Sun StorEdge™ Availability Suite Software — Performance Improvement

A Best Practice

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 816-7184-10
August 2002, Revision A

Send comments about this document to: docfeedback@sun.com

Contents

Sun StorEdge Availability Suite Software — Performance Improvement

The Sun StorEdge™ Availability Suite Software includes the point-in-time copy software and the remote mirror software. The point-in-time copy software and the remote mirror software are drivers that are inserted into the Solaris™ Operating Environment (Solaris OE) I/O stack. The storage volume (SV) kernel module intercepts I/O operations at entry to the Solaris device driver and redirects them to the point-in-time copy and remote mirror drivers. The drivers determine how to satisfy the I/O request and pass it to the underlying volume manager or raw storage device to complete.

This best practice document provides background information about how the point-in-time copy software and the remote mirror software impact the performance of the I/O data path. This document also provides general tuning recommendations to limit the impact of the software with respect to general application behavior and overall system performance.

The point-in-time copy software and the remote mirror software use system resources like any other software product. You should analyze your system to determine if it has enough memory, CPU, and network resources to support the desired configuration. The Sun StorEdge Availability Suite SoftwareSuite Software is not a volume manager. The point-in-time copy software and the remote mirror software do not configure or optimize underlying storage. The system administrator must lay out and tune the volumes to be used.

Each system is unique and performance can vary from system to system. You should individually test each tuning and configuration option that you are considering, and analyze the results to determine the optimal configuration for the system.

Bitmap Volumes

A bitmap volume tracks differences between a master volume and shadow volume in a point-in-time copy volume set. A bitmap volume also tracks the differences between the primary volume and the secondary volume in a remote mirror volume set. Each bit in the bitmap corresponds to a 32 Kbyte block of data on the volume being tracked. Depending on the type of point-in-time copy or remote mirror volume set and the behavior of the application accessing a volume, the bitmap volume can be accessed frequently. For this reason, you should understand how and when the bitmap volume is being accessed in a given environment so that the bitmap volume can have an optimal configuration. For details, see:

- "Bitmap Access Behavior" on page 11 for the remote mirror software
- "Bitmap Access Behavior" on page 13 for the point-in-time copy software

The point-in-time copy software and the remote mirror software do not alter the size of the I/O request from the application. In a point-in-time copy volume set, if an 8 Kbyte write to the master volume forces a copy-on-write operation, a 32 Kbyte block of data is copied to the shadow volume, but only the 8 Kbyte I/O requested by the application is written to the master volume.

Bitmap Volume Location

In write-intensive environments, performance is affected dramatically when bitmap volumes are placed on the same physical devices that contain the data volumes. In this situation, the disk has to write the data requested as well as write changes to the bitmap, which is on the same physical device. Placing the data volumes and bitmap volumes on different devices reduces contention, thus helping to improve overall performance.

If possible, place bitmap volumes on cached storage arrays. Cached storage arrays can provide quicker access for the frequent I/O operations performed on the bitmap.

Underlying Storage for Bitmap Volumes

For optimal performance and security, use RAID 1 + 0. When using RAID 1, use a good RAID controller to minimize impact when performing write operations. Configure bitmaps to reside on disk to insure that valid point-in-time copy and remote mirror data is retained if the system fails. When a bitmap is configured in this mode, changes to the bitmap are immediately written to nonvolatile storage: disk or non-volatile random access memory (NVRAM). If it is acceptable to perform a full copy to restore shadow or secondary data after a system failure and the loss of a dependent volume will not cause the loss of important data, then you can configure bitmaps to reside in memory to increase performance. When a bitmap is configured in this mode, the bitmap is read into kernel memory when the set is resumed or created, and changes are written to disk when the set is suspended.

Note – An unacceptable configuration and practice is to perform an update from a dependent shadow volume to the master volume with the bitmap in memory. If a system crash occurs in this scenario, the data on both the master volume and shadow volume become invalid.

To change the bitmap mode for the point-in-time copy software , change the value of `ii_bitmap` in `/usr/kernel/drv/ii.conf`. To change the bitmap mode for the remote mirror software, change the value of `rdc_bitmap_mode` in `/usr/kernel/drv/rdc.conf`.

See the following table, the `rdc.conf` file, and the `ii.conf` file for details on setting the bitmap mode.

Variable value	Point-in-Time Copy Software `ii_bitmap`	Remote Mirror Software `rdc_bitmap_mode`
`0`	The point-in-time copy bitmap is located in memory	Autodetect mode for the remote mirror bitmap is based on the state of the storage device block cache (SDBC). See **“SDBC Configuration and Tuning” on page 5**.
`1`	The point-in-time copy bitmap is located on disk	The remote mirror bitmap is located on disk
`2`	If the Sun StorEdge Fast Write Cache software is present, the point-in-time copy bitmap is stored in NVRAM, otherwise, the bitmap is located in memory	The remote mirror bitmap is located in memory

File System Considerations

File systems write meta data that is associated with each file or file system. This meta data must also be serviced by the point-in-time copy software and the remote mirror software. When you are using volumes containing file systems with point-in-time copy and remote mirror volume sets, investigate the behavior of the file system to determine the best layout and mount options for the file system to bring about optimal performance. Changing mount options can significantly reduce performance overhead.

SDBC Configuration and Tuning

The point-in-time copy software and the remote mirror software use the SDBC as an internal caching layer. This cache is used exclusively by the point-in-time copy software and the remote mirror software. Tune this cache based on the point-in-time copy software and the remote mirror software configurations.

Cache Hints

Cache behavior has four modes. These modes can be tuned for the entire system or for a specific volume. The SDBC operates only as a read cache on systems that do not have Sun StorEdge Fast Write Cache software and NVRAM cards.

- Read Cache On (`rdcache`)

  SDBC first looks in its cache to satisfy a read operation, if it is not found in cache, the data is retrieved from the underlying storage.

- Read Cache Off (`nordcache`)

  In this mode, cache blocks used to satisfy a read operation immediately become available to be swapped out if needed. SDBC still queries the cache before the disk to satisfy read requests.

- Write Through (`wrthru`)

  In `wrthru` mode, all writes are immediately written to the underlying storage. This is the default behavior for a system without the Fast Write Cache software installed.

- Write Through Off (`nowrthru`)

  In `nowrthru` mode, writes are written to the SDBC in non-volatile memory. The SDBC module then writes the data to the underlying system at a later time. This mode is only available when NVRAM cards and the Fast Write Cache software is installed.

Note – Fast Write Cache is not supported by the Sun StorEdge Availability Suite Software, therefore `wrthru` and `nowrthru` mode are not discussed further in the document.

The default cache mode for all volumes is `rdcache`. Depending on the type of the point-in-time copy or remote mirror volume set and application behavior, access to the bitmap volume can be very frequent. For optimal performance of bitmap access, the bitmap volumes should be configured for `rdcache` mode. If the cache mode for a bitmap volume is changed to `nordcache`, SDBC is more likely to get the data from the underlying storage each time the bitmap is accessed.

▼ To Change the Cache Mode for a Volume

- **Type the following examples to see the cache mode change.**

```
# /usr/opt/SUNWesm/sbin/scmadm -o /dev/vx/rdsk/testdg/vol0
scmadm: cd(0) Current options are: wrthru, rdcache (overridden by system)

# /usr/opt/SUNWesm/sbin/scmadm -o /dev/vx/rdsk/testdg/vol0 nordcache
scmadm: cd 0 option nordcache now set.

# /usr/opt/SUNWesm/sbin/scmadm -o /dev/vx/rdsk/testdg/vol0
scmadm: cd (0) Current options are: wrthru, nordcache (overridden by system)
```

Cache Size

The SDBC module allocates system memory from the Solaris OE as needed and then returns the memory to the Solaris OE when it is no longer needed. The cache size grows dynamically to an upper limit. The default upper limit is 64 Mbyte; the maximum size allowed is 1024 Mbyte. The upper limit is tunable and is changed with the utility `scmadm`, see “To Change the Size of the SDBC” on page 7. Increasing the upper limit of the SDBC size enables the Sun StorEdge Availability Suite Software to retain a larger amount of data in cache, reducing the chance of a cache miss that forces an I/O request to be satisfied by the underlying storage system.

Choosing an improper cache size can have a noticeable effect on the system. If the cache is configured to be too large, performance can suffer. On a system with limited memory, specifying too large a cache size enables SDBC to hold on to system memory that could be better utilized by other applications. If the cache is configured to be too small, the cache layer spends more time swapping, which limits the chances for a cache hit.

The minimum size of the cache should be 1 Mbyte per volume set configured plus 20%. When determining how much larger than the minimum to set the size of the cache, consider the following:

- I/O access pattern

  Random I/O patterns benefit more from a properly tuned cache size because there is a greater chance of a cache hit.

- I/O transfer size

  Smaller I/O transfer sizes can benefit from a properly sized cache, as there is a greater chance the I/O request is still located in the cache.

- Ratio of cache to working set of data

  Increasing the ratio of cache to working set of data helps increase the chance for a cache hit when an I/O request is serviced. No single ratio of cache to data is optimal for every system. Do a performance analysis for each system SDBC is to run on.

- Total amount of system memory

  The total amount of system memory available is one of the heaviest weighted factors when determining how to size the cache. On systems with lower amounts of system memory, overall system performance might be impacted if too much memory is allocated to SDBC.

▼ To Change the Size of the SDBC

1. **Read the current cache size.**

```
# /usr/opt/SUNWesm/sbin/scmadm -C
cache_size          : 64                    /* total cache size */
```

2. **Change the cache size.**

```
# /usr/opt/SUNWesm/sbin/scmadm -C cache_size=XX
cache_size          : 64 = XX            /* total cache size */
/** Note: cache_size is specified in Mbyte **/
Changed configuration parameters will take effect when the cache is restarted
```

3. **Restart the cache to verify the cache size change.**

```
# /usr/opt/SUNWesm/sbin/scmadm -d
scmadm: cache has been deconfigured
# /usr/opt/SUNWesm/sbin/scmadm -e
scmadm: cache has been configured
# /usr/opt/SUNWesm/sbin/scmadm -C
cache_size          : XX                    /* total cache size */
```

Configuring and Tuning the Remote Mirror Software

When the remote mirror software is being used, there should be adequate network bandwidth available to support the network traffic that it generates. Highly latent network links can have a dramatic effect on application performance for volumes configured in a remote mirror volume set. See the best practice document, *Sun StorEdge™ Availability Suite Software – Improving Data Replication Over a Highly Latent Network* (part number, 816-7180), for further information.

Remote mirror volume sets can be configured in two modes: sync mode and async mode. Volume sets in sync mode wait for a confirmation from the secondary site that the write was successful before returning control to the calling application. Volume sets in async mode place a write onto a remote mirror async I/O queue and return control immediately to the calling application. An async flusher thread then propagates the write to the secondary volume and removes it from the async queue. Under normal operation, reads are satisfied from the primary volume.

Sync Remote Mirror Volume Sets

Since a sync volume set requires confirmation from the secondary host, write intensive applications or write operations that occur over highly latent networks can take a longer time. No specific tuning parameters exist for a sync volume set, but performing larger sized writes can reduce the number of individual requests sent over the network.

Async Remote Mirror Volume Sets

Each async volume set has an I/O queue associated with it. When a write operation is requested, data is placed on the async volume set's associated I/O queue, the data is written to the local volume, and then control is returned to the calling application. If the async queue is full, the write is blocked until the async flusher thread has written enough data to the secondary host to enable the incoming write request to be added to the async I/O queue.

After an async I/O queue has filled, the I/O performance behaves more like a sync volume set until the I/O queue is emptied enough to enable writes to be added at a rate that does not fill the I/O queue. For this reason, the remote mirror async mode might not be a good solution for environments where the throughput of the network connection is less than the sustained write I/O generated by an application to the remote mirror primary volume. Async mode is intended to improve performance over highly latent networks where the sustained throughput of an application does not consistently exceed the capacity of the I/O queue associated with the remote mirror volume set. It is also intended for applications with write patterns that have burst peaks greater than the network throughput.

Async I/O Queue Sizing

Consider the following factors when choosing a size for an async I/O queue:

- System memory

  Each I/O queue is located in memory using memory that is allocated directly from the Solaris OE, with the default size of a queue being approximately 8 Mbyte. The memory for the async queue is allocated dynamically as required. For systems with limited memory, ensure that overall system performance is not impacted by too much memory being allocated to the async I/O queues.

- Number of remote mirror sets

  Each async volume set has its own I/O queue. If there are many async volume sets, the I/O queues can quickly consume system memory. Twenty remote mirror volume sets at the default value consume 160 Mbyte of system memory.

Note – When you are grouping volume sets, all remote mirror volume sets in the same group share a single I/O queue. See “Grouping” on page 12 for more detail.

- Application I/O

  Compare the amount of write I/O generated by applications accessing the primary volume to the network bandwidth that is available. The sustained write throughput should not exceed the available network bandwidth.

- Network latency

  Network latency is a major factor in how quickly data is transferred to the secondary site. On a fast network with low latency, data is transferred faster and the I/O queue fills slower, so you can use a smaller I/O queue. On a slow or highly latent network, data transfer takes more time, so the queue fills faster. In this situation, you might have to increase the size of the I/O queue so it does not fill up so fast.

- Acceptable differences

  Before you set the size of the I/O queue, determine how far out of sync the volume set can be at any point in time. If the size of the I/O queue is increased to 32 Mbyte, the secondary volume might be out of sync with the primary volume by as much as 32 Mbyte. Tuning the I/O queue size down to 4 Mbyte limits the difference to 4 Mbyte between the primary and secondary volume at any point in time.

Tuning the Async Queue

The async queue has two tunable parameters: the size of the queue and the number of writes in the queue. The size of the queue is the amount of data that can be queued at any one time. The number of writes is the amount of pending writes to be sent to the secondary volume at any one time.

kstat Statistics

There are two `kstat` statistics that are useful for determining to what size the async I/O queue should be set:

- `async_block_hwm`

  If the `async_block_hwm` exceeds the `maxqfbas`, then more data is being put on the queue than can be written to the secondary volume. If this happens, increase the size of the queue by increasing the value for `maxqfbas`.

- `async_item_hwm`

  If the `async_item_hwm` grows larger than the value of `maxqitems`, then individual write activity is being added to the queue faster than it can be replicated to the secondary volume. If this happens, increase the value of `maxqitems`.

▼ To Increase the Value of maxqitems

- **Type the following to see the indicated changes.**

```
# sndradm -P
/dev/vx/rdsk/testdg/vol0 ->shost:/dev/vx/rdsk/testdg/vol0
autosync: off, max q writes: 4194304, max q fbas: 16384, mode: async, state:
logging
# sndradm -W 8000  shost:/dev/vx/rdsk/testdg/vol0
Change SNDR tunable? (Y/N) [N]: y
#sndradm -P
/dev/vx/rdsk/testdg/vol0 ->shost:/dev/vx/rdsk/testdg/vol0
autosync: off, max q writes: 8000, max q fbas: 16384, mode: async, state: logging
# sndradm -F 600 shost:/dev/vx/rdsk/testdg/vol0
Change SNDR tunable? (Y/N) [N]: y
# sndradm -P
/dev/vx/rdsk/testdg/vol0 ->shost:/dev/vx/rdsk/testdg/vol0
autosync: off, max q writes: 8000, max q fbas: 600, mode: async, state: logging
```

Bitmap Access Behavior

When a remote mirror async volume set is in replicating mode, a write to the primary volume updates the bitmap to reflect the change to the primary volume. The bitmap update indicates that the write was replicated to the secondary volume only when the primary host receives confirmation from the secondary host that the write was successful. When a remote mirror volume set is in logging mode, the bitmap on the local host is updated to reflect any write issued to the local volume. This is true on both the primary and secondary hosts.

When a sync or reverse sync is issued, the primary and secondary bitmaps are compared to determine which blocks need to be replicated to the target volume. For read operations to the primary volume during a reverse sync, the bitmap is consulted to determine if the requested block must first be copied from the secondary volume before performing the read operation.

RPC Timeout Values

In an environment with high network latency, remote mirror operations can time out before the operation completes. Timing out causes a remote mirror volume set to be forced into logging mode when you might not want it to. In this situation, you can increase the RPC timeout value to help prevent remote mirror volume sets from going into logging mode due to network latency.

▼ To Increase the Timeout Value

- **Add the following line to the** `/etc/system` **file.**

```
set rdc:rdc_rpc_tmout=nn
```

Where *nn* equals the desired timeout.

Grouping

In the Sun SNDR 2.0 software, which is an earlier version of the remote mirror software, the only way to guarantee write ordering across multiple volume sets was to configure each set in sync mode. This changed with the introduction of I/O groups in Sun SNDR 3.0. Multiple async volume sets can be configured within the same I/O group and write ordering is preserved within the I/O group.

When a remote mirror group is created, all volume sets in the group share a single async I/O queue. Make sure that you account for all I/O to all sets in the group when you determine the size of the async I/O queue for the group. Make sure that all sets in an I/O group are optimally configured. Since write ordering is preserved across the group, one poorly configured set could significantly impact all volumes in the I/O group. Consider the following scenarios:

- An I/O group contains 10 remote mirror volume sets. Seven volume sets connect to hostA through gigabit ethernet, two volume sets connect to hostB through 100BASE-T, and one set connects to hostC through 10BASE-T. If either the 100BASE-T or 10BASE-T networks become saturated, access to the volumes in the sets connected to the secondary system through the gigabit network is restricted until the saturated network is relieved.
- An I/O group contains three remote mirror volume sets. Two of the volumes in the set are configured as RAID 0 volumes on fast storage, and the remaining volume is configured as a plain volume on slow storage. If the I/O to the volume located on slow storage is forced to wait, I/O to the volumes on fast storage must also wait.

Interaction With the Point-in-Time Copy Software

Connecting volume sets through a fast connection can help enable the remote mirror software to send larger chunks of data across the connection. If the secondary remote mirror volume is in a point-in-time copy volume set, the incoming I/O is broken into 32 Kbyte writes to the remote mirror secondary volume.

Configuring and Tuning the Point-in-Time Copy Software

This section contains additional information for understanding how the point-in-time copy software affects system performance.

Bitmap Access Behavior

The bitmaps associated with point-in-time copy volume sets are accessed as follows.

- Write to master volume

  When a write to the master volume occurs, the bitmap is consulted to determine if a copy on write is required. The bitmap is then updated to reflect the change to the master volume.

- I/O to a dependent or a compact dependent shadow volume

  When a read occurs to a dependent or a compact dependent shadow volume, the bitmap is consulted to determine if the read should be satisfied from the master volume or the shadow volume. When a write to the shadow volume occurs, the bitmap is updated to reflect the change to the shadow volume.

- Write while copy or update in progress

  When a copy or update is in progress, the bitmap is consulted to determine if a copy on write must first be performed. Then the bitmap is updated to reflect the change written to the target volume.

- Read while copy or update in progress

  When a copy or update is in progress, the bitmap is consulted to determine whether the read can be satisfied from the requested volume. If not, the requested data needs to be copied first from the volume initiating the copy to the volume receiving the copy before satisfying the read.

Tuning Copy Parameters

Use the iiadm -P command to tune system copy parameters. You can change the number of chunks of data (copy units) copied without a pause from 100 to 60000. You can change the number of system clock ticks (delay) that the system waits between copying each series of data chunks from 2 ticks to 10000 ticks. If you set copy units to 60000 and delay to 2 ticks, the copy proceeds at the highest possible speed. If you set the copy units to 100 and delay to 10000, the copy proceeds at the slowest possible speed. During a high-speed copy, system resources are heavily committed to the copy, possibly at the expense of other applications. During a slow-speed copy, other tasks execute almost normally, but the copy can take a long time. Set the copy parameters to suit your particular environment. If your system is committed to just a few tasks, you can probably set the parameters for a fast copy. If your system is performing a number of tasks, set the parameters for a slower copy to allow the other tasks useful access to system resources.

See the *Sun StorEdge Availability Suite 3.1 Point-in-Time Copy Software Administration and Operation Guide* or the *Sun StorEdge Instant Image 3.0.1 System Administrator's Guide* for more information.

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 USA
650-960-1300 Fax 650-969-9131

For U.S. sales office locations, call:	800-821-4643
In other countries, call:	
Corporate headquarters:	650-960-1300
Intercontinental sales:	650-688-9000