Symantec™ Storage Foundation and High Availability Solutions 6.1 SmartIO for Solid State Drives Solutions Guide - Linux

Symantec.

Symantec™ Storage Foundation Solutions SmartIO Solutions Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Product version: 6.1

Document version: 6.1 Rev 2

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization

- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information

- Upgrade assurance that delivers software upgrades

- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis

- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/contact_techsupp_static.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level

- Hardware information

- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec website.

https://sort.symantec.com/documents

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/
forums/storage-and-clustering-documentation

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Contents

Chapter 3 Administering SmartIO 42

Chapter 4 Troubleshooting and error handling 59

Appendix A Command reference 63

Index 65

Introducing SFHA Solutions SmartIO

This chapter includes the following topics:

- About SmartIO for solid-state drives
- About SmartIO in an SFHA environment

About SmartIO for solid-state drives

Solid-state drives (SSDs) are devices that do not have spinning disks. Today's solid-state technologies, such as DRAM and NAND flash, provide faster data access, are more efficient, and have a smaller footprint than traditional spinning disks. The data center uses solid-state technologies in many form factors: in-server, all flash arrays, all flash appliances, and mixed with traditional HDD arrays. Each form factor offers a different value proposition. SSDs also have many connectivity types: PCIe, FC, SATA, and SAS.

Due to the current cost per gigabyte of SSD devices, the best value of SSDs is not as high capacity storage devices. The benefit of adopting SSDs is to improve performance and reduce the cost per I/O per second (IOPS). Data efficiency and placement is critical to maximizing the returns on any data center's investment in solid state.

The SmartIO feature of Storage Foundation and High Availability Solutions (SFHA Solutions) enables data efficiency on your SSDs through I/O caching. Using SmartIO to improve efficiency, you can optimize the cost per IOPS. SmartIO does not require in-depth knowledge of the hardware technologies underneath. SmartIO uses advanced, customizable heuristics to determine what data to cache and how that data gets removed from the cache. The heuristics take advantage of SFHA Solutions' knowledge of the characteristics of the workload.

SmartIO uses a cache area on the target device or devices. The cache area is the storage space that SmartIO uses to store the cached data and the metadata about the cached data. The type of the cache area determines whether it supports VxFS caching or VxVM caching. To start using SmartIO, you can create a cache area with a single command, while the application is online.

When the application issues an I/O request, SmartIO checks to see if the I/O can be serviced from the cache. As applications access data from the underlying volumes or file systems, certain data is moved to the cache based on the internal heuristics. Subsequent I/Os are processed from the cache.

SmartIO supports read and write caching for the VxFS file systems that are mounted on VxVM volumes, in several caching modes and configurations. SmartIO also supports block-level read caching for applications running on VxVM volumes.

See "About SmartIO read caching for applications running on VxVM volumes" on page 11.

See "About SmartIO read caching for applications running on VxFS file systems" on page 17.

See "About SmartIO writeback caching for applications running on VxFS file systems " on page 24.

See "About SmartIO caching for Oracle databases on VxFS file systems " on page 30.

See "About SmartIO caching for databases on VxVM volumes" on page 38.

About SmartIO in an SFHA environment

In a clustered environment, the SmartIO cache is local to each node in the cluster. The cache area cannot be brought offline from one node and then brought online on a different node.

When SmartIO read caching is used in an Active/Passive environment such as with SFHA, you can deport and import the data volumes and file systems to another node. The SmartIO cache is not moved to the other node. Caching starts on the new node with the local SmartIO cache area there. SmartIO does not provide the capability to migrate the cache in a high availability cluster during a failover.

For VxFS writeback caching, failing over the disk group is not recommended. In the case of local mount, if there is no cache mirror, the disk copy of the file can be incomplete or stale if the node fails. Symantec recommends using SFCFSHA instead, which provides data reflection of the cache.

Using the SmartIO feature: use cases

This chapter includes the following topics:

- About SmartIO read caching for applications running on VxVM volumes

- About SmartIO read caching for applications running on VxFS file systems

- About SmartIO writeback caching for applications running on VxFS file systems

- About SmartIO caching for Oracle databases on VxFS file systems

- About SmartIO caching for databases on VxVM volumes

About SmartIO read caching for applications running on VxVM volumes

SmartIO supports block-level read caching for Veritas Volume Manager (VxVM) volumes. This type of SmartIO caching primarily supports the applications that run directly over raw volumes. For example, database instances running directly over raw volumes. Volume-level caching can also be used in cases where VxFS caching cannot be used. SmartIO only supports read caching at the volume level.

The SmartIO cache typically resides on one or more SSD devices or other fast devices. SmartIO accelerates the read I/O performance because the application read I/Os are serviced from the SSD-based cache rather than the standard storage.

SmartIO does not require complex configuration to set up caching. You simply set up a cache area, which is the storage space for the cached data and metadata about the cache. For volume-level read caching, the cache area has the VxVM type. A single VxVM cache area is used per system. By default, the SmartIO cache area enables automatic caching for all VxVM volumes on the system. If you prefer,

you can configure the cache area as noauto. For a noauto cache area, you must explicitly enable SmartIO read caching for the VxVM volumes. The configuration of the cache area is persistent.

See "Automatic caching for VxVM volumes" on page 13.

For each VxVM volume on which caching is enabled, SmartIO determines which data to cache or to evict from the cache. SmartIO uses its knowledge of the workload to optimize its use of the cache.

In release 6.1, the SmartIO feature supports only one VxVM cache area on a system. For each system, all VxVM volumes that are cached share a single cache area of VxVM type. Multiple VxVM cache areas are not supported, although the same system can have both a VxFS cache area and a VxVM cache area.

A cache area is private to each node in a cluster. The cache contents are not shared across the nodes in the cluster. A SmartIO VxVM type cache area only supports caching for I/Os from a single node. If a volume receives I/Os from multiple nodes, SmartIO disables caching for that volume on all nodes.

A SmartIO cache preserves cache coherency at the volume level. If the cache device becomes inaccessible while caching is enabled, the application continues to function normally. However, application performance may be reduced.

The data in the read cache is not persistent by default. In the case of a planned system reboot, you can choose to create a warm cache.

See "Support for a persistent or 'warm' VxVM cache" on page 59.

Required configuration for SmartIO read caching for VxVM volumes

You can set up SmartIO for read caching for VxVM volumes with the following configurations:

- A Storage Foundation Cluster File System High Availability (SFCFSHA) 6.1 cluster. A cache area cannot be on a shared volume. The VxVM cache area must be configured as local to each node, as shared access of cache area is not supported.
  The SmartIO VxVM caching is not supported with cluster file systems. Use VxFS caching for cluster file systems.
  See "About SmartIO read caching for applications running on VxFS file systems" on page 17.

- A Storage Foundation High Availability (SFHA) 6.1 cluster. The VxVM cache area must be configured as local to each node, as shared access of cache area is not supported.
  See "About SmartIO in an SFHA environment" on page 10.

- A stand-alone Storage Foundation 6.1 system.

The volumes to be cached must have the disk group version 190 or above.

The devices used for the cache area have the following characteristics:

- Utilize faster devices such as solid-state drives (SSDs) supported by Veritas Volume Manager (VxVM) to accelerate read IO performance. However, any devices supported by VxVM can be used for the cache area.

- Must be initialized for use with VxVM, and must have the `cdsdisk` format.

Automatic caching for VxVM volumes

The association type of a cache area indicates whether or not automatic caching is enabled for the system. The association type attribute for the VxVM cache area is persistent. The association type can be one of the following:

- `auto` attribute (default)

  The cache area is enabled for automatic caching. All VxVM data volumes on the system are cached unless you explicitly disable caching for that volume. You do not need to explicitly enable caching on a volume.

  SmartIO does not support caching RAID-5 volumes and DCO volumes. Also, SmartIO does not enable automatic caching for volumes used for logging and cache objects including Storage Replication Logs (SRLs), Data Change Maps (DCMs), and volumes used for space-optimized snapshot cache objects.

  By default, a VxVM cache area has the `auto` attribute.

- `noauto` attribute

  The cache area is not enabled for automatic caching. No volumes are cached automatically. You must explicitly enable caching for each volume that you want cached. You do not need to explicitly disable a volume for caching, except to exclude a volume that was previously enabled. You can enable caching when you create a volume. You can also selectively enable or disable read caching on an existing VxVM volume, without quisceing the I/O.

Setting up SmartIO read caching for VxVM volumes

In read mode, the SmartIO feature caches the VxVM I/Os. To set up SmartIO for read caching for a VxVM volume, simply create the cache area.

Setting up SmartIO read caching for VxVM volumes

1 Create a VxVM type cache area on an SSD device, using one of the following commands:

   - Specify one or more devices using the disk access name (*daname*) of the device. The device should be initialized for VxVM use and have the `cdsdisk` format.

```
# sfcache create -t VxVM [size] daname[...] \
[cacheline_size=cacheline_size] [--auto|--noauto] [cachearea_name]
```

Where:

daname specifies the disk access name of the device on which to create the cache area.

cachearea_name specifies a customized name for the cache area. If unspecified, the SmartIO feature automatically generates a name for the cache area.

size specifies the size for the cache area. By default, the cache area uses all of the available space on the device.

`--noauto|--auto` specifies the type of cache area. The default is `--auto`.

cacheline_size specifies the unit that SmartIO uses for caching. When the application I/O accesses the data, the SmartIO moves the data to the cache according to the cacheline size. Generally, you do not need to change the *cacheline_size*.

For example:

```
# sfcache create -t VxVM ssd0_0
```

- Alternatively, you can specify the name of a disk group and a volume to use for the cache area. In this case, you cannot specify a size: the cache area uses the entire volume.

```
# sfcache create -t VxVM  [cacheline_size=cacheline_size] \
 [--noauto|--auto] dg/vol
```

Where:

dg/vol specifies the disk group name and volume name that you want to use for the cache area. SmartIO uses this name for the cache area.

`--noauto|--auto` specifies the association type of the cache area. The default is `--auto`.

For example:

```
# sfcache create -t VxVM --auto ssd_dg/ssd_vol
```

2   Enable caching for the required VxVM volumes. If the cache area is auto, this step is not required. Caching is enabled by default when the SSD device comes online, for any VxVM volumes.

If the cache area is `noauto`, you must enable caching for the volumes to be cached.

```
# sfcache enable dg/vol [--read]
```

Where:

dg/*vol* specifies the disk group name and volume name of the volume that you want to cache.

For example:

```
# sfcache enable mydg/vol1
```

Verifying the VxVM cache area and monitoring the caching

After the SmartIO feature is configured, you can verify that the cache area is present and that caching is occurring.

To verify and monitor the cache area

1   Use the following command to display information about the cache areas on
the system.

```
# sfcache list -l

Cachearea: sfcachearea_1
Assoc Type: AUTO
Type: VxVM
Size: 30.00g
Cacheline Size: 64.00k
Memory Size: 16.00m
State: ONLINE

ASSOCIATED DATA OBJECTS:

Volume: testdg/testvol1
Size: 500.00g
State: ENABLED
Kstate: ENABLED
Caching Mode: read

Volume: testdg/testvol2
Size: 500.00g
State: ENABLED
Kstate: ENABLED
Caching Mode: read
```

See "Displaying information about a cache area" on page 45.

2 Use the following command to display information about a specific cache area.

```
# sfcache list sfcachearea_1
Cachearea: sfcachearea_1
Assoc Type: AUTO
Type: VxVM
Size: 30.00g
Cacheline Size: 64.00k
Memory Size: 16.00m
State: ONLINE


ASSOCIATED DATA OBJECTS:


ASSOC DATAOBJECT NAME    CACHING-MODE   STATE     KSTATE
testdg/testvol1          read           ENABLED   ENABLED
testdg/testvol2          read           ENABLED   ENABLED
```

See "Viewing the SmartIO cache statistics " on page 55.

3 To see statistics on the cache usage, use the following command:

```
# sfcache stat sfcachearea_1

                     HIT RATIO    ART(Hit)ms   ART(Miss)ms   BYTES
NAME                 %CACHE RD    WR    RD    WR    RD    WR    RD      WR
TYPE: VxVM
sfcachearea_1        13.43  91.24 94.20 0.142 0.819 0.414 0.798 15.31g  4.21g

ASSOCIATED DATA OBJECTS:
testdg/testvol1      6.10   90.00 96.00 0.141 0.459 0.348 0.448 6.77g   1.89g
testdg/testvol2      7.32   91.00 92.00 0.143 1.179 0.480 1.149 8.54g   2.31g
```

About SmartIO read caching for applications running on VxFS file systems

Storage Foundation High Availability Solutions supports read caching on solid-state drives (SSDs) for applications running on VxFS file systems. In this scenario, application reads are satisfied from the cache whenever possible. As the application accesses the file system, the file system loads data from the disk into the cache. Application writes go to the disk in the usual way. With each write, the file system synchronizes the cache to ensure that applications never see stale data. If a cache device fails, a file that is cached in read mode is completely present on the disk.

Therefore, the cache failure does not affect the application I/Os for the file and the application I/Os continue without interruption.

Required configuration for SmartIO read caching for a VxFS file system

You can set up SmartIO for read caching for a VxFS file system with the following configurations:

- A Storage Foundation Cluster File System High Availability (SFCFSHA) cluster, or a Storage Foundation High Availability (SFHA) cluster, with private SSDs in each node.
  See "About SmartIO in an SFHA environment" on page 10.

- A standalone Storage Foundation system.

The file system must have the following characteristics:

- The file system layout version 10.

- Must be mounted on a VxVM volume.

Automatic caching for VxFS file systems

In release 6.1, the SmartIO feature supports only one VxFS cache area on a system. For each system, all VxFS file systems that are cached share a single cache area of VxFS type.

The association type of a cache area indicates whether or not automatic caching is enabled for the system. The association type attribute for the VxFS cache area is persistent. The association type can be one of the following:

- `auto` attribute (default)
  The cache area is enabled for automatic caching. All file systems on the system are cached unless you explicitly disable caching for that file system. You do not need to explicitly enable caching on a file system. By default, a VxFS cache area has the `auto` attribute.

- `noauto` attribute
  The cache area is not enabled for automatic caching. No file systems are cached automatically. You must explicitly enable caching for each file system that you want cached. You do not need to explicitly disable a file system for caching, except to exclude a file system that was previously enabled.

A cache area is private to each node in a cluster. For a cluster file system, each of the nodes in the cluster has its own cache area. Caching occurs on a per-node

basis and the cache contents are not shared across the nodes in the cluster. A file system with caching enabled is associated with the local cache area on each node.

Setting up SmartIO read caching for VxFS file systems

In read mode, the SmartIO feature caches the VxFS file system read I/Os. To set up SmartIO for read caching for a VxFS file system, simply create the cache area.

Setting up SmartIO read caching for VxFS file systems

1 Create the VxFS cache area on the SSD device, using one of the following commands.

   ■ Specify one or more devices using the disk access name (*daname*) of the device. The device should be initialized for VxVM use and have the cdsdisk format.

   ```
   # sfcache create [-t VxFS] [size] daname[...] [--auto|--noauto] \
    [cachearea_name]
   ```

   Where:
   daname specifies the disk access name of the device on which to create the cache area.
   cachearea_name specifies a customized name for the cache area. If unspecified, the SmartIO feature automatically generates a name for the cache area.
   size specifies the size for the cache area. By default, the cache area uses all of the available space on the device.
   --noauto|--auto specifies the type of cache area. The default is --auto.
   For example:

   ```
   # sfcache create ssd0_0
   ```

   ■ Alternatively, you can specify the name of a disk group and a volume to use for the cache area. In this case, you cannot specify a size: the cache area uses the entire volume.

   ```
   # sfcache create [-t VxFS] [--noauto|--auto] dg/vol
   ```

   Where:
   dg/vol specifies the disk group name and volume name that you want to use for the cache area. SmartIO uses this name for the cache area.
   --noauto|--auto specifies the type of cache area. The default is --auto.
   For example:

```
# sfcache create --auto ssd_dg/ssd_vol
```

2   If the file system is not already mounted, mount the VxFS file system.

   ■ If the cache area is auto, read caching is enabled when you mount the
     VxFS file system.
     For example:

     ```
     # mount -t vxfs /dev/vx/dsk/testdg/vol1 /mnt1
     ```

   ■ If the cache area is noauto, you must enable caching for each VxFS file
     system that you want to cache. To enable caching, mount the file system
     with the -o smartiomode option.
     For example:

     ```
     # mount -t vxfs -o smartiomode=read /dev/vx/dsk/testdg/vol1 /mnt1
     ```

     You can also enable caching after the file system is mounted.

     ```
     # sfcache enable mount_point
     ```

     Where:
     mount_point is the mount point for the file system.
     For example:

     ```
     # sfcache enable /mnt1
     ```

3   If required, you can further customize the caching behavior.

   See "Customizing the caching behavior" on page 22.

Verifying the VxFS cache area and monitoring the caching

After the SmartIO feature is configured, you can verify that the cache area is present
and that caching is occurring.

For a VxFS cache area, the sfcache list command shows the caching mode for
the file or directory. If the mode is not explicitly set, the file or directory inherits the
caching mode of the mount point. If the mode is explicitly set for a file or directory,
that value persists across remounts. The displayed caching mode may differ from
the mode that is enabled for the mount point. The writeback mode is not enabled
unless the file system is mounted in writeback mode. If a file or directory is set to
writeback mode, but the file system is mounted in another mode, the file or directory
inherits the caching mode of the mount point.

To verify and monitor the cache area

1   Use the following command to display information about the cache areas on
    the system.

    ```
    # sfcache list

    NAME            TYPE SIZE  ASSOC-TYPE  STATE   DEVICE
    sfcachearea_2 VxFS 31.97g AUTO         ONLINE  ssd0_0
    ```

2   Use the following command to display information about a specific cache area.

    ```
    # sfcache list sfcachearea_2
    Cachearea: sfcachearea_2
    Assoc Type: AUTO
    Type: VxFS
    Size: 9.96g
    State: ONLINE

    /dev/vx/dsk/sfcache_defaultdg/sfcachearea_2:
    FSUUID                              SIZE    MODE  MOUNTPOINT
    3671ff51-9696-0000-872c-000056bcb408 39.0 MB   read  /mnt1
    ```

3   To display information about a specific file system:

```
# sfcache list /mnt1
```

```
/mnt1:
READ CACHE      WRITEBACK      MODE      PINNED   NAME
    39.0 MB         0 KB       read      yes      /mnt1/dir
    39.0 MB         0 KB       read      yes      /mnt1
```

4   To see statistics on the cache usage, use the following command:

```
# sfcache stat /mnt1
```

```
Cache Size:    9.97 GB
Cache Utilization:    39.0 MB ( 0.38 %)


Read Cache
Hit Ratio     Data Read     Data Written


/mnt2:
   0.00 %         0 KB          39.0 MB
```

The output displays statistics for the cached data.

Customizing the caching behavior

By default, SmartIO caches the file data based on the workload. SmartIO loads portions of files into the cache based on I/O access. When the cache area fills, data may be evicted to make room for caching new data. SmartIO uses criteria such as frequency of access to evict data. While the data is in the cache, the subsequent I/Os to that file data are satisfied from the cache. If the data is evicted, any subsequent I/O request is served from the primary storage. SmartIO may then cache the data again.

To maximize the use of the cache, you can customize the caching behavior to control when files are loaded or evicted from the cache. You can customize the caching behavior, using the following operations:

- The `load` operation preloads files into the cache before the I/O accesses the files. The files are already in the cache so that the I/Os return more quickly. By default, the files are loaded in the background. Use the `-o sync` operation to load the files synchronously, which specifies that the command does not return until all the files are loaded. The files that are loaded in this way are subject to the usual eviction criteria.

- The `pin` operation prevents the files from being evicted from the cache. You can pin commonly used files so that SmartIO does not evict the files and later need to cache the files again. A pinned file is kept in the cache indefinitely, until it is deleted or explicitly unpinned. If you pin a file with the `-o load` option, the operation also caches the file contents synchronously. If you do not specify the `-o load` option, the file contents are cached based on I/O access.

- The `unpin` operation removes files from the pinned state. The `unpin` operation does not cause the file to be immediately evicted. SmartIO considers the file for eviction in the same way as any other file, when space is required in the cache.

For each of these operations, you can specify files individually, or specify a directory name to affect all of the files in a directory. Use the `-r` option to make the selection recursive.

To load a file or directory

◆ To load a file or directory to the cache, specify the file name or the directory name to the following command.

```
# sfcache load [-r] [-o sync] {file|dir}
```

Use the `-r` option to make the selection recursive.

Use the `-o sync` option to specify that the command does not return until all the files are loaded.

To pin a file or directory

◆ To pin a file or directory to the cache, specify the file name or the directory name to the following command.

```
# sfcache pin [-o load] [-r] {file|dir}
```

Use the `-r` option to make the selection recursive.

Use the `-o load` option to load the file synchronously into the cache.

To unpin a file or directory

◆ To unpin a file or directory to the cache, specify the file name or the directory name to the following command.

```
# sfcache unpin [-r] {file|dir}
```

Use the `-r` option to make the selection recursive.

About SmartIO writeback caching for applications running on VxFS file systems

Storage Foundation and High Availability Solutions supports writeback caching on solid-state drives (SSDs) for applications running on Veritas File System (VxFS) file systems. In this scenario, application reads and writes are satisfied from the cache whenever possible.

SmartIO provides write caching in the writeback mode. In writeback mode, an application write returns success after the data is written to the SmartIO cache, which is usually on an SSD. At a later time, SmartIO flushes the cache, which writes the dirty data to the disk. Writeback caching expects to improve the latencies of synchronous user data writes. Write order fidelity is not guaranteed while flushing the dirty data to the disk.

Writeback caching is superset of read caching. When writeback caching is enabled, read caching is implicitly enabled. Reads are satisfied from the cache if possible, and the file system transparently loads file data into the cache. Both read and writeback caching may be enabled for the same file at the same time.

The writeback caching mode gives good performance for writes, but also means that the disk copy may not always be up to date. If a cache device fails, a file that is cached in writeback mode may not be completely present on the disk. SmartIO has a mechanism to flush the data from the cache device when the device comes back online. Storage Foundation Cluster File System High Availability (CFS) provides additional protection from data loss with cache reflection.

In the case of CFS, when writeback caching is enabled, SmartIO mirrors the writeback data at the file system level to the other node's SSD cache. This behavior, called cache reflection, prevents loss of writeback data if a node fails. If a node fails, the other node flushes the mirrored dirty data of the lost node as part of reconfiguration. Cache reflection ensures that writeback data is not lost even if a node fails with pending dirty data.

In the case of local mount, if there is no cache mirror, the disk copy of the file can be incomplete or stale if the node fails.

After writeback caching is enabled on the mount point, the qualified synchronous writes in that file system are cached. SmartIO determines if a write qualifies for writeback caching, using criteria such as the following:

- The write request must be PAGESIZE aligned (multiple of 4k).

- The write request is not greater than 2MB.

- The file on which the writes are happening is not mmapped.

- The writeback caching is not explicitly disabled by the administrator.

You can also customize which data is cached, by adding advisory information to assist the SmartIO feature in making those determinations.

Required configuration for SmartIO writeback caching for a VxFS file system

You must have an Enterprise license to use SmartIO with writeback caching for Storage Foundation or Storage Foundation High Availability.

You can set up SmartIO for writeback caching for a VxFS file system with the following configurations:

- A Storage Foundation Cluster File System High Availability (SFCFSHA) cluster with exactly 2 nodes. Writeback caching is not enabled if the cluster has more than 2 nodes. If another node is added while writeback caching is configured, writeback caching is disabled. Caching continues in read mode.
  If the cluster file system is unmounted on one of the two nodes while writeback caching is configured, then writeback caching is disabled. Caching continues in read mode. If the cluster file system on the second node is remounted, then writeback caching is enabled automatically.

- Local mount configuration.

In the case of CFS, writeback caching uses LLT transport to mirror the writeback data. Application writes that are cached are also written to the remote cache before the write is returned. Symantec recommends that you configure LLT over RDMA to avoid impact to the throughput.

For information on configuring LLT, see the *Symantec Storage Foundation Cluster File System High Availability Installation Guide*.

The file system must have the following characteristics:

- The file system has disk layout Version 10.

- Must be mounted on a VxVM volume.

Setting up SmartIO writeback caching for VxFS file systems

In writeback mode, the SmartIO feature caches the VxFS file system read and write I/Os. To set up SmartIO for writeback caching for a VxFS file system, create the cache area and mount the file system in writeback mode.

Setting up SmartIO writeback caching for VxFS file systems

1   Create the VxFS cache area on the SSD device, using one of the following commands.

- Specify one or more devices using the disk access name (*daname*) of the device. The device should be initialized for VxVM use and have the `cdsdisk` format.

```
# sfcache create [-t VxFS] [size] daname[...] [--auto|--noauto] \
 [cachearea_name]
```

Where:
daname specifies the disk access name of the device on which to create the cache area.
cachearea_name specifies a customized name for the cache area. If unspecified, the SmartIO feature automatically generates a name for the cache area.
size specifies the size for the cache area. By default, the cache area uses all of the available space on the device.
`--noauto|--auto` specifies the type of cache area. The default is `--auto`.
For example:

```
# sfcache create ssd0_0
```

- Alternatively, you can specify the name of a disk group and a volume to use for the cache area. In this case, you cannot specify a size: the cache area uses the entire volume.

```
# sfcache create [-t VxFS] [--noauto|--auto] dg/vol
```

Where:
`--noauto|--auto` specifies the type of cache area. The default is `--auto`.
dg/vol specifies the disk group name and volume name that you want to use for the cache area. SmartIO uses this name for the cache area.
For example:

```
# sfcache create --auto ssd_dg/ssd_vol
```

2   Mount the VxFS file system and set the `smartiomode` option to writeback. If
    the file system is already mounted, you must remount the file system to set
    the `smartiomode` option to `writeback`.

    Example of local mount:

```
# mount -t vxfs -o smartiomode=writeback /dev/vx/dsk/testdg/vol1 \
 /mnt1
```

    Example of remount:

```
# mount -t vxfs -o remount,smartiomode=writeback \
/dev/vx/dsk/testdg/vol1 /mnt1
```

    For a cluster file system, the file system must be mounted on both the nodes
    with the `smartiomode` and `cluster` options set.

    When you mount the CFS file system with these options, SmartIO automatically
    reflects the cache on the other node's SSD cache.

    Example of CFS mount:

```
# mount -t vxfs -o cluster,smartiomode=writeback \
/dev/vx/dsk/testdg/vol1 /mnt1
```

    To enable writeback caching, the `smartiomode` option must be set to `writeback`
    regardless of whether the cache area is `auto` or `noauto`. If the cache area is
    `auto` and the `smartiomode` is not set, SmartIO caching is enabled in the default
    read mode for the file system.

3   If required, you can further customize the caching behavior.

    See "Customizing the caching behavior" on page 22.

    See "Tuning the writeback caching" on page 53.

Verifying the VxFS cache area and monitoring the caching (writeback mode)

After the SmartIO feature is configured, you can verify that the cache area is present
and that caching is occurring.

For a VxFS cache area, the `sfcache list` command shows the caching mode for
the file or directory. If the mode is not explicitly set, the file or directory inherits the
caching mode of the mount point. If the mode is explicitly set for a file or directory,
that value persists across remounts. The displayed caching mode may differ from

the mode that is enabled for the mount point. The writeback mode is not enabled unless the file system is mounted in writeback mode. If a file or directory is set to writeback mode, but the file system is mounted in another mode, the file or directory inherits the caching mode of the mount point.

To verify and monitor the cache area

1 To display information about the cache areas on the system.

   # **sfcache list**

   For example, a single-node VxFS cache area displays output as follows:

   ```
   NAME            TYPE SIZE  ASSOC-TYPE  STATE   DEVICE
   sfcachearea_2 VxFS 9.96g AUTO        ONLINE  ssd0_0
   ```

   For a cluster file system with writeback enabled, if you issue the sfcache list command just after you mount with writeback enabled (and before any read caching has happened), the output shows the space used in cache as 1.0 GB. Cache reflection is configured, with a local log and a remote log that each have a log size of 512 MB.

2 To display information about a specific cache area:

   # **sfcache list sfcachearea_2**

```
Cachearea: sfcachearea_2
Assoc Type: AUTO
Type: VxFS
Size: 9.96g
State: ONLINE

/dev/vx/dsk/cachedg/ssdvol:
FSUUID                                              SIZE  MODE       MOUNTPOINT
5dfe4a52805b0b003c7600002cde6d2ce70300005dfe4a52   551.0 MB  writeback  /mnt1
5efe4a52eb76000041760000a0dec33fe70300005efe4a52    39.0 MB  read       /mnt2
5efe4a524a86050046760000b485af64e70300005efe4a52   551.0 MB  writeback  /mnt3
```

The output displays information about the cache configuration and attributes.

See "Displaying information about a cache area" on page 45.

3 To display information about a specific file system:

```
# sfcache list /mnt1

/mnt1:
READ CACHE     WRITEBACK     MODE       PINNED  NAME
    39.0 MB         0 KB     writeback  yes     /mnt1/dir
    39.0 MB         0 KB     writeback  yes     /mnt1
```

4 To see statistics on the cache usage, use the following command:

```
# sfcache stat sfcachearea_2

TYPE: VxFS
NAME: sfcachearea_2
                Cache Size:     9.97 GB
         Cache Utilization:    1.133 GB (11.37 %)
 File Systems Using Cache:        3
Writeback Cache Use Limit:        8 GB
  Writeback Flush Timelag:       10 s

Read Cache                              Writeback
Hit Ratio   Data Read  Data Written   Hit Ratio   Data Written

Total:
   0.00 %       0 KB       195.0 MB   100.00 %        78.0 MB

/mnt1:
   0.00 %       0 KB        78.0 MB   100.00 %        39.0 MB

/mnt2:
   0.00 %       0 KB        39.0 MB     0.00 %           0 KB

/mnt3:
   0.00 %       0 KB        78.0 MB   100.00 %        39.0 MB
```

The output displays statistics for the cached data.

See "Viewing the SmartIO cache statistics " on page 55.

5   To see statistics on cache usage for a particular file system, use the following command:

```
# sfcache stat /mnt1

Cache Size:     9.97 GB
Cache Utilization:   551.0 MB ( 5.40 %)

Read Cache                                Writeback
Hit Ratio    Data Read    Data Written    Hit Ratio    Data Written

/mnt1:
   0.00 %         0 KB        78.0 MB      100.00 %        39.0 MB
```

6   Check the `syslog` to verify whether writeback caching is enabled.

You should see a line such as the following in the `syslog`:

```
vxfs: msgcnt 4 writeback caching is enabled for /dev/vx/dsk/sharedg2/sharevol2
```

If writeback caching is disabled for a particular file system, you would see a line such as the following in the `syslog`:

```
vxfs: msgcnt 9 writeback caching is disabled for /dev/vx/dsk/sharedg2/sharevol2
```

About SmartIO caching for Oracle databases on VxFS file systems

SmartIO provides application templates to optimize caching for databases running on VxFS file systems. SmartIO uses the templates to apply policies to particular types of information in the database. For example, index files may have different caching policies from data files.

SmartIO provides the following application template for VxFS file systems:

■   Template name: oracle
Optimizes caching for Oracle databases running on VxFS file systems.

Prerequisites and configuration for using the SmartIO plug-in for Oracle

The SmartIO feature of Storage Foundation and High Availability Solutions includes a plug-in for Oracle databases. The `oracle` plug-in for the `sfcache` command line requires root privileges for some operations. The intended user of the SmartIO

`oracle` plug-in is the Oracle database administrator (DBA). Before running the SmartIO `sfcache` command line with the `oracle` plug-in, the system administrator has to provide SUDO authorization to the Oracle DBA.

Make sure the system is configured correctly to use the SmartIO plug-in for Oracle.

To configure the system to use the SmartIO plug-in for Oracle

1   To allow DBA to manage cache policies, the system administrator has to provide SUDO authorization to the Oracle DBA. To configure SUDO authorization, modify the file `/etc/sudoers`. Add the following line:

```
oracle  ALL=(ALL)        NOPASSWD: /opt/VRTS/bin/fscache
```

2   Before executing the Oracle sfcache plug-in, create the file system cache area and bring the cache area online.

3   The Oracle sfcache plug-in needs to query database catalog tables. Make sure the Oracle database is online and running on the same host where you need to run the `sfcache` commands for the Oracle plug-in.

Setting default SmartIO caching policies for a database running on a VxFS file system

SmartIO provides application templates to optimize caching for databases running on VxFS file systems. SmartIO uses the templates to apply policies to particular types of files in the database. For example, caching policies for index files and data files.

The oracle template sets the default policy for the Oracle database, as follows:

■   Turns off caching (nocache mode) for ARCHLOG files

■   Sets read caching (read mode) for TEMPFILEs

■   For OLTP database, sets read caching to the datafiles with the most frequent reads.
    For OLAP database, sets read caching to all datafiles that contain INDEXes.

To set the default SmartIO caching policies for a database

◆ To set the default SmartIO caching policies for a database, run the following command as ORACLE user:

```
# sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
   -H $ORACLE_HOME -o setdefaults --type={OLTP | OLAP}
```

Where:

$ORACLE_HOME and $ORACLE_SID are mandatory and uniquely identify the database.

OLAP or OLTP indicates the type of application load. OLAP, or Online Analytical Processing, applications process workloads intended for multi-dimensional analytical queries. OLTP, or Online Transaction Processing, applications process transaction-oriented workloads, such as for data entry and retrieval transaction processing.

Example of an OLTP database:

```
$ sfcache app  cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o setdefaults --type=oltp

INFO: Oracle Instance tpcc is runnig
INFO: Store DB details at /tpccdata/.CACHE_INFO
INFO: Setting oltp policies
INFO: Setting nocache mode to /tpccdata

INFO: Setting nocache mode to /tpcclog

INFO: Setting nocache mode to /tpcclog/log_1_1
INFO: Setting nocache mode to /tpcclog/log_1_2
INFO: Setting nocache mode to /tpcclog/log_1_3
INFO: Setting nocache mode to /tpcclog/log_1_4

INFO: Setting nocache mode to /home/oracle/app/oracle/product/11.2.0
/dbhome_1/dbs/arch
INFO: Setting read mode to /tpccdata/temp_0_0
INFO: Setting read mode to /tpccdata/temp_0_2
INFO: Setting read mode to /tpccdata/temp_0_1
INFO: Setting read mode to /tpccdata/temp_0_4

AWR snapid range min=1 max=7

INFO: Setting read mode to /tpccdata/stok_0_25
INFO: Setting read mode to /tpccdata/stok_0_24
INFO: Setting read mode to /tpccdata/stok_0_20
```

```
INFO: Setting read mode to /tpccdata/stok_0_29
INFO: Setting read mode to /tpccdata/stok_0_23
INFO: Setting read mode to /tpccdata/stok_0_22
INFO: Setting read mode to /tpccdata/cust_0_5
```

Example of an OLAP database:

```
$ sfcache app  cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o setdefaults --type=olap
INFO: Oracle Instance tpcc is runnig
INFO: Store DB details at /tpccdata/.CACHE_INFO
INFO: Setting olap policies
INFO: Setting nocache mode to /tpccdata

INFO: Setting nocache mode to /tpcclog

INFO: Setting nocache mode to /tpcclog/log_1_1
INFO: Setting nocache mode to /tpcclog/log_1_2
INFO: Setting nocache mode to /tpcclog/log_1_3
INFO: Setting nocache mode to /tpcclog/log_1_4

INFO: Setting nocache mode to /home/oracle/app/oracle/product/11.2.0
/dbhome_1/dbs/arch
INFO: Setting read mode to /tpccdata/temp_0_0
INFO: Setting read mode to /tpccdata/temp_0_2
INFO: Setting read mode to /tpccdata/temp_0_1
INFO: Setting read mode to /tpccdata/temp_0_4

INFO: Setting read mode to /tpccdata/icust2_0_30
INFO: Setting read mode to /tpccdata/ordr_0_32
INFO: Setting read mode to /tpccdata/iordr2_0_44
INFO: Setting read mode to /tpccdata/iordr2_0_29
INFO: Setting read mode to /tpccdata/iordr2_0_47
INFO: Setting read mode to /tpccdata/icust2_0_49
INFO: Setting read mode to /tpccdata/icust1_0_2
INFO: Setting read mode to /tpccdata/istok_0_1
INFO: Setting read mode to /tpccdata/ordr_0_33
INFO: Setting read mode to /tpccdata/ordr_0_37
INFO: Setting read mode to /tpccdata/iordr2_0_37
```

Setting SmartIO caching policies for database objects

A Database Administrator (DBA) with knowledge of the database activity and usage statistics may want to adjust the SmartIO caching policies based on this information. You can set a SmartIO caching policy for a specified database object, including a named tablespace, a recent partition, or a particular datafile. You can also pin a specified database object to hold it in the SmartIO cache area.

To set the caching policy for a specified database object

◆ Use the following command:

```
# sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o set --cachemode={nocache|read} \
{--datafile=name | --partition=name \
| --tablespace=name | --tablecluster=name \
| --filelist=name}
```

Pinning and unpinning database objects

Use this procedure to pin or unpin the specified database object, including a named tablespace, table-cluster, partition or datafile.

To pin or unpin the specified database object

◆ To pin or unpin the specified database object, including a named tablespace, table-cluster, partition or datafile, use the following command:

```
# sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o {pin | unpin} {--datafile=name \
| --partition=name | --tablespace=name}
```

Enabling and disabling caching for the database

The SmartIO plug-in for oracle allows you to enable or disable caching for the database.

You can use this operation to temporarilly disable caching for database jobs like backup or data warehouse ETL (extract, transform and load) operations. After the job completes, you can enable caching. You can enable and disable caching while the database is online.

To enable caching for the database

◆ Use the following command:

```
# sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
        -H $ORACLE_HOME -o enable
```

For example:

```
$ sfcache app  cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
        -H $ORACLE_HOME -o enable
INFO: Oracle Instance tpcc is runnig
INFO: Store DB details at /tpccdata/.CACHE_INFO
INFO: Setting enable mode to /tpccdata

INFO: Setting enable mode to /tpcclog
```

To disable caching for the database

◆ Use the following command:

```
# sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
        -H $ORACLE_HOME -o disable
```

For example:

```
$ sfcache app  cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
        -H $ORACLE_HOME -o disable
INFO: Oracle Instance tpcc is runnig
INFO: Store DB details at /tpccdata/.CACHE_INFO
INFO: Setting disable mode to /tpccdata

INFO: Setting disable mode to /tpcclog
```

Listing cache policy details for the database

Use this procedure to list caching policies for the specified database object, including a datafile, partition or tablespace.

To list caching policies for the database

◆ Use the following command:

```
# sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o list {--datafile=name \
| --partition=name | --tablespace=name}
```

For example, to list the caching policies for the tablespace stok_0:

```
$ sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o list --tablespace=stok_0
INFO: Oracle Instance tpcc is runnig
INFO: Store DB details at /tpccdata/.CACHE_INFO
FILENAME                          MODE     PINNED    CACHE_USED
--------                          ----     ------    ----------
/tpccdata/stok_0_0                read     no          1.05 GB
/tpccdata/stok_0_2                read     no         1.046 GB
/tpccdata/stok_0_1                read     no         1.055 GB
/tpccdata/stok_0_4                read     no          1.03 GB
/tpccdata/stok_0_3                read     no         1.043 GB
/tpccdata/stok_0_5                read     no         1.055 GB
/tpccdata/stok_0_6                read     no         1.044 GB
/tpccdata/stok_0_8                read     no         1.054 GB
/tpccdata/stok_0_7                read     no         1.048 GB
/tpccdata/stok_0_9                read     no          1.03 GB
/tpccdata/stok_0_10               read     no         1.029 GB
/tpccdata/stok_0_12               read     no          1.05 GB
/tpccdata/stok_0_11               read     no         1.045 GB
```

Listing cache statistics for the database

A DBA can monitor the cache usage and hit ratio at a database level. You can use the statistics to evaluate the current cache policies. If the hit ratio is not satisfactory, you can consult Oracle AWR reports and change the policies.

To list cache statistics for a database

◆ To view cache statistics for all of the mountpoints of the database, use the following command:

```
$ sfcache app cachearea=cachedg/ssdvol oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o stat

INFO: Oracle Instance tpcc is running
INFO: Store DB details at /tpccdata/.CACHE_INFO
                VxFS SmartIO Statistics
              --------- ----------------------
Cache Size = 1.096 TB
 Cache Utilization = 64.3 GB ( 5.73 %)
Mount        Hit Ratio Cached Pinned  Read      Written   Data
                       Files  Files   Bytes     Bytes     Pinned
-----        --------- -----  ------  --------  --------  ------
/tpccdata    67.80 %   445    10      134.4 GB  179.0 GB  160 KB
/tpcclog     38.10 %   9      0       4 KB      8 KB      0 KB
```

About SmartIO caching for databases on VxVM volumes

SmartIO provides application templates to optimize caching for databases running on VxVM volumes. SmartIO uses the templates to apply policies to particular types of volumes in the database. For example, index volumes may have different caching policies from data volumes.

SmartIO provides the following application templates for VxVM volumes:

■ Template name: oracle
Optimizes caching for Oracle databases running on VxVM volumes.

■ Template name: sybase
Optimizes caching for Sybase databases running on VxVM volumes.

Applying a SmartIO database caching template for a VxVM volume

SmartIO provides application templates to optimize caching for databases running on VxVM volumes. SmartIO uses the templates to apply policies to particular types of volumes in the database. For example, caching policies for index volumes and data volumes.

To apply a SmartIO sybase caching template for a VxVM volume

1 Log in as root user.

2 Apply a SmartIO sybase caching template for a VxVM volume using the following command:

```
# sfcache app cachearea=cachedg/ssdvol sybase \
 {olap|oltp} SYBASE_HOME db_username db_server [db_name]
```

Where:

SYBASE_HOME indicates the path of Sybase home.

olap or oltp indicates the type of application load. OLAP, or Online Analytical Processing, applications process workloads intended for multi-dimensional analytical queries. OLTP, or Online Transaction Processing, applications process transaction-oriented workloads, such as for data entry and retrieval transaction processing.

db_user_name indicates the Database User Name.

db_server indicates the Database Server Name.

db_name indicates the database name. The database name is optional. If database name is not specified, then the command runs for all databases under the user *db_user_name*.

For example:

```
# sfcache app cachearea=cachedg/ssdvol sybase olap sa VXVMSYBS vmdb
```

3 Enter the password for the database user at the prompt.

4 If required, you can view the command activity in the caching log file.

The log file is /etc/vx/log/sfcache.log.

For example:

```
Enabling caching for sybdg/DB1DATA
sfcache enable sybdg/DB1DATA
Disabling caching for sybdg/DB1LOG
Sfcache disable sybdg/DB1LOG
```

To apply a SmartIO oracle caching template for a VxVM volume

1   Log in as root user.

2   Apply a SmartIO oracle caching template for a VxVM volume using the following command:

```
# sfcache app cachearea=cachedg/ssdvol oracle \
 {olap|oltp} oracle_user_name ORACLE_HOME \
 ORACLE_SID [ASM_HOME [ASM_SID]]
```

Where:

olap or oltp indicates the type of application load. OLAP, or Online Analytical Processing, applications process workloads intended for multi-dimensional analytical queries. OLTP, or Online Transaction Processing, applications process transaction-oriented workloads, such as for data entry and retrieval transaction processing.

oracle_user_name indicates the user name of the Oracle user.

ORACLE_HOME indicates the directory where the Oracle software is installed. Usually the ORACLE_HOME value is stored in the `bash.rc` file or the `profile` file of the Oracle user.

ORACLE_SID indicates the System ID used to uniquely identify the database.

ASM_HOME indicates the directory where the ASM software is installed.

ASM_SID indicates the System ID for the ASM instance. By default, the value is `+ASM`. For multiple instances of ASM, the value may differ.

The following examples show the variations of the app command for the Oracle database.

The first example shows the Oracle database created directly on VxVM volumes. In this scenario, the ASM_HOME and ASM_SID values are not applicable.

Example of Oracle on Raw volumes:

```
# sfcache app cachearea=sfcachearea_1 oracle olap oracle /ora_base/db_home rawdb
```

The next example shows the Oracle ASM database created on VxVM volumes. In this scenario, you must specify the ASM_HOME. If required, specify the ASM_SID.

Example of Oracle ASM:

```
# sfcache app cachearea=sfcachearea_1 oracle oltp oracle /orabin/dbbase/dbhome
\ testdb /orabin/gridhome
```

3 Enter the password for the database user at the prompt.

4 If required, you can view the command activity in the caching log file.

The log file is `/etc/vx/log/sfcache.log`.

Example log file for Oracle on Raw volumes:

```
Fri Jun  7 22:04:31 IST 2013 sfcache app cachearea=sfcachearea_1
oracle olap oracle /ora_base/db_home rawdb
Enabling caching for rawdg/rawvol02
/usr/sbin/vxprint -v -g rawdg -e 'v_name="rawvol02"'
/usr/sbin/vxprint -g rawdg -F %cache_area_type rawvol02
/usr/sbin/vxprint -g rawdg -F %iscachevol rawvol02
/usr/sbin/vxprint -g rawdg -F %caching rawvol02
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
Enabling caching for rawdg/rawvol06
/usr/sbin/vxprint -v -g rawdg -e 'v_name="rawvol06"'
/usr/sbin/vxprint -g rawdg -F %cache_area_type rawvol06
/usr/sbin/vxprint -g rawdg -F %iscachevol rawvol06
/usr/sbin/vxprint -g rawdg -F %caching rawvol06
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
Disabling caching for rawdg/rawvol01
/usr/sbin/vxprint -v -g rawdg -e 'v_name="rawvol01"'
/usr/sbin/vxprint -g rawdg -F %cache_area_type rawvol01
/usr/sbin/vxprint -g rawdg -F %iscachevol rawvol01
/usr/sbin/vxprint -g rawdg -F %caching rawvol01
```

Example log file for Oracle ASM:

```
Enabling caching for testdg/testvol
/usr/sbin/vxprint -v -g testdg -e 'v_name="testvol"'
/usr/sbin/vxprint -g testdg -F %cache_area_type testvol
/usr/sbin/vxprint -g testdg -F %iscachevol testvol
/usr/sbin/vxprint -g testdg -F %caching testvol
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
Enabling caching for testdg/testvol2
/usr/sbin/vxprint -v -g testdg -e 'v_name="testvol2"'
/usr/sbin/vxprint -g testdg -F %cache_area_type testvol2
/usr/sbin/vxprint -g testdg -F %iscachevol testvol2
/usr/sbin/vxprint -g testdg -F %caching testvol2
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
```

Administering SmartIO

This chapter includes the following topics:

Creating a cache area

SmartIO introduces the concept of a cache area. The cache area is the storage space that SmartIO uses to store the cached data and the metadata about the cached data. You create a cache area to use for I/O caching. Usually, you use SSD devices or other fast devices for the cache area. A cache area can be used for VxFS caching or VxVM caching. You can create only one cache area of each type on each system.

To create a cache area on a device, specify the device name (disk access name) or the name of a disk group and volume on the device.

By default, a cache area has an association type of auto. All of the data objects of the same type (VxVM or VxFS) are implicitly associated with the auto cache area. If the association type is noauto, you must explicitly associate the data objects to the cache area.

In a cluster, each node has a separate, locally-mounted cache area.

To create a VxVM cache area

◆ Create the VxVM cache area on the SSD device, using one of the following commands:

- Specify one or more devices using the disk access name (*daname*) of the device. The device should be initialized for VxVM use and have the `cdsdisk` format.

```
# sfcache create -t VxVM [size] daname[...] \
[cacheline_size=cacheline_size] [--auto|--noauto] [cachearea_name]
```

Where:
daname specifies the disk access name of the device on which to create the cache area.
cachearea_name specifies a customized name for the cache area. If unspecified, the SmartIO feature automatically generates a name for the cache area.
size specifies the size for the cache area. By default, the cache area uses all of the available space on the device.
`--noauto|--auto` specifies the type of cache area. The default is `--auto`.
cacheline_size specifies the unit that SmartIO uses for caching. When the application I/O accesses the data, the SmartIO moves the data to the cache according to the cacheline size. Generally, you do not need to change the *cacheline_size*.
For example:

```
# sfcache create -t VxVM ssd0_0
```

- Alternatively, you can specify the name of a disk group and a volume to use for the cache area. In this case, you cannot specify a size: the cache area uses the entire volume.

```
# sfcache create -t VxVM  [cacheline_size=cacheline_size] \
 [--noauto|--auto] dg/vol
```

Where:

dg/*vol* specifies the disk group name and volume name that you want to use for the cache area. SmartIO uses this name for the cache area.

`--noauto|--auto` specifies the association type of the cache area. The default is `--auto`.

For example:

```
# sfcache create -t VxVM --auto ssd_dg/ssd_vol
```

To create a VxFS cache area

◆   Create the VxFS cache area on the SSD device, using one of the following commands.

    ■   Specify one or more devices using the disk access name (*daname*) of the device. The device should be initialized for VxVM use and have the `cdsdisk` format.

```
# sfcache create [-t VxFS] [size] daname[...] [--auto|--noauto] \
 [cachearea_name]
```

    Where:

    daname specifies the disk access name of the device on which to create the cache area.

    cachearea_name specifies a customized name for the cache area. If unspecified, the SmartIO feature automatically generates a name for the cache area.

    size specifies the size for the cache area. By default, the cache area uses all of the available space on the device.

    `--noauto|--auto` specifies the type of cache area. The default is `--auto`.

    For example:

```
# sfcache create ssd0_0
```

    ■   Alternatively, you can specify the name of a disk group and a volume to use for the cache area. In this case, you cannot specify a size: the cache area uses the entire volume.

```
# sfcache create [-t VxFS] [--noauto|--auto] dg/vol
```

    Where:

    dg/*vol* specifies the disk group name and volume name that you want to use for the cache area. SmartIO uses this name for the cache area.

    `--noauto|--auto` specifies the type of cache area. The default is `--auto`.

    For example:

```
# sfcache create --auto ssd_dg/ssd_vol
```

Displaying information about a cache area

SmartIO uses a cache area on the SSD devices for caching. Each cache area has attributes such as the cache type, size, association type, the caching state, and the devices used for caching. The cache area also has associated data objects.

For a VxFS cache area, the sfcache list command shows the caching mode for the file or directory. If the mode is not explicitly set, the file or directory inherits the caching mode of the mount point. If the mode is explicitly set for a file or directory, that value persists across remounts. The displayed caching mode may differ from the mode that is enabled for the mount point. The writeback mode is not enabled unless the file system is mounted in writeback mode. If a file or directory is set to writeback mode, but the file system is mounted in another mode, the file or directory inherits the caching mode of the mount point.

To display information about a cache area

1   Use the following command to display information about the cache areas on
    the system.

    # **sfcache list**

    ```
    NAME           TYPE SIZE   ASSOC-TYPE STATE   DEVICE
    sfcachearea_1 VxVM 9.91g  AUTO       ONLINE ssd0_2809
    sfcachearea_2 VxFS 31.97g AUTO       ONLINE ssd0_0
    ```

2   Use the following command to display information about a specific cache area.

    Example of a VxVM cache area:

    # **sfcache list sfcachearea_1**
    ```
    Cachearea: sfcachearea_1
    Assoc Type: AUTO
    Type: VxVM
    Size: 30.00g
    Cacheline Size: 64.00k
    Memory Size: 16.00m
    State: ONLINE


    ASSOCIATED DATA OBJECTS:


    ASSOC DATAOBJECT NAME    CACHING-MODE  STATE     KSTATE
    testdg/testvol1          read          ENABLED   ENABLED
    testdg/testvol2          read          ENABLED   ENABLED
    ```

    Example of a VxFS cache area:

    # **sfcache list sfcachearea_2**
    ```
    Cachearea: sfcachearea_2
    Assoc Type: AUTO
    Type: VxFS
    Size: 31.97g
    State: ONLINE


    /dev/vx/dsk/sfcache_defaultdg/sfcachearea_2:
    FSUUID                                SIZE      MODE    MOUNTPOINT
    3671ff51-9696-0000-872c-000056bcb408  4 KB      read    /mnt1
    ```

Enabling or disabling caching for a data object

With the SmartIO feature, you can disable or enable caching for a volume or a mount point.

If automatic caching is configured, you do not need to explicitly enable caching for any data objects. The SmartIO feature caches all of the data objects of the same type as the cache area (VxVM or VxFS). The data objects have the caching state of `auto`, by default. If required, you can disable caching for a particular data object. If the system has an `auto` cache area, the SmartIO feature caches any objects that have the `auto` state or the `enabled` state.

If automatic caching is not configured, you must explicitly enable caching for any data objects that you want the SmartIO feature to cache. If the system has a no-auto cache area, the SmartIO feature only caches the objects that have the `enabled` state. The SmartIO feature does not cache data objects with the caching state of `auto`, which is the default caching state.

See "Enabling or disabling caching for a file system" on page 47.

See "Enabling or disabling caching for a data volume " on page 48.

Enabling or disabling caching for a file system

The SmartIO feature enables you to disable or enable caching for a specific file system. For a cluster file system, you must run the commands on each node in the cluster.

To enable caching for a file system

◆ To enable caching for a file system, use the following command.

sfcache enable *mount_point*

Where:

mount_point is the mount point for the file system.

To disable caching for a file system

◆ To disable caching for a file system, use the following command. Use the `-o` `purge` option to clear the existing cache data from the cache. By default, the data is not purged.

sfcache disable [-o purge] *mount_point*

Where:

mount_point is the mount point for the file system.

To purge the caching data for a file system

◆ To purge the caching data for a file system, use the following command. Use the purge option to clear the existing cache data from the cache.

```
# sfcache purge {mount_point|fsuuid}
```

Where:

mount_point is the mount point for the file system.

fsuuid specifies the UUID of the file system. Specify the *fsuuid* to purge the caching data for a file system that is not mounted.

Enabling or disabling caching for a data volume

SmartIO enables you to disable or enable caching for a specific data volume.

To enable caching for a specific data volume

◆ To enable caching for a specific data volume, use the following command.

```
# sfcache enable [--auto] dgname/volname
```

To disable caching for a specific data volume

◆ To disable caching for a specific data volume, use the following command.

```
# sfcache disable dgname/volname
```

Adding a device to the cache area

You can add a device to the cache area when you increase the size of the cache area. This operation can be done online.

To resize a cache area

◆ To resize a cache area, use the following command. Specify a new device by the disk access name to expand the cache area onto that device. The maxsize option increases the cache area to the maximum size supported by the devices in the cache area.

```
# sfcache resize [daname ...] {newsize | maxsize} cachearea_name
```

Pausing caching from a volume to a cache area

You may want to temporarily stop caching to a cache area for a particular volume, without removing the association between the volume and the cache. You can pause the caching and resume it later. This option only applies to the VxVM type of cache areas. When caching is paused, no new data is cached on reads. The contents of the cache are updated if writes occur to a region that is already cached.

To pause caching

◆   To pause, use the following command.

```
# sfcache set dgname/volname --pause
```

To resume caching

◆   To resume caching for the volume, use the following command.

```
# sfcache set dgname/volname --resume
```

Removing a device from the cache area

If the cache area spans more than one device, you can remove one of the devices from the cache area.

To remove a device from a cache area

1   Make sure that the cache will fit on the remaining devices after you remove the device. If necessary, resize the cache to a size smaller than the size of the remaining devices.

```
# sfcache resize newsize cachearea_name
```

For example, suppose the current cache area sfcachearea_1 has a maximum size of 5G. You want to remove an SSD ssd0_0 that is 2G. First, resize the cache area to 2.5G:

```
# sfcache resize 2.5g sfcachearea_1
```

2   To remove a device from a cache area, specify the device by the disk access name to remove the cache area.

```
# sfcache rmdev [daname ...]
```

For example, to remove the SSD ssd0_0:

```
# sfcache rmdev ssd0_0
```

3   Optionally, you can use the maxsize option to resize the cache area to the maximum size that fits on the remaining devices in the cache area.

```
# sfcache resize maxsize cachearea_name
```

For example:

```
# sfcache resize maxsize sfcachearea_1
```

Destroying a cache area

You can destroy a cache area, which removes all data from the cache. Before you destroy the cache area, you must bring the cache area offline so that no caching is taking place.

To destroy the cache area

1   To bring the cache area offline:

      # **sfcache offline** *cachearea_name*

2   To destroy the cache area, use the following command.

      # **sfcache delete** *cachearea_name*

Setting the attributes of the VxVM cache area

For a VxVM cache area, you can set the following attributes:

■ `memsz`. The amount of system RAM that the SmartIO solution requires to keep VxVM caching-related metadata. When the VxVM cache area is created, SmartIO calculates the best possible value. You do not need to change this value, unless the system is low on system RAM.

■ association type. The association type is either `--auto` or `--noauto`. The association type determines whether caching is automatically enabled for volumes on the system.
See "Automatic caching for VxVM volumes" on page 13.

To set the memory size

◆ To set the memory size, use the following command.

      # **sfcache set {*dg/vol*|*cachearea_name*} memsz=*size***

Where

dg/*vol* specifies the disk group and volume that is used for the cache area.

cachearea_name specifies the name of the cache area

size specifies the maximum size for the memory for in-core metadata.

For example:

      # **sfcache set myspecialcache memsz=128m**

To set the association type

◆ To set the association type, use the following command.

```
# sfcache set {--auto|--noauto} {dg/vol|cachearea_name}
```

Where

dg/vol specifies the disk group and volume that is used for the cache area.

cachearea_name specifies the name of the cache area

For example:

```
# sfcache set --noauto mydg/myvol
```

Setting or changing the caching mode

For a VxFS cache area, the caching mode determines what kind of caching is performed for the specified mount point. The mode can be `nocache`, `read`, or `writeback`. The default mode is `read`.

A VxVM cache area only supports read mode.

You cannot change the type (VxVM or VxFS) of a cache area. You must destroy the cache area and create a new cache area of the required type.

You can set the caching mode of a VxFS mount point with the `-o smartiomode` option. The caching mode that is set with the mount command represents the highest level of caching that can be enabled for objects on the mount point. If you specify nocache mode, the SmartIO caching is disabled for the mount point. You cannot enable SmartIO caching for any data objects in that mount point. You must remount the file system to enable caching.

Similarly, if you specify read mode during the mount, you cannot enable SmartIO writeback caching for any data objects in that mount point.

To set the caching mode of a VxFS mount point

◆ To set the caching mode when you mount the VxFS file system, use the following command:

```
# mount -t vxfs -o smartiomode=[mode] /dev/vx/dsk/testdg/vol1 /mnt1
```

Where:

mode is one of the following:

■ writeback

■ read

- nocache

To change the caching mode of a VxFS mount point

◆ To change the caching mode of a file or directory, use the following command. You cannot change the caching mode to a higher level of caching than the mode set on the *mount_point*. For example, you cannot enable read caching if you specified the mode as `nocache` when you mounted the file system.

```
# sfcache set [-r] mode=[nocache|read|writeback] {file|dir}
```

Use the `-r` option to make the change recursive.

Flushing dirty data from a writeback cache area

With SmartIO, dirty data in the cache is automatically flushed to the disk during normal operations. The dirty data is flushed when the file system is unmounted, or during other operations that require a file system freeze. The dirty data is also flushed periodically at intervals. You can control the interval by configuring the tunable parameters.

See "Tuning the writeback caching" on page 53.

Disabling writeback caching for a file also flushes any writeback dirty data for that file.

In some cases, you may want to manually trigger flushing of the dirty data from the cache to the disk. For example, to ensure data consistency, you would flush the cache before you create an array level snapshot.

You can manually trigger flushing of the dirty data using the following command.

```
# sfcache flush [-r] {mount_point|directory|file}
```

Use the `-r` option to make the selection recursive.

Tuning the writeback caching

When writeback caching is enabled, any data that is read from the disk is cached, unless the file is explicitly marked for "no caching" or if the cache is full. For writes, certain writes cause the data to be cached. You can load a file to speed up the application. Pinning a file in the cache ensures that the data does not get evicted. If some data is already cached, and that portion of the disk is overwritten, then SmartIO also writes the new data to the cache device to ensure that the cached data remains up to date.

If you are using a database template, SmartIO caches according to the template rules.

See "About SmartIO caching for Oracle databases on VxFS file systems " on page 30.

You can use the following tunable parameters to adjust the size of the cache and how long data is held in the cache.

Setting the maximum space used for dirty data per node

Setting the maximum retention time used for dirty data

Setting the maximum space used for dirty data per node

When writeback is enabled, you can configure a maximum for the space used for dirty data per node. By default, there is no maximum. If you configure a maximum, you must allow at least 512 MB for each file system or cluster file system. For a cluster file system, the space required for data reflection is not included in the maximum.

Run the following command to configure the maximum. For a cluster file system, run the command on each node of the cluster to make the setting cluster wide.

```
# sfcache set writeback_size=size
```

Use the following command to view the current value:

```
# sfcache stat cachearea_name
```

See "Viewing the SmartIO cache statistics " on page 55.

Setting the maximum retention time used for dirty data

Dirty data is data in the cache that has not been flushed to the disk and so is out of sync with the data disk. The retention time determines how long the dirty data might remain unflushed. The default is 10 seconds.

For a cluster file system, run the command on each node of the cluster to make the setting cluster wide.

```
# sfcache set writeback_interval=interval
```

Use the following command to view the current value:

```
# sfcache stat cachearea_name
```

See "Viewing the SmartIO cache statistics " on page 55.

Viewing the SmartIO cache statistics

Use the `sfcache stat` command to display the cache statistics for the system.

Table 3-1 describes the cache statistics in the output.

Table 3-1   Cache statistics

Field	Description
HIT RATIO (VxVM cache)	Percentage of total I/Os that are satisfied from the cache. Displayed for reads and writes.
ART(Hit)ms (VxVM cache)	Average response time for I/Os that are satisfied from the cache.Displayed for reads (RD) and writes (WR).
ART(Miss)ms (VxVM cache)	Average response time for I/Os that are not satisfied from the cache. Displayed for r reads (RD) and writes (WR) .
BYTES (VxVM cache)	Total size of I/Os for reads (RD) and writes (WR).
NAME (VxVM cache)	Name of the cache area.
TYPE (VxVM cache)	Whether the cache area is VxVM or VxFS.
%CACHE (VxVM cache)	Percentage of the cache area that is currently used for data.for all data objects.
Cache Size (VxFS cache)	Size of the cache area.
Cache Utilization (VxFS cache)	Percentage of the cache area that is currently used for data.
File Systems Using Cache (VxFS cache)	Number of file systems using the cache.
Writeback Cache Use Limit (VxFS cache)	Size of the cache area that is used for writeback. You can set the size with the `writeback_size` attribute. If no size is set, the field displays Unlimited. See "Tuning the writeback caching" on page 53.

Table 3-1 Cache statistics *(continued)*

Field	Description
Writeback Flush Timelag	Interval between when the data is written to the cache and when it is flushed to the disk. If the Writeback Flush Timelag is small, such as 10 seconds, then sfcache statistics will not show. Data is flushed to disk faster. In this case, you can determine the caching usage based on the WB Hit Ratio.
Hit Ratio (VxFS cache)	Percentage of total I/Os that are satisfied from the cache. Displayed for reads and writes.
Data Read (VxFS cache)	Data read from the cache.
Data Written (VxFS cache)	Data written to the cache.
Files Cached (VxFS cache)	Number of files present in the cache.
Files Pinned (VxFS cache)	Number of pinned files in the cache.
Data Pinned (VxFS cache)	Amount of data pinned in the cache.

To view the cache statistics

◆ Use the following command:

```
# sfcache stat
```

```
                   HIT RATIO    ART(Hit)ms   ART(Miss)ms    BYTES
NAME            %CACHE RD    WR    RD    WR    RD    WR    RD     WR
TYPE: VxVM
sfcachearea_1   13.43  91.24 94.20  0.142 0.819  0.414 0.798  15.31g 4.21g


TYPE: VxFS
NAME: sfcachearea_2
          Cache Size:     48.0 GB
      Cache Utilization:    72.2 MB ( 0.15 %)
 File Systems Using Cache:       1
Writeback Cache Use Limit:  Unlimited
  Writeback Flush Timelag:     10 s


Read Cache
Hit Ratio  Data Read  Data Written  Files Cached  Files Pinned  Data Pinned
   0.00 %      0 KB          0 KB             0             0         0 KB
```

Viewing the detailed caching stats for a VxVM cache area

To view the detailed caching statistics for a VxVM cache area

◆ Use the following command:

    # **sfcache stat sfcachearea_1**

```
                      HIT RATIO    ART(Hit)ms   ART(Miss)ms   BYTES
NAME                %CACHE RD    WR     RD    WR     RD     WR      RD     WR
TYPE: VxVM
sfcachearea_1       13.43  91.24 94.20  0.142 0.819  0.414  0.798  15.31g 4.21g

ASSOCIATED DATA OBJECTS:
testdg/testvol1     6.10   90.00 96.00  0.141 0.459  0.348  0.448  6.77g  1.89g
testdg/testvol2     7.32   91.00 92.00  0.143 1.179  0.480  1.149  8.54g  2.31g
```

Viewing the detailed caching stats for a VxFS cache area

For a VxFS cache area, the statistics do not change after you unmount and mount the file systems. For a cluster file system, the statistics do not change after you reboot the cluster nodes.

To view the detailed caching statistics for a VxFS cache area

◆ Use the following command:

```
# sfcache stat sfcachearea_2

TYPE: VxFS
NAME: sfcachearea_2
              Cache Size:    9.97 GB
       Cache Utilization:    1.133 GB (11.37 %)
 File Systems Using Cache:       3
Writeback Cache Use Limit:      8 GB
  Writeback Flush Timelag:     10 s


Read Cache                                Writeback
Hit Ratio    Data Read    Data Written    Hit Ratio     Data Written

Total:
   0.00 %         0 KB       195.0 MB     100.00 %         78.0 MB

/mnt1:
   0.00 %         0 KB        78.0 MB     100.00 %         39.0 MB

/mnt2:
   0.00 %         0 KB        39.0 MB       0.00 %            0 KB

/mnt3:
   0.00 %         0 KB        78.0 MB     100.00 %         39.0 MB
```

Troubleshooting and error handling

This chapter includes the following topics:

- Support for a persistent or 'warm' VxVM cache

- Cache area is lost after a disk failure (3158482)

- Cache is not online after a reboot

- Recovering the writeback cache after a node failure

Support for a persistent or 'warm' VxVM cache

A warm cache means that the contents of the cache remain persistent across planned reboots. By default, SmartIO does not provide a warm cache capability for VxVM caches. The cache area metadata is not flushed during a system shut down. The cache is invalidated after the system reboot, and whenever the volumes need to be restarted. This behavior is known as a 'cold' cache.

Symantec does not recommend that you configure a warm cache, because it may lead to data inconsistency. However, in some circumstances, when proper care is taken, it can be beneficial to configure a warm cache. For example, in the case of a controlled, planned reboot you may want to explicitly enable a warm cache. This feature enables you to flush the metadata to the cache, thus creating a warm cache.

To enable a warm cache for a planned reboot

1   Before the system reboot, shut down all the applications.

2   Create the warm cache by running the following command:

```
# sfcache offline --flushmeta cachearea_name
```

If desired, you can add this command to your shutdown script.

You must run the above command before the application restart or system reboot. If not, the data in the cache is purged and the cache is not populated after the reboot.

Primary volume failure with a stale cache could cause possible data corruption

If the data in the primary volume needs to be restored, the existence of a warm cache may lead to the use of a stale cache. In some cases, the stale cache could cause data corruption. This situation is not common but the possibility exists if a restoration method such as the following is used:

- Restore from an array snapshot
- Restore of the primary LUN from backup using NetBackup or other backup software.
- VxVM configuration backup and restore.

Migrating a cache during HA failover is not supported

Even if a warm cache is enabled, SmartIO does not provide a capability to migrate the cache in a high availability cluster during a failover. The warm cache is not available to the application after failover.

Cache area is lost after a disk failure (3158482)

SmartIO supports one VxFS cache area and one VxVM cache area. If you create one cache area, and the disk fails, the cache area becomes disabled. If you attempt to create a second cache area of the other type before the cache disk group is enabled, then the first cache area is lost. It cannot be brought online.

For example, first you created a VxFS cache area. The disk failed and the cache area is disabled. Now create the VxVM cache area. While creating VxVM cache area, SmartIO looks for an existing default cache area. Due to the failed disk, the existing cache area cannot be found. So SmartIO creates a VxVM cache area with the same name. Now even if disk containing VxFS cache area comes up, SmartIO

cannot access the original cache area. In this scenario, the VxFS cache area is lost. Losing the cache area in this case does not result into any data loss or data inconsistency issues.

Workaround:

Create a new VxFS cache area.

Cache is not online after a reboot

Generally, the SmartIO cache is automatically brought online after a reboot of the system.

If the SSD driver module is not loaded automatically after the reboot, you need to load the driver using the Linux commands and bring the cache disk group online manually.

To bring a cache online after a reboot

1   Load the SSD driver module with the `insmod` command.

    See the Linux documentation for details.

2   Perform a scan of the OS devices:

    # **vxdisk scandisks**

3   Bring the cache online manually:

    # **vxdg import** *cachedg*

Recovering the writeback cache after a node failure

In writeback mode, SmartIO stores data in the cache before writing the data to the disk. If the device that contains the cached data fails, the unflushed data in the cache area must be flushed to the disk when the system comes back online. Until the cached data is flushed, the file data that is cached in writeback mode may not be completely present on the disk.

In some cases, the SmartIO cache containing dirty writeback data is not available for flushing. For example, an error may have occurred on the SSD device that contains the cache area. When the cache device comes back online, then dirty data for such files is skipped from flushing. The files that have pending dirty data are not accessible. Any I/O on those files (except deletion) returns an I/O error (EIO).

The following error message is displayed in the syslog:

```
Writeback cache recovery is failed for mounted_device with error error_code:
```

If you want to restore access to such files, use the following procedure.

Restoring access to files that are not accessible

1 After the data is restored, use the following command to enable access. This command requires root privileges.

```
# sfcache restore-access -r {mount_point|directory|file}
```

For example:

```
# /usr/sbin/sfcache restore-access /testFS
```

For cluster file systems, run the `sfcache restore-access` command on each node of the cluster.

2 To enable the writeback caching for the file system, use the following commands:

```
#  sfcache disable /testFS
# sfcache enable /testFS
```

Command reference

This appendix includes the following topics:

- SmartIO command reference

SmartIO command reference

Table A-1 lists commands for using the SmartIO feature.

Table A-1   SmartIO command reference

Command	Description
sfcache app	Applies the specified template name.
sfcache create	Creates a cache area.
sfcache delete	Deletes the specified cache area.
sfcache disable	Disables caching for the specified data object.
sfcache enable	Enables caching for the specified data object.
sfcache flush	Flushes any write-back data for this file system or cache.
sfcache list	Displays the cached file systems or volumes and their cache usage.
sfcache load	Loads the specified file into the cache area.

Table A-1   SmartIO command reference *(continued)*

Command	Description
sfcache maxsize	Displays the amount of free space in the devices that are already provisioned for caching.
sfcache offline	Stops VxFS or VxVM from using a cache area.
sfcache online	Explicitly makes a cache area available.
sfcache pin	Marks a file or directory to be held in the cache until the file or directory is deleted, truncated, or unpinned.
sfcache purge	Removes the cached contents for the specified file system.
sfcache resize	Resizes the specified cache area.
sfcache restore-access	Enables read or write access to files that are missing writeback data. This command does not restore the missing data.
sfcache rmdev	Removes the device or devices from use for caching.
sfcache set	Sets the values for the specified attributes.
sfcache stat	Displays the cache statistics, including cache hit rate, misses, average read and write latencies.
sfcache unpin	Removes the file or directory from the pinned state.

Index

V

W