OpenVMS Technical
Journal V11

Host-Based Minimerge and Automatic Minicopy on Volume Processing in HP Volume Shadowing for OpenVMS

Akila Balasubramanian, OpenVMS Cluster Engineering

Overview

Volume Shadowing for OpenVMS guarantees that data is the same on all the members of a shadow set. The merge recovery operation is used to compare data on shadow set members to ensure that all of them are identical on every logical block. This recovery operation should be done as quickly as possible so that the shadow driver returns the identical data when read from any shadow set member at any time. Several enhancements have been introduced in the area of merge and copy operations. **Host Based Minimerge (HBMM)** improves merge operations by decreasing the number of comparisons needed to complete the merge operation. **Automatic Minicopy on Volume Processing (AMCVP)** helps the removed shadow set member to return to the shadow set using a minicopy operation instead of a full copy. Both of these enhancements allow significant gains in volume shadowing performance.

Overview of Merge Operations

Merge operations are required to avoid any discrepancy between the data (logical block number, or LBN) on shadow set members. During a merge operation, application I/O continues but at a slower rate.

Any of the following events can initiate a merge operation:

1. A system failure results in the possibility of incomplete application writes.

   When a write request to a shadow set fails from the system in which it is mounted, and the system fails before a completion status is returned to the application, the data might be inconsistent on the shadow set members:

   - All members might contain the new data.
   - All members might contain the old data.
   - Some members might contain new data and others might contain old data.

   The exact timing of the failure during the original write request determines the outcome. Volume Shadowing for OpenVMS reconciles the data using the MERGE operation.

2. A shadow set enters mount verification and then times out or aborts mount verification, under certain conditions.

   A shadow set that enters mount verification and either times out or aborts mount verification, will enter a merge state if the following conditions are true:

   - There are outstanding write I/O requests in the shadow driver's internal queues on the system or systems on which it has timed out.
   - The shadow set is mounted on other systems in the cluster.

3. The SET SHADOW/DEMAND_MERGE command is issued.

   This command is useful if:

   - The shadow set was created with INITIALIZE/SHAD command and without the /ERASE qualifier.

- You want to measure the impact of a minimerge or a full merge on I/O throughput.

Types of Merge Operations

A merge operation can be a full merge or minimerge. In a full merge operation, the members of a shadow set are compared with each other to ensure that they contain the same data. This is done by performing a block-by-block comparison of the entire volume and can be a very lengthy procedure.

In a minimerge operation, the merge is performed on only those areas of the shadow set where write activity occurred. This limitation avoids the need for the entire volume scan that is required by full merge operations, thus reducing consumption of system I/O resources.

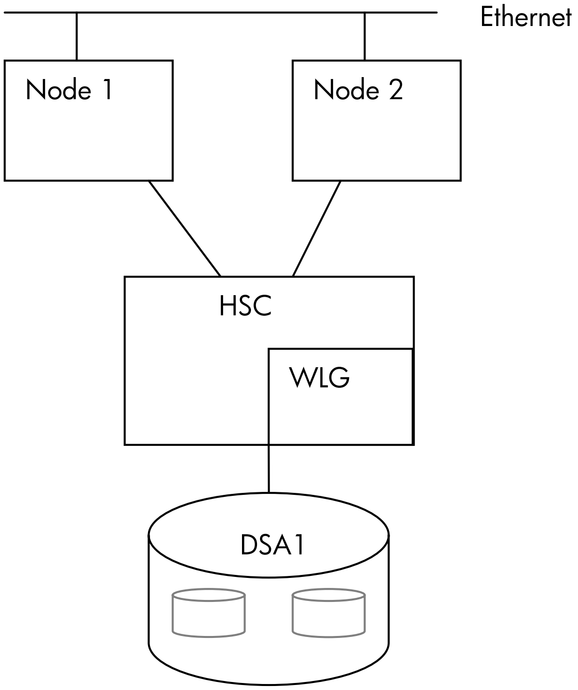
There are two types of minimerge that are based on the characteristic of the device:

- Controller based
- Host based (HBMM)

A controller-based minimerge is performed if the write-logging bit is set for the device in its UCB. Otherwise, a host-based minimerge is performed. If any of the members of the shadow set are capable of controller-based write logging (WLG), then HBMM is not allowed on the shadow set.

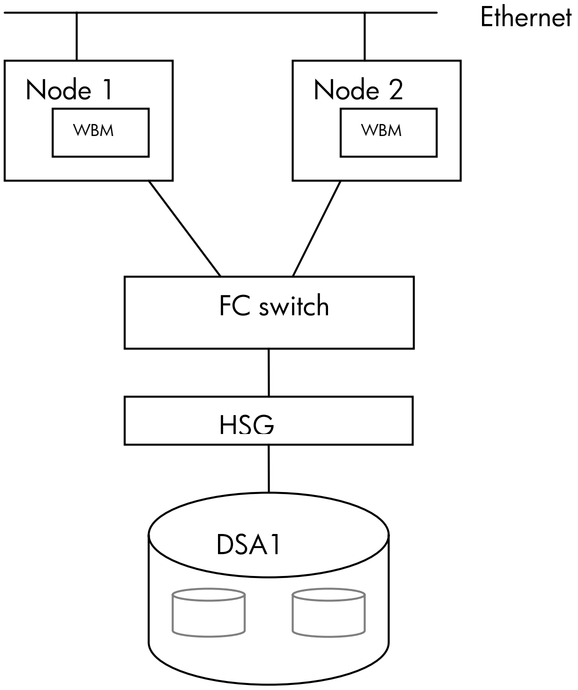
Controller-based minimerge is supported on HSC and HSJ controllers. As shown in Figure 1, by using information about write operations that were logged in controller memory, the minimerge is able to merge only those areas of the shadow set where write activity was known to have been in progress. The write logs contain information about exactly which LBNs in the shadow set had write I/O requests outstanding (from a failed node). The node that performs the minimerge operation uses the WLG to merge those LBNs that might be inconsistent across the shadow set. No controller-based write logs are maintained for a one-member shadow set or if the shadow set is mounted on only one system.

Figure 1. Controller-based minimerge



HBMM can be used with all disks that are supported by Volume Shadowing for OpenVMS except disks on HSJ, HSC, and HSD controllers. As shown in Figure 2, HBMM make use of write bitmaps (WBM) to track the writes that are occurring on the disk. The minimerge operation acts on only the LBNs marked in the bitmap. The next section discusses HBMM in detail.

Figure 2. Host-based minimerge



HBMM

Write bitmaps

HBMM depends on Write Bitmap (WBM) to perform the merge operation. For every LBN modified, the corresponding bit in the WBM is set. Each bit in the WBM corresponds to 127 blocks. The HBMM uses this WBM and merges only the LBNs set in the WBM. Bitmap entry is recorded before writing to disk.

On OpenVMS Alpha systems, bitmaps are created in S2 space. The memory required for the bitmap is calculated based on the size of the shadow set volume. For every gigabyte of storage of a shadow set mounted on a system, 2 KB of bitmap memory is required on that system for each bitmap.

Local and master bitmaps

Bitmaps can be local bitmaps or master bitmaps. A master WBM contains a record of all the writes to the shadow set from every node in the cluster that has the shadow set mounted. A minimerge operation can occur only on a system with a master bitmap.

A local WBM tracks all the writes that the local node issues to a shadow set. If a node with a local bitmap writes to the same LBN of a shadow set more than once, only the LBN of the first write is sent to the master WBM.

If the same block in a file is modified many times in a shadow set, the local bitmap helps to avoid the many messages to be sent to the master bitmap node to set the already-set bit in the master bitmap. This results in a noticeable performance improvement.

When a node that has the local bitmap and Virtual Unit VU mounted, modifies a LBN, then the "set bit" request is sent to the master bitmap node. The local bitmap bit is set only after the corresponding bit in the master bitmap is set.

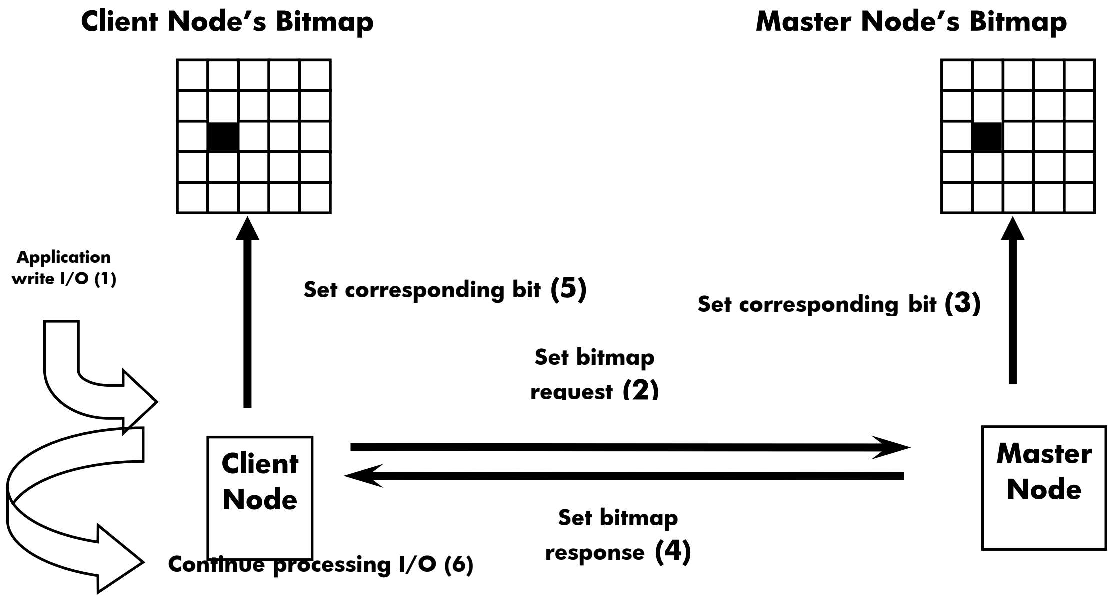
Multiple bitmaps

Existence, state, and master node of a bitmap are coordinated and communicated clusterwide via a per-bitmap lock and its value block. If only one master bitmap exists for the shadow set, and the system with the master bitmap fails or shut down, then the bitmap will not be there. Local bitmaps cannot be used for recovery. This results in a full merge operation. To avoid making the master bit map a single point of failure, HBMM provides the option of specifying more than one node on which master bitmaps are created. Multiple master bitmaps increase the likelihood of an HBMM operation rather than a full merge in the event of a system failure.

Single-message and buffered-message mode

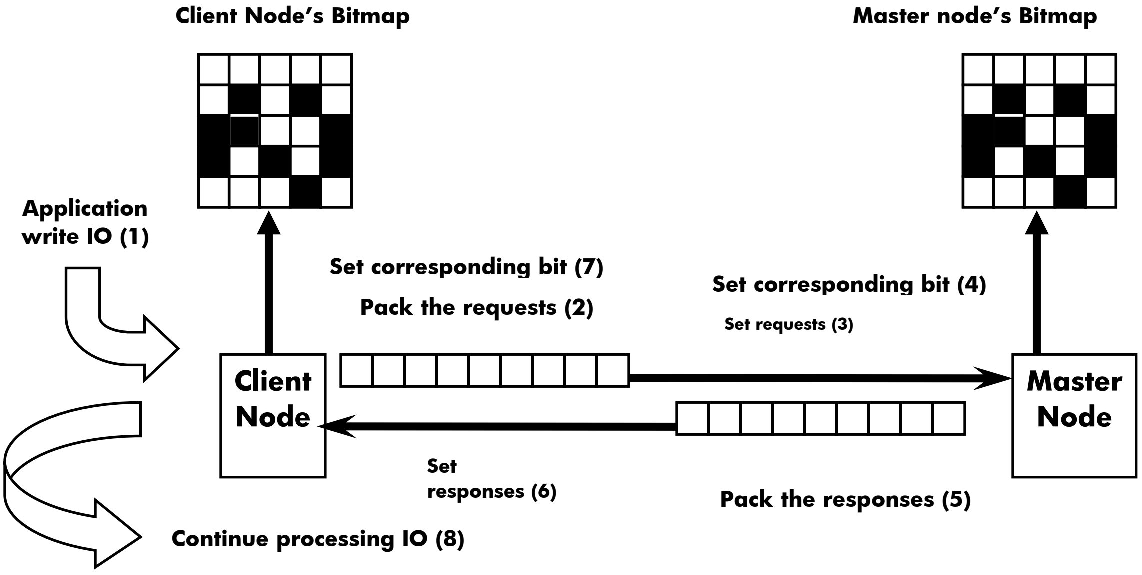
Node-to-node set-bit operations use System Communications Services (SCS)messages. [1] The SCS messages can be sent in single-message mode or buffered-message mode. In single-message mode, only one logical WBM write request per SCS message is sent to the master bitmap node. Figure 3 illustrates single-message mode. In buffered-message mode, shown in Figure 4, up to nine logical WBM write requests per SCS message can be sent. The messages are dispatched to the remote node when the message buffer is full or when the WBM internal timer expires. The WBM internal timer is calculated based on the WBM_MSG_INT system parameter, which specifies the maximum time a message waits before it is sent.

Figure 3. Set-bit Write request to master



[1] SCS – System Communication Services provides basic connection management and communication services, implemented as a logical path, between system applications on nodes in an OpenVMS Cluster system.

Figure 4. Buffered Set- Bit write request to master bitmap



HBMM policy

HBMM policies are defined to implement the decisions regarding master bitmaps. HBMM policy specifies the following attributes for one or more shadow sets:

- **MASTER_LIST** – Names of systems that are eligible to host a master bitmap.
- **COUNT** – Number of systems that will host a master bitmap (not to exceed six). If this number is omitted, the first six available systems of those you specified are selected.
- **RESET_THRESHOLD** – Threshold (in 512-byte blocks) at which the bitmaps are reset. If omitted, the threshold defaults are applied. In OpenVMS Version 8.3 and higher, the default is 1,00,000 blocks. For prior versions, the default is 50,000 blocks.
- **MULTIUSE** – Turns on AMCVP.

Writes that need to set bits in the bitmap are slightly slower than writes to areas that are already marked as having been written. Therefore, if many of the writes to a particular shadow set are concentrated in certain ''hot'' files, then the reset threshold should be made large enough so that the same bits are not constantly set and then cleared. On the other hand, if the reset threshold is too large, then the advantages of HBMM are reduced. For example, if 50% of the bitmap is populated (that is, 50% of the shadow set has been written to since the last reset), then the HBMM merge will take approximately 50% of the time of a full merge.

Using the SET SHADOW/POLICY command is the only method for specifying HBMM policies. You use the SET SHADOW/POLICY command with HBMM-specific qualifiers to define, assign, deassign, and delete policies and to enable and disable HBMM on a shadow set.

The following example shows the SET SHADOW/POLICY command and its output:

```
$ SET SHADOW DSA999:/POLICY=HBMM ( -
_$ (MASTER_LIST=(NODE1,NODE2,NODE3), COUNT=2), -
_$ (MASTER_LIST=(NODE4,NODE5), COUNT=1), -
_$ (MASTER_LIST=(NODE6,NODE7,NODE8), COUNT=2), -
_$ RESET_THRESHOLD=500000)

$ SHOW SHADOW DSA999:/POLICY=HBMM
HBMM Policy for device _DSA999:
HBMM Reset Threshold: 500000
HBMM Master lists:
Up to any 2 of the nodes: NODE1, NODE2, NODE3
Any 1 of the nodes: NODE4, NODE5
Up to any 2 of the nodes: NODE6, NODE7, NODE8
```

In this example:

- 5 master bitmaps will be present.
- The following nodes will host master bitmaps:
  Any 2 of NODE1, NODE2, and NODE3 (Site 1)
  Any 1 of NODE4, NODE5 (Site 2)
  Any 2 of NODE6, NODE7, NODE8 (Site 3) nodes.
- A threshold of 500000 blocks must be reached before clearing the bitmap.

HBMM policy attempts to maintain at least one HBMM master bitmap at each site in a multiple-site OpenVMS Cluster system, thereby minimizing the need to perform a full merge.

Some sites might find that a single HBMM policy can effectively implement the decisions. Other sites might need greater granularity and therefore implement multiple policies. The most likely need for multiple policies is when the cluster includes enough high-bandwidth systems that you want to ensure that the merge load is spread out. Multiple HBMM policies are also useful when shadow sets need different bitmap reset thresholds. The master list can be the same for each policy, but the threshold can differ.

Activating HBMM

HBMM is automatically activated on a shadow set under the following conditions:

- An HBMM policy exists for a given shadow set and that shadow set is then mounted on one or more systems defined in the master list.
- An HBMM policy is created for a mounted shadow set and at least one system that has it mounted is defined in the master list.

You can also activate HBMM with the SET SHADOW/ENABLE=HBMM command, provided a policy exists and the shadow set is mounted on a system defined in the master list of the shadow set policy, and the count has not been exceeded.

Hierarchy of Transient State Operations and Shadow Set Priority

A shadow set is in a **steady state** when none of the following operations is pending or active:

- Minimerge
- Minicopy
- Full copy
- Full merge

A shadow set is in a **transient state** if it has one or more of these operations pending, or one operation active. Although a combination of these transient states is valid, only one operation at a time can be performed.

Shadow set operations for a specific shadow set are performed in the following order:

1. Minimerge
2. Copy (either minicopy or full copy)
3. Full merge

You can assign a unique priority to every mounted shadow set on a per-system basis using the `SET SHADOW/PRIORITY=`*`n`* `DSA`*`n`* command. The priority assigned to a shadow set does not affect the hierarchy of transient state operations.

For example, if HBMM is not enabled after a device is added to a shadow set, it is marked as being in a full-copy transient state. If the system on which this shadow set is mounted fails, the shadow set is further marked as being in a full-merge state. In this case, the full copy operation is performed before the full merge is started.

A merge transient state is an event that cannot be predicted. The management of merge activity, on a specific system for multiple shadow sets, can be predicted if the priority level settings for the shadow sets differ.

In the following example, there are four shadow sets, and the SYSGEN parameter SHADOW_MAX_COPY on this system is equal to 1. A value of 1 means that only one merge or copy operation can occur at the same time. This example illustrates how the priority level is used to select shadow sets when only merge operations are involved.

Two shadow sets are assigned a priority level and two have the default priority level of 5000. The four shadow sets DSA1, DSA20, DSA22, and DSA42, are mounted on two systems. DSA20 and DSA42 are minimerge enabled.

```
$ SET SHADOW/PRIORITY=7000 DSA1:
$ SET SHADOW/PRIORITY=3000 DSA42:
 ! DSA20: and DSA22: are at the default priority level of 5000
```

In this example, when one system fails, all shadow sets are put into a merge-required state. The SHADOW_REC_DLY system parameter specifies the length of time a system waits before it attempts to manage recovery operations on shadow sets that are mounted on the system. After the delay resulting from recovery of this significant event, this system evaluates the shadow sets and the operations are performed in the following order:

1. A minimerge operation starts on DSA20, even though its priority of 5000 is lower than DSA1's priority of 7000. A minimerge operation always takes precedence over other operations. DSA20 and DSA42 are both minimerge enabled, but DSA20's higher priority causes its minimerge operation to start first.

2. A minimerge operation starts on DSA42. Its priority of 3000 is the lowest of all the shadow sets, but a minimerge operation takes precedence over other operations.

3. Because there are no other minimerge capable units, DSA1, with a priority level of 7000, is selected to start a merge operation, and it runs to completion.

4. A merge operation starts on DSA22, the one remaining shadow set whose priority is the default value of 5000, and runs to completion.

AMCVP – Multiuse bitmap

Automatic Minicopy on Volume Processing (AMCVP) means that an existing HBMM bitmap is made available to function as a minicopy bitmap. Specifically, the minimerge bitmap is made "Multiuse" bitmap. Before the introduction of automatic bitmap creation on volume processing, returning expelled members to a shadow set, after connectivity was restored, was a lengthy process. The expelled members could be returned only by undergoing a full copy. The availability of a multiuse bitmap enables the use of a minicopy operation, which takes considerably less time than a full copy operation. To enable AMCVP, you need to establish an HBMM policy for the shadow sets, and include the new MULTIUSE keyword in the policy.

The conversion of HBMM bitmap to minicopy bitmap happens automatically when connectivity to one or more shadow set members is lost and is not restored during the member's timeout period. When such connectivity is lost, the shadow set is paused for volume processing—that is, writes and reads are temporarily suspended until connectivity is restored or until the timeout period (established by the value of the SHADOW_MBR_TMO parameter) expires, whichever comes first.

If connectivity is not restored by the end of the timeout period, the lost members are removed from the shadow set, read and write I/O to the remaining member resumes, and the bitmap keeps track of the writes. The bitmap functions as a minicopy bitmap for the members that are removed. When the removed member is reinstated, then minicopy is performed.

While one or two members are expelled and after all members are restored to membership in the shadow set, the HBMM bitmap functionality remains in effect. The HBMM bitmap functionality is useful in the case of an expelled member only when the shadow set has three members and one member is expelled.

The following example shows how the HBMM bitmap is used as a multiuse bitmap and results in a minicopy operation. Shadow set DSA1: has two members $1$DKA100: (connected to NODE1, available to others via MSCP) and $2$DKB200: (connected to NODE2, available to others via MSCP). The shadow set policy for DSA1: is set to use HBMM bitmap as a multiuse bitmap, thereby enabling AMCVP. The master WBM is available on nodes NODE1 and NODE2.

```
$ SET SHADOW DSA1:/POLICY=HBMM ( -
_$ (MASTER_LIST= (NODEA), COUNT=1, MULTIUSE=1), -
_$ (MASTER_LIST= (NODEB), COUNT=1, MULTIUSE=1))
```

```
$ SHOW SHAD DSA1

_DSA1:    Volume Label: SHAD2
  Virtual Unit State:   Steady State
  Enhanced Shadowing Features in use:
        Dissimilar Device Shadowing (DDS)
        Host-Based Minimerge (HBMM)
        Automatic Minicopy (AMCVP)

  VU Timeout Value      3600    VU Site Value          0
  Copy/Merge Priority   5000    Mini Merge       Enabled
  Recovery Delay Per Served Member                    30
  Merge Delay Factor     200    Delay Threshold      200

  HBMM Policy
    HBMM Reset Threshold: 1000000
    HBMM Master lists:
      Any 1 of the nodes: NODE1 Multiuse: 1
      Any 1 of the nodes: NODE2 Multiuse: 1
    HBMM bitmaps are active on NODE2,NODE1
    Modified blocks since bitmap creation: 254

  Device $1$DKA100              Master Member
    Read Cost              2    Site 0
    Member Timeout       120

  Device $2$DKB200
    Read Cost            501    Site 0
    Member Timeout       120
```

The following command displays information such as the type of Bitmap, master write bitmap nodes, bitmap name, and so on:

```
$ SHOW DEVICE /BIT/FULL
Device          BitMap     Size      Percent     Type of   Master  Active  Creation
Cluster   Local Delete   Bitmap
 Name             ID      (Bytes)  Populated    Bitmap     Node           Date/Time
Size     Set  Pending  Name
DSA1:      0030007     17496      0.01%   Minimerge  NODE2    Yes  18-JUL-2007 02:08:22.34
127     0.01%    No    HBMC$DSA0001HBMM00010007
                 00050008     17496      0.01%   Minimerge  NODE1    Yes  18-JUL-2007
02:12:18.55  127     0.01%    No    HBMC$DSA0001HBMM00010005
```

The network cable to NODE2 was pulled off for a period of time greater than the value of the SHADOW_MBT_TMO parameter. Now connectivity to shadow set member $2$DKB200: (connected to NODE2) is lost by the shadow set. The connection cannot be restored until a time period equal to the value of the SHADOW_MBT_TMO parameter has elapsed. Hence, $2$DKB200: is removed from shadow set DSA1. At this point, the minimerge bitmap becomes a multiuse bitmap and keeps track of writes to the shadow set, as shown in the following example:

```
$ SHOW SHADOW DSA1/BIT/FULL
Device         BitMap     Size     Percent     Type of   Master  Active  Creation
Cluster   Local Delete   Bitmap
 Name           ID      (Bytes)  Populated    Bitmap     Node           Date/Time
Size     Set  Pending  Name
DSA1:         00050008     17496      0.01%   Multiuse   NODE1    Yes  18-JUL-2007
02:12:18.55  127     0.01%    No    HBMC$SHAD20000 18-JUL-2007 02:12:19.27
```

The network cable of NODE2 is now plugged in again. The node crashed (CLUEXITed) and then rebooted because of the long delay. When shadow set member $2$DKB200 is brought back to the shadow set, as shown in the following example, the minicopy takes place. In the preceding example, the bitmap for NODE2 disappeared because NODE2 was rebooted.

```
$ MOUNT/SYS DSA1:/SHAD=$2$DKB200: SHAD2
%MOUNT-I-MOUNTED, SHAD2 mounted on _DSA1:
%MOUNT-I-SHDWMEMCOPY,_$2$DKB200:(NODE2) added to the shadow set with a
copy operation
%MOUNT-I-ISAMBR, _$1$DKA100: (NODE1) is a member of the shadow set
```

```
$ SHOW DEVICE DSA1/BIT/FULL
Device         BitMap     Size     Percent     Type of   Master  Active  Creation
Cluster   Local Delete   Bitmap
 Name           ID      (Bytes)  Populated    Bitmap     Node           Date/Time
Size     Set  Pending  Name
DSA1:         00050008     17496      0.01%   Multiuse   NODE1    Yes  18-JUL-2007
02:12:18.55  127     0.01%    No    SHAD$_$2$DKB200:.0000  9-AUG-3685 22:40:30.34
```

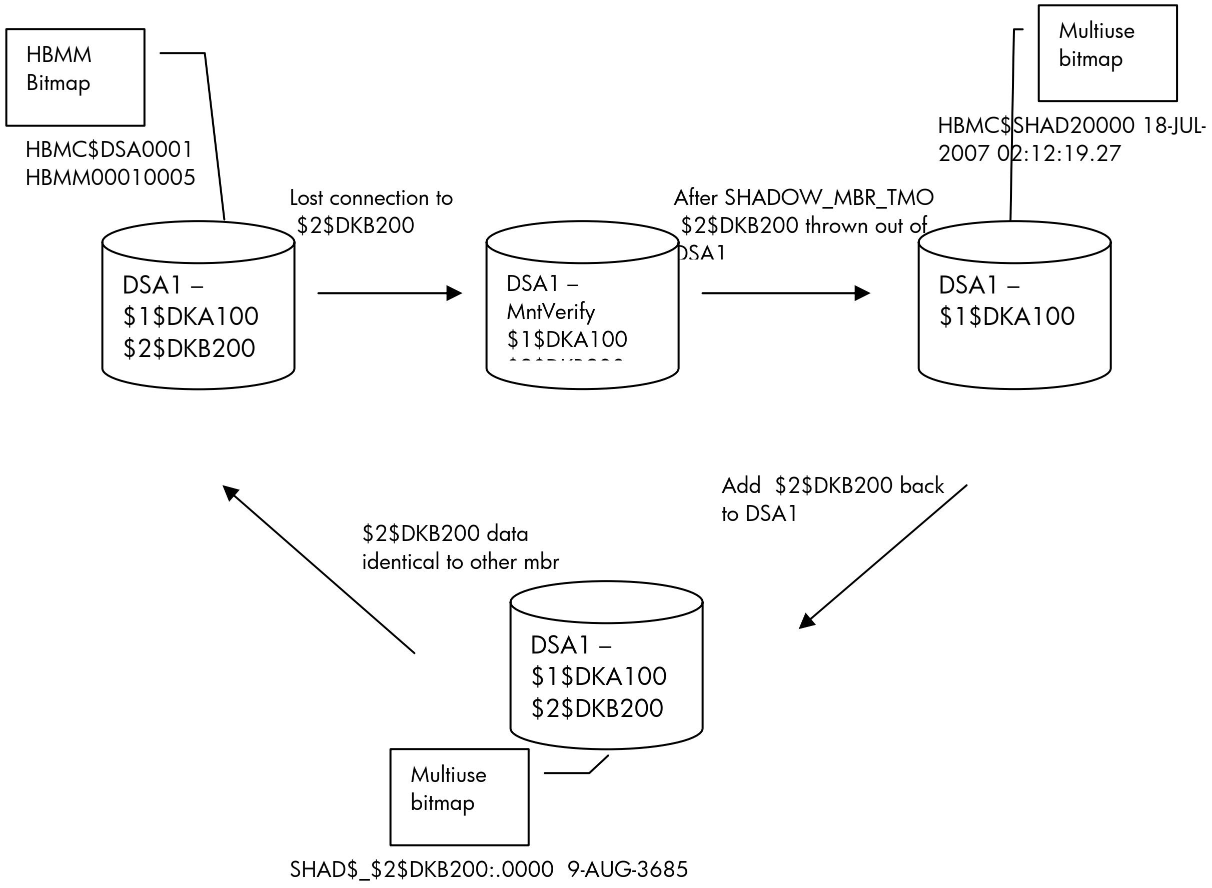
```
$ SHOW SHADOW DSA1
_DSA1:    Volume Label: SHAD2
  Virtual Unit State:   MiniCopy Active (90%) on NODE1
  Enhanced Shadowing Features in use:
        Dissimilar Device Shadowing (DDS)
        Host-Based Minimerge (HBMM)
        Automatic Minicopy (AMCVP)

  VU Timeout Value      3600    VU Site Value          0
  Copy/Merge Priority   5000    Mini Merge       Enabled
  Recovery Delay Per Served Member                    30
  Merge Delay Factor     200    Delay Threshold      200

  HBMM Policy
    HBMM Reset Threshold: 1000000
    HBMM Master lists:
      Any 1 of the nodes: NODE1 Multiuse: 1
      Any 1 of the nodes: NODE2 Multiuse: 1
    HBMM bitmaps are active on NODE1
    Modified blocks since bitmap creation: 254

  Device $1$DKA100              Master Member
    Read Cost              2    Site 0
    Member Timeout       120

  Device $2$DKB200              Copy Target (90%)
    Read Cost            501    Site 0
    Member Timeout       120
```

The multiuse bitmap now becomes the normal HBMM bitmap, as shown in the following example:

```
$ SHOW DEV DSA1/BIT/FULL
Device         BitMap     Size     Percent     Type of   Master  Active  Creation
Cluster   Local Delete   Bitmap
 Name           ID      (Bytes)  Populated    Bitmap     Node           Date/Time
Size     Set  Pending  Name
DSA1:         00050008     17496      0.01%   Minimerge  NODE1    Yes  18-JUL-2007
02:12:18.55  127     0.01%    No    HBMC$DSA0001HBMM00010005
```

Figure 5 summarizes the process that occurred in the preceding command examples.

Figure 5 Summary of steps involved in AMCVP



For more information

For more information about HBMM and AMCVP, see the following OpenVMS manuals:

- *HP OpenVMS Version 8.2 New Features and Documentation Overview*
- *HP OpenVMS Version 8.3 New Features and Documentation Overview*
- *HP Volume Shadowing for OpenVMS*