copy operation as a unique, in-memory metadata operation with no back-end media impact. XVC enables instantaneous, high performance copies of any data set in almost any quantity desired. It is entirely space-efficient, with data services such as inline deduplication and compression, and does not impact production or other copies.

Critical business processes typically require multiple copies of each database instance and application data for various purposes such as development, analytics, operations, and data protection. To improve organizational agility and competitiveness, "more is more" (more copies more frequently, with more operational self-service across process cycles). While there are several approaches for copy data management (CDM, as described by Gartner and IDC), such approaches all struggle fundamentally with storage sprawl, limited copy frequency, limited performance and complex operational processes.

With its "bullet-proof" consistent IOPS and latency, and its ability to scale out more performance as needed while causing no application downtime, XtremIO delivers incredible performance to production and non-production applications, without impacting the production SLAs.

XVC technology enables instantaneous creation of copy images for volume data, capturing the data exactly as it appeared at the specific point in time during which the copy was created. XVC enables users to save the volume data state and access the specific volume data at a later time (as needed), even after the source volume has changed.

XVCs are implemented in a unique way, which for the first time maintains space efficiency on copies (writeable and read-only) for both metadata and data. Combined with XtremIO’s unique in-memory metadata architecture, XVC allows for large numbers of high performance, low latency read/writeable copies.

XVCs are efficient in metadata and physical space, can be created instantaneously, have no performance impact, and provide the same data services as any other volume in the cluster (such as thin provisioning and Inline Data Reduction).

XVCs are a key foundation of XtremIO’s iCDM, enabling multiple use cases including:

- Test and development environment management
- Database analytics and offload processing
- Data protection
- Operation
- Bulk provisioning of VMs

XVCs are easy to use and appear and are managed just like standard volumes in the cluster.
Conventional Snapshots

Conventional snapshots were invented primarily for the purpose of providing data protection in a space-efficient way. Taking a snapshot of a volume, or of a group of volumes, creates a point-in-time copy of the original data set to which the user can roll back to, if and when it is needed.

Conventional Volume Management

In conventional block storage arrays, a logical volume is the range of logical addresses within that volume.

![Figure 1: Conventional Volume Management](image)

The logical addresses are mapped to the physical data blocks on the disk. The actual mapping procedure is the responsibility of the logical volume manager. This procedure can be straightforward or complex, depending on factors such as thin provisioning, tiering, deduplication, compression and other factors.

Copy-on-Write Snapshots

Legacy implementations of snapshots were based on a technology called "copy-on-first-write". In this scheme, metadata pertaining to where original data is stored is copied at the time of the snapshot's creation, and a new storage pool is defined and reserved in the cluster for the snapshot. No physical copy is produced yet. Every new write triggers a data movement operation between the production volume and the reserved snapshot pool.

The I/O flow is as follows:

1. The cluster receives the new write.
2. The system reads the original data from the production volume.
3. The cluster writes the original data to the reserved snapshot pool.
4. The cluster overwrites production with the new data.
With this scheme, only the metadata pertaining to where original data is stored is copied at the time of the snapshot's creation. No physical copy of the data is performed. The volume manager then tracks the changing blocks on the original volume while writes to the original volume are performed. The original data being written to is copied to the designated storage pool, and set aside for the snapshot before the original data is overwritten (hence the name "copy-on-write").

This means that every write has a penalty of an additional two I/O operations. One may imagine that when using high performance media such as SSDs, the approach would be less problematic. This is not necessarily true, as the methodology still poses significant overheads in terms of data movement management. It impacts the latency for both write and read operations, limits the flexibility and performance for complex snapshot topologies (such as snapshots of snapshots), and poses a negative impact on the SSD media endurance.

**Figure 2: Copy-on-Write – Host Write**

**Disadvantages:**

- Metadata is copied when a snapshot is taken, consuming time and capacity, making it impossible to manage it in memory.
- The reserved snap pool must often be allocated up front, even if not fully utilized, and can run out of space.
- Performance is heavily penalized, especially for writes.
- Complex snapshot topologies experience severe performance degradation.
Read operations are performed in a similar way. Read I/Os to the production volume are always served from the production data reserved pool. Reads to the snapshot are served from the production pool for blocks that remain unchanged, and are served from the snapshot reserved pool for blocks that have changed.

**Disadvantage:**
- Complex snapshot topologies experience severe performance degradation.

**Figure 3: Copy-on-Write – Host Read**

**Redirect-on-Write Snapshots**

With this scheme, the only thing that is copied at the time of the snapshot's creation is the metadata pertaining to where original data is stored. No physical copy of the data is performed. The volume manager tracks the changing blocks on the original volume while writes to the original volume are performed. However, when new data is written to the production volume, the data is written directly to the storage pool and the volume manager updates (redirects) the production volume's metadata to the new physical data location (hence the name "redirect-on-write").
8

Introduction to XtremIO

Virtual Copies

Figure 4: Redirect-on-Write - Host Write

Disadvantages:

- Typically, metadata is copied when a snapshot is taken, consuming time and capacity.

- In some cases additional operations are required to make snapshots read/writable.

- Some implementations do not copy metadata when snapshots are taken, as they are read only. However, once the snapshot is to be accessed in read/write, metadata is copied, thus consuming time and memory.
When a snapshot is created, the metadata of the production, and that of the snapshot volumes within the volume manager, point to the same physical blocks. Once the production volume receives new writes (assuming that the logical block addresses [LBAs] of blocks B and D are overwritten), its metadata pointers update to the new block’s location.

Reads are performed using the metadata pointers in the volume manager in order to determine where the physical block resides.

**Disadvantage:**
- As every snapshot consumes high amounts of metadata in memory, and the amount of memory in the array is limited, it is necessary to de-stage metadata to SSD, which impacts performance, even on an all-flash array.

**Figure 5: Redirect-On-Write - Host Read**

Redirect-on-write snapshots are data-efficient, yet they are not metadata-efficient, as the methodology involves copying the original volume’s entire metadata set during the snapshot creation process.
Other Copy Technologies

Other technologies are available for making copies of volumes. Clones, which are typically taken from a static snapshot, provide a full copy of the data (since clones are typically made to physically separate hard drives or SSDs). Whereas clones can potentially provide the same performance as production volumes in the cluster, a clone’s creation time is typically very lengthy as data is copied to the clone. SLAs may be impacted during the clone operation production. In addition, clones are inefficient because they consume double the capacity and metadata.

Split mirrors are used to create clones with more efficiency from a dynamic source. Mirroring is established between the production volume and the clone (a synchronization process that remains on-going until the clone catches up with the production). Once achieved, administrators can split the clone in order to provide an independent copy of the data.

Both technologies offer good performance. However, creation and refresh times are very long when using either of these options. Both clones and split mirrors are inefficient in metadata and data capacity.

Conventional Snapshots - Efficiency and Performance

Writes to snapshots result in fragmentation, with the same result holding true for both redirect-on-write and copy-on-write. Furthermore, when multiple snapshots are created, access to the original data and the tracking of metadata changes for all snapshots, and fragmentation and reconciliation upon snapshot deletion results in heavy performance penalties.

Deleting a redirect-on-write snapshot involves scanning and processing the snapshot metadata, and removing any corresponding data blocks belonging exclusively to the snapshot, hence potentially moving data from the snapshot reserve pool to the production volume pool. The time taken for this process to complete is proportionate to the original volume size, as opposed to being proportionate to the amount of changed blocks from those in the original volume (since the snapshot's creation).

Copy-on-write snapshots are not efficient in writes to the original volume. Both copy-on-write and redirect-on-write snapshots have to deal with data fragmentation over time, as well as significant metadata changes. This means that once a snapshot is taken, the I/O performance on the original volume is often negatively impacted.

Such performance issues cannot be mitigated in legacy dual-controller architecture, let alone with active-passive architecture. The main reason for this is that, per volume or snapshot, only one controller (or at best two controllers) can be involved in the volume’s or snapshot’s management. Scalability is not possible and a large number of snapshot and production volumes create a significant overhead on the controller, impacting the performance on snapshots, production volumes, or both.
Legacy Snapshots Use Cases

Snapshots were originally created and used for short periods of time, typically to create a copy of "live" production data for backup purposes. Snapshots enabled administrators to freeze the production application for a short period of time, take a snapshot, and then resume normal operations. These actions resulted in a static copy of the production data, which was typically accessed in read-only mode, and could be backed up to an external backup device. The reasons why snapshots were only used for short periods of time included performance issues, capacity utilization considerations, and the limited number of supported snapshots.

Following the development of redirect-on-write technologies, snapshot usage was extended for longer periods of time, mainly for testing and development processes. However, snapshot usage was limited, because in most cases performance was impacted, either due to copy-on-write snapshot implementations, posing a high penalty on the production environment's performance, or due to redirect-on-write snapshots introducing increased read-latency, resulting from linked metadata data scanning in the array or overheads on the controllers’ CPUs and data fragmentation.
Introduction to XtremIO Virtual Copies

XtremIO Virtual Copies are either writeable or read-only. Read-only copies enable maintaining immutability of the copies. It is possible to create a virtual copy from a production volume or from a copy of any other production volume.

XVCs can be used in a number of use cases, including the following examples:

- **Logical Corruption Protection**: Create frequent point-in-time copies (according to RPO intervals – seconds, minutes, hours) and use them to recover from any logical data corruption. An XVC can be kept in the system for as long as needed. If logical data corruption occurs, it is possible to use a point-in-time copy of an earlier application state (prior to the occurrence of the logical data corruption) to recover the application to a known 'good' point in time. It is also possible to perform a complete restore of production volumes from the backup copy.

- **Backup**: Create virtual copies to be presented to a backup server/agent. The copies can be used in order to offload the backup process from the production server.

- **Test and Development**: Create virtual copies of the production data, and then create multiple (space-efficient high-performance) copies of the production system, to present for test and development purposes. Since XtremIO Virtual Copy technology enables unlimited copy hierarchy, multiple test and development processes are supported. It is possible to refresh Test and Dev environments with new or updated production data. The refresh operation is easy and immediate. The SCSI entities used by the test and development servers are preserved during the refresh, changing only the underlying data, and hence avoiding host side SCSI bus rescans. All the environments can be easily scripted via CLI or RESTful API.

- **Near Real-Time Data Analytics**: Use XVC technology as a means of offloading the processing of data such as ETL from the production server. For example, if there is a need to run a heavy process on the data (potentially effecting the production server's performance), it is possible to use XVC to create a current copy of the production data and mount it on a different server. This process can then be run on the other server without consuming the production server's resources. This capability enables on-demand near-real-time analytic capacities.

Creating XtremIO Virtual Copies

For detailed instructions on creating XVCs from volumes, sets of volumes or Consistency Groups, refer to the *XtremIO Storage Array User Guide*.
Capabilities

XtremIO Virtual Copies provide the following capabilities:

- Copies are created instantaneously, providing a workable, writeable copy of the production volume.
- Copies are either read-write or read-only* copies of source volumes.
- Copies are regular volumes in the cluster.
- Copies have the same data services as any volume in the system, whereby the inline global deduplication and thin provisioning features are in constant operation.
- Copies are both metadata and data efficient.
- Copies require no reserved space.
- The system supports a consistent copy image on multiple volumes (Consistency Group).
- Copies can be created from any copy, at any hierarchy level or span.
- Deletion of a volume, or any of its XVCs, does not affect the child copy or parent XVC/volume.
- The system provides predictable and consistent performance on production volumes or any copies.
- Production volumes can be easily restored from any of the backup copy images.
- Test and Dev environments can be easily updated or refreshed with new information, while preserving all SCSI information (therefore eliminating the need for SCSI-BUS rescan on the host side).

*XVCs that are created as read-only are immutable. To gain write access to a read-only copy, a new R/W copy needs to be created from the source read-only copy.
Comparison

Table 1 compares some of the key differences between the various copy technologies and the XtremIO Virtual Copies.

**Table 1: Legacy Copy Technologies vs. XtremIO Virtual Copies**

<table>
<thead>
<tr>
<th></th>
<th>Copy-on-Write</th>
<th>Redirect-on-Write</th>
<th>Full Clones</th>
<th>XtremIO Virtual Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space-Efficient Data</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes + Inline Data Reduction</td>
</tr>
<tr>
<td><strong>Space-Efficient Metadata</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Volumes and Snapshots Metadata</strong></td>
<td>Metadata is served from disk and memory.</td>
<td>Metadata is served from disk and memory.</td>
<td>Metadata is served from disk and memory.</td>
<td>Metadata is always 100% in-memory.</td>
</tr>
<tr>
<td><strong>Creation Time</strong></td>
<td>Instantaneous</td>
<td>Instantaneous</td>
<td>Long time</td>
<td>Instantaneous</td>
</tr>
<tr>
<td><strong>Performance Impact on Production</strong></td>
<td>High impact</td>
<td>Moderate impact</td>
<td>No impact after clone is completed</td>
<td>No impact</td>
</tr>
<tr>
<td><strong>Performance of Snapshots</strong></td>
<td>Degraded</td>
<td>Degraded</td>
<td>Can be as production.</td>
<td>Same as production</td>
</tr>
<tr>
<td><strong>Rapid Deletes</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Delete Limitations</strong></td>
<td>Limited</td>
<td>Limited</td>
<td>N/A</td>
<td>Any copy in the hierarchy tree can be deleted without effecting the parents or children.</td>
</tr>
<tr>
<td><strong>Topology Limitations</strong></td>
<td>Limited</td>
<td>May support snapshot on snapshot</td>
<td>No snap on snap support</td>
<td>Any topology</td>
</tr>
<tr>
<td><strong>Up-front Space Reservation</strong></td>
<td>Yes</td>
<td>May require space reservation</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Data Services Limitations</strong></td>
<td>Yes</td>
<td>May limit data services</td>
<td>No</td>
<td>No. Full data services are available.</td>
</tr>
<tr>
<td><strong>Instant Restore/Refresh Between any copy Hierarchy Child</strong></td>
<td>No</td>
<td>Very limited</td>
<td>May require re-synching and may impact production SLA.</td>
<td>Yes. Instant Restore/Refresh between any copies in the topology.</td>
</tr>
</tbody>
</table>
Architecture Advantages

XtremIO Virtual Copies architectural advantages include:

- The write process and performance are the same for both the production volume and its copies.

- Space-efficient metadata:
  - Creation of a virtual copy does not consume metadata.
  - Metadata is only consumed for new and globally unique data blocks.

- No performance impact:
  - Creation of virtual copies causes no impact.
  - Read performance is equal on all copy hierarchy levels.

- Highly scalable:
  - Supports high numbers of copies.
  - Supports a large number of Consistency Groups.

- Inline Data Reduction-enhanced "redirect-on-unique-write" method:
  - Only consumes space for new and globally unique data blocks.
  - No physical data movement is incurred on new writes.

- No tuning, no optimization; a constant, even distribution of system resources:
  - The cluster's Storage Controllers are all constantly engaged in managing the I/O data flow and metadata, regardless of the entity type.
  - More CPU power and memory remains available (via a single controller versus multiple controllers).
  - Consistent, even distribution of workload is provisioned across all available resources.

- Optimized for flash memory:
  - XtremIO Virtual Copies are optimized for maximum flash endurance.
  - No data movements are incurred on copy creation or during writes.
  - Inline Data Reduction provides added value in capacity efficiency and flash endurance.
  - Additional benefits are provided with flash media (in terms of performance).
  - The system provides efficient metadata and capacity utilization.
XtremIO’s virtual copies technology is implemented by leveraging the array’s content-addressing capabilities along with in-memory metadata, and the system’s SSD media-optimized dual stage metadata (providing Inline Data Reduction), with a unique metadata tree structure that directs I/Os to the right data timestamp. This leveraging enables efficient copy technology that sustains high performance, while maximizing media endurance, both in terms of ability to create multiple copies, and the amount of I/Os that a copy can support.

Figure 6: Address-to-Content Mapping

Figure 6 is a logical volume diagram, showing that every written block address is mapped to a fingerprint. This mapping is called Address-to-Content mapping. Additionally, there is a separate metadata mapping of the content to the actual unique physical blocks that are written to the SSDs (thus forming the dual-stage metadata structure).

As every XtremIO volume or virtual copy is thinly provisioned, the addresses that have not been written to remain empty, and do not occupy metadata (or data) space. Therefore, XtremIO’s thin provisioning is 100% space efficient.

When creating a virtual copy, the system generates a pointer to the ancestor metadata of the actual volume in the cluster. Therefore, creating a copy is a very quick operation that does not impact the cluster, and does not consume physical or logical capacity, and therefore, does not impact the production SLA. Virtual copy capacity consumption occurs only if a change requires writing a new block.

Figure 7: XtremIO Virtual Copies
When a copy is created, the volume's existing metadata becomes an "ancestor" entity that is shared between the production volume and the copy. New empty containers are created for subsequent changes to both the production volume and the virtual copy volume. Therefore, the act of creating a copy is instantaneous and involves no data or metadata copies.

When a new block is written to the ancestor, the system updates the ancestor volume's metadata to reflect the new write, and stores the block in the cluster. This is done by using the standard write flow process. As long as this block is shared between the copies and the ancestor volume, it is not deleted from the cluster following a write. This applies to both a write in a new location on the volume (an unused LBA), and to a rewrite on an already-written-to location.

The cluster manages both the copy's metadata and the ancestor's metadata via a tree structure.

Figure 8 shows the copy and the ancestor volumes, represented as "leaves" in this structure.

![Figure 8: Metadata Tree Structure](image)

The metadata is shared between all copies that remain unchanged (from the copy's original ancestor). The copy maintains unique metadata only for an LBA with a data block differing from the existing data, thus providing economical metadata management.

When a new copy is created, the cluster always creates two leaves (two descendant entities) from the source entity. One of the leaves represents the copy, and the other one becomes the source entity. The original metadata container for the source entity is no longer used directly, but is kept in the cluster for metadata management purposes (only).
Figure 9 illustrates a 16-block volume in an XtremIO system. The first row (marked as $A_{(t0)}/S_{(t0)}$) shows the volume at the time that the first virtual copy was created ($t_0$). At $t_0$, the ancestor ($A_{(t0)}$) and the copy ($S_{(t0)}$) have the same data and metadata, because $S_{(t0)}$ is the read-only copy of $A_{(t0)}$ (containing the same data as its ancestor).

Note: Of the 16 blocks, only eight blocks are used. Blocks 0 and 4 consume only one block of physical capacity, as a result of deduplication. The blanked, dotted blocks represent the blocks that are thinly provisioned and do not consume physical capacity.

As shown in Figure 9, before creating the virtual copy at $S_{(t1)}$, two new blocks are written to $P$:

- H8 overwrites H2 at LBA 3.
- H2 is written to LBA D. The data does not consume more physical capacity, since it has an identical fingerprint to that of the data stored in LBA 3 in $A_{(t0)}$ (H2).
S_{02} is a read/write copy. It contains two additional blocks (at LBA 2 and LBA 3) that differ from its ancestor source entity.

Unlike traditional snapshot implementations (that reserve space for changed blocks, and reserve an entire copy of the metadata for each snapshot), XtremIO does not require reserved physical space for its virtual copies, and never has metadata "bloat". Virtual copies only use such resources when needed, and the resources are consumed from the cluster's global resource pool. There is no pool management in XtremIO.

All accessible entities in the copy tree representing the volume, including all copies originating from that volume are managed by an entity called a Volume Snapshot Group (VSG).

An XtremIO Virtual Copy only consumes metadata for new writes (unshared blocks), and specifically utilizes shared metadata from the copy’s ancestor entities. This allows the cluster to efficiently maintain large numbers of copies using a very small storage overhead, which is dynamic and proportional to the amount of changes in the entities.

For example, at time t2, LBAs 0, 3, 4, 6, 8, A, B, D and F are shared with the ancestor’s entities. Only LBA 5 (H6) is unique for this copy. Therefore, XtremIO consumes only one metadata unit. The remaining blocks are shared with the ancestors and use the ancestor data structure in order to compile the correct volume data and structure.

**Cross-Consistent Copies of Multiple Volumes**

XtremIO supports the creation of virtual copies on a set of volumes. All copies taken from the volumes in the set are cross-consistent. Copies on a set of volumes can be created manually, by either selecting a set of volumes for copy operations, or by placing volumes in a Consistency Group container and creating a copy of the Consistency Group. A Snapshot Set (which is a logical object) is created to correlate with all of the created copies.

The cluster quiesces the volumes within microseconds, thus guaranteeing that newly-created virtual copies remain cross-consistent. There is no impact on the system performance even when this operation is repeated in short intervals. Only the source volumes are quiesced during the virtual copy creation operation.

In order to guarantee consistency, the cluster temporarily holds any acknowledgments back to the hosts for any writes to the source volumes during the quiesce procedure, thus guaranteeing that no new writes are generated from the initiator during the quiescence. As a result, all copies that are created are cross-consistent.
Consistency Groups

XtremIO enables grouping of multiple volumes for data protection and for other use cases. When volumes are placed together in a Consistency Group, it enables the creation of a cross-consistent copy on all members in the Consistency Group.

Whenever a reoccurring need arises to create copies of multiple volumes, it is advisable to place multiple volumes together. The result of creating a virtual copy on a Consistency Group is that each of the Consistency Group's member volumes creates a copy.

A Snapshot Set (which is a logical object) is created to correlate with all of the created copies.

When a volume is added to a Consistency Group, it is assigned a position (offset). This offset position is used during restore and/or refresh operations to correlate between the right object to be refreshed. The offset is maintained within the Snapshot Set object.

A volume can be a member of multiple Consistency Groups. Refer to the current XtremIO Release Notes document for the most up-to-date information on Consistency Group numbers, restrictions and limitations.

Snapshot Sets

Whenever a virtual copy is created, the system creates a new Snapshot Set object to represent the point-in-time copy. This occurs irrespective of whether or not the source object is a single volume or a set of volumes (a volume list or a Consistency Group).

A Snapshot Set is a logical representation of a specific point-in-time, and maintains all newly-created volumes to present that point-in-time. It can be used to correlate with different volumes created from the same copy creation operation.
**Existence Bitmap**

XtremIO’s Storage Array has an additional data structure which is called an existence bitmap.

The implementation of the XtremIO Virtual Copy technology enables taking a copy of a copy without restrictions. In systems that allow such cascading copies, finding the location from which the data should be retrieved is a challenge that may impact the read performance. For each LBA in the copy, the data may be found either in that copy’s volume itself (in cases where that LBA was written to after the copy was created), or in one of its ancestors.

The native algorithm for reading from a space-efficient copy is to check whether or not the data was written to the copy’s volume itself, and if not, to check its parent, and so on. In some cases, such as when that LBA was never written to, the original ancestor must be located, resulting in a lengthy search. This algorithm has a high performance penalty on reads. Furthermore, the performance is unpredictable, as it depends on the length of the chain. In addition to that, copies that are situated far away from the root volume experience worse performance than copies that are closer to the root.

The existence bitmap data structure and algorithm optimize read operations on space-efficient copies. The existence bitmap provides predictable, even performance for all copies, regardless of their distance from the root volume.
There is one structure per Volume Snapshot Group (VSG) containing bitmaps; one for each LBA in the original volume. The number of bits in each bitmap is equal to the maximal number of volumes per VSG. Whenever data is written to a volume, the corresponding bit to that volume is set in the corresponding LBA’s bitmap, regardless of whether it is the original volume or a copy.

The bitmap in Figure 10 (on page 21) maps, per LBA, where the LBA is written to per copy. Each index in the bitmap represents a copy. For example, only $A(t_0)$ is written to LBA 0. Therefore, every read of LBA 0 for copies created after time $t_1$, should be served by the metadata information stored at $A(t_0)$.

When reading data from a specific LBA, the cluster first reads the bitmap associated with the specific LBA (which typically fits on one cache line, and is very efficient). It then finds which volume, within the bitmap, should be accessed in order to read the data. The cluster then goes directly to that volume to retrieve the data. As bitmap manipulation is an inexpensive operation in terms of performance, the depth of the copy in the copy chain neither affects the read performance, nor the write performance.

**Virtual Copy Deletions**

XVC deletions are only proportional to the amount of changed blocks between the entities. The cluster uses its content-aware capabilities to handle copy deletions.

Each unique data block has a counter indicating the number of its instances of that block in the cluster. When a block is deleted, the counter value is decreased by one. Any block with a counter value of zero – indicating that there is no LBA across all volumes or copies in the cluster referring to this block – is overwritten by XtremIO Data Protection (XDP), when new unique data enters the cluster.

Deleting a copy in the middle of the tree triggers a process that merges the metadata of the deleted entity’s children with that of their grandparents. This process ensures that the tree structure is not fragmented.

**No Garbage Collection**

With XtremIO, every block that needs to be deleted is immediately marked as "free". Therefore, there is no garbage collection in the SSD, and the cluster does not have to perform a scan process to locate and delete orphan blocks.

XtremIO’s Virtual Copies implementation is entirely metadata driven and leverages the array’s Inline Data Reduction to ensure that data is never copied within the array. Thus, many concurrent copies can be maintained.
**Volume Snapshot Groups**

A Volume Snapshot Group is an entity that represents all external entities (mappable entities) of a copy tree. All copies originating from a single ancestor share the same Volume Snapshot Group, which is created whenever a new volume is defined in the cluster.

The VSG information can be viewed in the GUI's Volume Properties pane.

![Figure 11: Volume Properties Pane in GUI](image)

The VSG information can also be viewed under the VSG index by using one of the following CLI commands:

- `show-volumes`
- `show-volume`

Use the following CLI command to list the information on all existing Volume Snapshot Groups in the cluster:

- `show-volume-snapshot-groups`
Remove Shadow Writes

Writes to the same LBA on a volume, and that of its copy, frees the LBA data in the entity’s shared ancestor. This improves array utilization, both in terms of metadata consumption and physical capacity consumption.

Once a virtual copy is created, the result is a simple structure with the following three entities:

- The ancestor entity
- The copy (new entity)
- A new production entity

The example in Figure 12 shows a volume that is completely full with unique non-deduplicated data and all LBAs containing data.

Observe the impact on managing the data and metadata under the following scenario: the copy is written on LBA 4, and after a while LBA 4 on the production is over-written.

**Figure 12: Removing Shadow Writes**

When the virtual copy is initially created, both the production and the copy contain no unique data. All reads are served using the metadata and data of the ancestor entity.
Assume that the later writes to LBA 4 were written both for the production and the copy. The ancestor LBA 4 is now shadowed by the updates to LBA 4 on the copy and the production volume. Therefore, the data in the ancestor is no longer required and can be purged from the cluster. The cluster frees the metadata related to LBA 4 and decreases the reference count of the fingerprint stored on LBA 4. If this fingerprint is not used elsewhere in the cluster, the fingerprint and its corresponding content are deleted from the cluster, freeing both physical and logical capacity. Removal of shadow writes occurs asynchronously, without impacting the cluster performance.

**Refresh Capabilities**

XtremIO has the ability to attach a SCSI personality to any entity within the tree. XtremIO refresh is invoked from a command that creates a new copy, and attaches the new copy to an existing SCSI personality. Therefore, the operation enables refreshing any mapped volume to any of the copies in the in-memory copy tree. The XtremIO refresh process is invoked by performing the following procedure:

1. Select the SCSI personality to be refreshed.
2. Select the entity from which you want to refresh from.
3. Set whether or not the refreshed source entity should be discarded or kept in the system.
The following diagrams illustrate the XtremIO refresh process of a volume from one of its virtual copies.

Assume a volume attached to a host, and a virtual copy attached to a different host.

1. Create a new copy on the copy you wish to refresh from (the original copy is kept in the system).

2. The production volume’s SCSI personality moves to the new data container created in XtremIO memory.

3. The copy’s SCSI personality is moved to point to the other metadata created (based on the entity you want to refresh from).

4. The system either deletes or keeps the refreshed entity’s old data.

The above method enables a refresh operation of any entity in the virtual copy tree to any entity, without restrictions. The end result is that the SCSI personality is refreshed with data from the desired entity, without needing to perform mapping or needing to perform host side SCSI bus settings in order to access the data. This saves a lot of administration work and time.
Refresh Guidelines

Refresh guidelines are as follows:

- **Individual volumes**
  Any individual volume or virtual copy can be refreshed from any volume or virtual copy within the Volume Snapshot Group, without restrictions.

- **Consistency Groups**
  Copies created from a Consistency Group can be restored or refreshed in all combinations.

- **Snapshot Sets**
  Restore or refresh operations are only supported on Snapshot Sets generated from the same Consistency Group, or from other Snapshot Sets created from that Consistency Group.

- **Volume list**
  Restore or refresh operations are not supported on Snapshot Sets generated from Volume Lists.

When a refresh operation is initiated, a new Snapshot Set is created and then attached to the refreshed SCSI entities. The new Snapshot Set represents a new point-in-time that has been created in the system. This may introduce a challenge if you intend to have a fixed handler or referenced object intended for use in a refresh operation (for example, if you want to perform daily refreshes). The method for achieving this is using a Tag as a fixed handler for a refresh operation. Create a Tag on Snapshot Sets, and then use the Tag "Refresh" option. The refreshed SCSI entities change the Snapshot Set name. However, the Tag updates with the newly-created Snapshot Set, hence enabling a fixed handler.

Tags

XtremIO supports the use of Tags. Tags can be assigned to almost any type of object, such as a Consistency Group, a Snapshot Set, a Volume, an Initiator Group, a Scheduler, and more.

Using Tags on Snapshot Sets is highly recommended when working with a large number of objects, or when working with XtremIO Virtual Copies.
Use Cases

Offloading Backup Operations
Backup operations can typically consume a lot of environmental resources, when transferring data from application servers to backup or media servers, causing high network utilization, huge backup windows, and severe application impacts, all of which are problematic issues that are well-known to backup and storage administrators.

XtremIO Virtual Copies can be used for offloading the backup of production data into an external backup device. The backup can be taken from a copy instead of the production copy, thus offloading the process of moving the data to an external backup server.

Another solution that is enabled using XVCs is ProtectPoint, which leverages copies with RecoverPoint in order to directly backup data from XtremIO into DataDomain, in an efficient-aware manner, with deduplication capabilities.

Key Benefits
The key benefits of using XVCs for backup operations include:

- Immediate creation, update and mapping of XVCs using:
  - Instant refresh and/or update functionality to support agility of backup cycles.
  - Time saving on HBA rescans, useful on every backup or media server, as HBA rescanning tends to mount hundreds of devices.

- Space-efficient copies:
  - No physical or logical capacity is reserved or used.
  - Efficient logical memory consumption.

- Resource savings during backup cycles:
  - No production CPU resources are used, as there is no need to backup directly from the application server.
  - Less network resources, as the backup set is not transferred from the application server over the LAN.
Backup copies can be instantly updated or refreshed from production on a daily or weekly basis. No rescan on the HBA level is required on the backup or mount server. This approach saves a tremendous amount of time, since backup servers generally mount hundreds of devices within the organization.

**Creating an XVC for Backup Use - Behind the Scenes**

This section describes an example on how to create a backup, using XtremIO Virtual Copies, and what happens behind the scenes.

1. The production host is mapped to volumes from "Prod CG", as shown in Figure 14 (on page 30). The copy creation operation is initiated from "Prod CG" (done in order to create a copy for backup operations).

2. The newly-created Snapshot Set hosts the newly-created copies, which is then mapped to the backup host and mounted.

3. A Tag to the backup Snapshot Set is created and labeled "BackupCopyTag". This Tag is used during the refresh operation.

Figure 13: Offloading a Backup Operation –00
A backup operation is – at the least, in most customers’ environments – a daily task. This means that the backup copy needs to be refreshed on every cycle. XtremIO Virtual Copies natively support such requirements. Commands for doing so can be either run manually or easily scripted via the CLI, or by using the RESTful API. The CLI command to create an XVC for backup in this instance is:

```
create-snapshot-and-reassign from ProdCG to BackupCopyTag
```

Figure 15 shows the internal impact.
A new Snapshot Set is created, pointing to the same volumes (eliminating the necessity to rescan on the host level).

The new "SnapshotSet02" remains marked by the "BackupCopyTag" Tag, which enables running the same scripted command during the next cycle.

XtremIO automatically preserves "backup/undo" copies of the refreshed volumes, done in order to provide fast recovery in case of human error. This "backup/undo" copy is located in the old Snapshot Set in the "Snapshot Set01" example. This Snapshot Set can either be deleted manually or be deleted by argument, by using the "no-backup" command.

Testing and Development Use Cases

XtremIO Virtual Copies can be leveraged to provide Test and Dev copies of production data. Multiple master copies can be created, and each copy can be processed (such as an anonymization or sanitization process), to be prepared as a golden image for testing and development purposes. Multiple copies can then be created from each master copy, and presented to various development teams. Provisioning more copies is an easy and instantaneous process, and further copies of the provisioned copies can also be created.

XVC efficiency enables copies to be created, based on the demand for maximum business efficiency, rather than based on storage capacity or performance limitations.

Figure 16: Test and Dev Use Cases
Furthermore, XtremIO's easy refresh process enables an instant update of the development environments.

The tree layout in Figure 17 shows that the copy (snapshot) environment for "DevTeam1" is created. An instant copy of the production is created and then mapped to the DevTeam1 server.

The production environment suffers no impact from the operation, during which "SnapshotSet01" is created containing the copy of the volumes which are visible to the DevTeam1 server.

It is recommended to tag SnapshotSet01 to point to the newly-created development copy, using the "DevTeam1" Tag.

![Figure 17: Creating the Dev Environment for DevTeam1](image)

Test and Dev teams gain extremely high value from refresh operations. Updating a Dev copy from a master copy has never-before been easier, nor has it ever been so instantaneously achievable.

A single RESTful API call, or a single CLI command (or GUI wizard) enables the execution of the task.

The CLI command to create an XVC for Test and Dev in this instance is:

```bash
create-snapshot-and-reassign from ProdCG to DevTeam1
```
The Dev environment for team1 is instantly refreshed from the production copy. The operation also preserves all LUN information in order to eliminate rescanning HBA on the host level.

From a protection perspective, the "backup/undo" volumes are kept in case of human error, or in case performing a roll back is required by the user. The backup copy is always kept in the old Snapshot Set ("SnapshotSet01" in the above example). The backup copy can be removed if it is no longer required.

XtremIO enables having about 512 copies from a production volume, thus supporting the requirements of the most demanding customers.

The Devteam1 Tag enables easy repetition of this procedure, or its scripting for automatic usage.

Any other copy in the development life cycle, such as QA, Pre-Prod, Test, etc., can be also labeled by a Tag.

The refresh operation can be easily performed between Tags, as well as being performed from a Tag to a Consistency Group.

Figure 18: Creating Additional Dev Environment for DevTeam2
Key Benefits
The key benefits of using XVCs for testing and development operations include:

- Extreme agility to support the development life cycle.
- Space-efficient copies:
  - Resources are only appropriated for new writes.
  - No physical or logical capacity is reserved.
- Dozens of Test and Dev copies can be created, thus creating a high performance sandbox for every feasible engineer, and not a time-consuming operation.
- Quick refresh of test or development environments.
- Deduplication, compression and thin provisioning are always enabled.
- Copies of copies are supported.

Logical Data Protection
It is possible to create copies of the production volumes in order to protect against logical data corruption. Multiple copies can be created over a short interval in order to provide a fine RecoverPoint Objective (RPO) in a cyclic manner (a superior RPO to a backup).

For example, it is possible to create 48 copies every 30 minutes, and delete the latest copy. This provides an RPO of 30 minutes for the last day of production changes.

Figure 19: Multiple Point-in-Time Copies for Logical Data Protection

Flexible retention policies can be easily configured by using XtremIO's built-in Scheduler, from explicit intervals to daily and weekly retentions. The default backup copies are created as read-only (immutable).
This brings automation to the backup field and removes the need for external scripting. XVCs that are created using the built-in Scheduler are crash-consistent copies only. If application-aware copies are required, EMC's AppSync can provide scheduling capabilities with application-aware support.

Figure 20: Scheduler Configuration Window

The restore flow supports the Scheduler theme, enabling restore operations to be performed from "read-only" copies only (the refresh flow enables the use of any Snapshot Set as a source).

Figure 21: Restore Flow
**Offload Processing and Data Analytics**

XtremIO Virtual Copies can also be used to provide real-time analytics, thereby enabling users to perform the following actions:

- Offload processing to an external server.
- Extract, transform and load (ETL) processes to load data into a data warehouse.
- Achieve near real-time analytics for business intelligence (BI) reports.
- Consolidate online transaction processing (OLTP) and real-time, on a single platform.

*Figure 22: Offload Processing and Data Analytics*
**Key Benefits**

The key benefits of using XVCs for offload processing and data analytics include:

- Immediate updated copies of production without the need to make a brute force SAN copy
- Consolidating different workloads into a single all-flash scale-out platform, with consistent and predictable performance
- Accurate real-time BI analytics, based on the most updated copy of production data
- Space-efficient copies:
  - No physical or logical capacity is reserved or used in copies.
  - Deduplication and thin provisioning are always on.
  - Copy of copy always contains unique delta only.
- Offloading processing from the production server:
  - Free SAN BW/IOPS resources
  - Free CPU resources
  - FREE Network resources
- High performance on copies of production
Provision Bulk VMs by Using XVC on a Virtual Machine File System (VMFS)

Taking XVCs of a VMFS volume containing many virtual machines has proven to be the fastest method of cloning VMs. This method is much faster than any of the alternatives, such as the VAAI XCOPY method.

Figure 23 shows an example of a native clone created for a VM employing the use of XVC capabilities, thus shortening the operation time to a matter of seconds.

To make it easier for VMware administrators, EMC delivers a free plugin (VSI) for VMware Virtual Center.

The VSI plugin enables the integrated management of XtremIO from VC, while adding the following additional benefits:

- Integrated management and monitoring of XtremIO
- Automated restore/refresh operations on the VM level and DataStore level
- Applying XtremIO best practices to an ESXi infrastructure
- Supporting VDI deployments (for both Citrix and View)
- Fast provisioning of large scale VMs
- Space-efficient cloning of VMs
Conclusion

The XtremIO Virtual Copies feature offers a large number of high performance, low latency, read-only or writeable copies.

XtremIO Virtual Copies are efficient in metadata and physical space, can be created instantaneously, have no performance impact, and have the same data services as any other volume in the cluster (such as thin provisioning and Inline Data Reduction).

XVCs are easy to use and manage, and leverage a sophisticated metadata management engine that provides superior support for flash media, enabling high performance copying.

XVC technology introduces a key foundation for providing integrated copy data management capabilities, thus enabling customers to consolidate production and non-production tier-1 workloads on a scale-out platform with predictable, consistent performance.

XVCs can be used for test and development, backups, operation, data protection and near real-time analytics.