Configuring Boot Disks With Solaris™ Volume Manager Software

Erik Vanden Meersch - Global Sales
Kristien Hens - Enterprise Services

Sun BluePrints™ OnLine - October 2002

Sun microsystems

Configuring Boot Disks With Solaris™ Volume Manager Software

This document is a practical guide for system disk configuration. It describes how to partition the system disk, mirror it, and create and maintain a contingency boot disk.

We attempt to cover a wide range of availability requirements by presenting two-, three-, and four-disk configurations. Every additional disk adds protection. The idea is to select one configuration, based on the cost/benefit ratio in a given situation, then use the corresponding runbook as a guide for implementation. The SUNBEsdm `package` is also provided with scripts for a fully automatic implementation.

In this document, we follow the concepts of the Sun BluePrints™ book "Boot Disk Management—A Guide for the Solaris™ Operating Environment" by John S. Howard and David Deeths. We add an explicit comparison of the use of Solaris™ Volume Manager software and VERITAS Volume Manager (VxVM) software for mirroring the system disk.

This article is an update to "Configuring Boot Disks With Solstice DiskSuite™ Software," by Erik VandenMeersch and Kristien Hens (Part #816-4526-10). The basis for the current text is the Solaris 9 Operating Environment, Solaris Volume Manager software, and VxVM 3.2 software.

This Sun BluePrints OnLine article covers the following topics:

- Hardware configuration
- System disk partitioning
- Two-disk configuration
- Three-disk configuration
- Four-disk configuration
- Comparison of Solaris Volume Manager software and VxVM software
- Runbook for creating the Solaris Volume Manager software state database
- Runbook for a two-disk configuration

- Runbook for a three-disk configuration
- Runbook for a four-disk configuration
- Uses of SUNBEsdm scripts

Introduction

System disks can be protected two ways: disk mirroring and file system backup. These two methods are complementary and cover different types of failure. Disk mirroring keeps the system running when a disk stops responding to commands. Backup provides a recovery path when files (or entire file systems) are removed or corrupted.

Sun Microsystems supports two volume managers for mirroring system disks: VERITAS Volume Manager (VxVM) software and Solaris Volume Manager software. VxVM software is widely used on servers with a large number of disks and logical volumes and can be considered a de facto standard.

We advocate using Solaris Volume Manager software for mirroring the system disk, even when VxVM software is chosen for volume management of application data. This choice is not obvious at first sight because both Solaris Volume Manager software and VxVM software work perfectly under predictable circumstances. The difference becomes apparent in case of disaster (defined in this context as a situation where, due to some problem, the system no longer boots).

It is not the intention of this document to provide an absolute best practice for system disk configuration. Instead, our goal is to:

- Present the arguments in favor of using Solaris Volume Manager software for the system disk. These arguments are familiar to many people, who often learned them the hard way, but we have not found them to be formally documented.
- Make life easier for the Solaris Volume Manager software user by providing runbooks with the commands detailed and the correct sequences provided. An engineer who routinely installs systems may not need this information, but we believe that there is a large audience who can benefit from some assistance in this area. We assume some level of Solaris Operating Environment and Solaris Volume Manager software knowledge.
- Provide the scripts for fully automatic implementation. The scripts essentially execute runbooks.

Runbooks and scripts are based on system disk partitioning in `/`, `/var`, and swap. This partitioning is widely accepted as a good practice. Because a carelessly partitioned system disk may cause considerable trouble, we dedicate a separate section in this article to discussing the rationale of the `/`, `/var`, and swap partitioning.

Hardware Configuration

Choosing the correct hardware is as important as choosing the correct volume manager. To start with, all disks used for the system should be of identical geometry. This prevents wasting disk space and facilitates the creation of the mirror, the contingency disk, and the hot spare disk. Further, to ensure maximum system availability, connect all disks to different host bus adapters (HBAs). If this is not possible, ensure that at least two HBAs are available, and that the system disk and its mirror are on different HBAs. Finally, if possible, ensure that HBAs are on different I/O boards to avoid a single point of failure within the server.

IDE, SCSI, or FC-AL drives can be used. For IDE drives, ensure that an IDE master is not mirrored to an IDE slave, as loss of the master means loss of access to the slave. We do not recommend that you use LUNs on a hardware RAID controller because this introduces extra components (and therefore, possible points of failure) to the boot path.

System Disk Partitioning

The Solaris Operating Environment installation procedure offers unlimited freedom for system disk partitioning. Creating a separate file system for a directory has a fundamental disadvantage: this file system can be full while there is ample free space elsewhere on the same disk. At that point, you may regret partitioning the disk and will have to have someone correct the situation. We recommend you follow this basic rule:

Do not partition unless there is a compelling reason to do so.

This leads to the following recommended system disk partitioning at installation. (The following partition sizes are reasonable defaults for a system disk of at least 9 GB.)

0	/	(recommended size + 1 GB)
1	swap	(rest of the disk)
3	/var	(recommended size + 2 GB)
4		(2 GB dedicated for crash dumps)
7		(30 MB reserved for the Solaris Volume Manager software state database)

/var is on a separate partition because unprivileged users can fill the /var directory in various ways that are hard to control, as shown in the following example.

/var/tmp	
/var/mail	(repeat mailx root < bigfile)
/var/spool/lp	(if a local printer is configured)
/var/adm/messages	(repeat logger -p daemon.err "message sent on $(date)")

By separating /var from /, unprivileged users cannot write to the root file system, and can, therefore, also not fill it.

The size of the /var file system must be sufficient to hold a kernel crash dump and is, therefore, at least as large as the dedicated crash dump partition.

Partition 4 is configured as a dedicated area for kernel crash dumps. The advantage of a dedicated, unmirrored dump partition is that the crash dump does not depend on the integrity of the mirror driver at the time of the panic. An additional advantage is faster boot after a system panic. The crash dump file can be constructed from the dump device in background, without holding up the boot process.

Configure the dedicated dump device using dumpadm -d /dev/dsk/c0t0d0s4.

The swap partition absorbs all the remaining disk space. If / or /var should fill up in spite of the large margins, the swap device can be taken off-line, shrunk, and added back. The obtained disk cylinders can be used to solve the problem (for example, by creating an /opt file system).

This configuration is preferable to keeping unused spare cylinders on the disk. If these spare cylinders must later be turned into additional swap space, two swap devices are created on different locations of the same physical disk, with a negative effect on system performance.

Two-Disk Configuration

Servers may have hardware or other limitations of two disks that can be used for the system.

In a two-disk configuration, the system disk is mirrored on a second identical disk. Both disks are bootable.

Although it is usually taken for granted that disks are mirrored in a two-disk configuration, we want to stress that this is not the best choice in all cases. For example, consider a system that is used as a JumpStart™ technology server where continuous availability of the server is less important than protecting the configuration files that exist on this system. From this perspective, using the second disk as a contingency disk may be a better choice than an on-line mirror.

Three-Disk Configuration

Whenever possible, use a third disk to protect the boot environments of systems with high availability requirements. This third disk could be used as a hot spare for the mirror, but in most cases, this is not optimal. Instead, we recommend that you configure the disk as a contingency system disk.

A contingency disk is a disk that contains backup `/` and `/var` file systems. These file systems are mounted at regular intervals and are synchronized with the active `/` and `/var` file systems. During the rest of the time, the file systems are unmounted and inaccessible.

The `system` and `vfstab` files on the contingency disk are modified to remove any dependencies on a volume manager. The contingency disk is made bootable and, as such, provides instant recovery from the accidental removal or corruption of system files (or of entire file systems) that are essential for booting. A contingency disk can also provide recovery from a failure of the only remaining disk in the mirror before the broken disk is replaced and completely synchronized.

A contingency disk is more likely to prove its usefulness during the lifetime of a server than a hot spare disk. Especially at those moments where an important reconfiguration or intervention is planned, it is very comforting to know that the contingency disk is there as a safety net (versus a much less attractive recovery from tape).

The protection provided by a contingency disk is not absolute. When a system file is modified or removed, the impact may not be seen until the next boot. By then, errors may have propagated to the contingency disk. While you should be aware that this is possible, it is rather unlikely. Bootability problems usually result from mistakes or setbacks that occur during major software or hardware reconfigurations, and they generally appear immediately.

Absolute protection can only be provided by synchronizing the contingency disk manually, on scheduled maintenance, after verifying that the system boots correctly. Between these maintenance slots, incremental backups must be made and stored separately, and must not be directly applied to the contingency disk.

Note that in the runbook we have chosen `ufsdump` to create the contingency disk. Another option would be to use Live Upgrade software. Refer to the *Solaris Live Upgrade Guide* at `http://docs.sun.com` for more information.

Four-Disk Configuration

The four-disk configuration is identical to the three-disk configuration, with the addition of a fourth disk, which is a hot spare for the mirror.

This configuration minimizes the time window between the failure of one of the mirrored disks and its replacement, as this now happens automatically. Although the time window is minimized, it is not eliminated because the failure of the remaining disk, before the hot spare is completely synchronized, is still fatal.

There is an argument for triple mirroring instead of configuring a hot spare disk. The third mirror can be seen as a hot spare that is already synchronized, and the time window, during which the system is vulnerable, is effectively reduced to zero. On the other hand, there may be a performance impact when writing intensively to the mirror. We leave the choice to you.

Comparison of Solaris Volume Manager Software and VERITAS Volume Manager

VERITAS Volume Manager (VxVM) software is widely used for logical volume management on systems with large amounts of data. One of the main reasons is its flexibility. Until the introduction of soft partitioning in the Solaris 9 Operating Environment, the logical volumes of Solaris Volume Manager software were based on partitions. Changing a Solaris Volume Manager software configuration involved the use of the `format` utility, a primitive command line tool where a fatal mistake was never far away. VxVM software replaces partitions with subdisks, which are unlimited in number, internally managed, and automatically allocated.

Despite the popularity of VxVM software, we strongly discourage its use on the system disk. VERITAS volumes do not, by default, correspond to partitions. In the situation, irrespective of the cause, where the system no longer boots, the system administrator must be able to gain access to the file systems on the system disk without the drivers of the volume management software. This is guaranteed to be possible when each volume corresponds to a partition in the volume table of contents (VTOC) of the system disk. You can then use the `fsck` command and mount the partition while booted from a CD, boot server, or contingency disk, without dependency on the logical volume configuration database.

Moreover, Solaris Volume Manager drivers are integrated in the Solaris 9 Operating Environment. Therefore, Solaris Volume Manager volumes can be accessed even when booted from CD-ROM. This eliminates the need of breaking off a mirror during upgrades, thus reducing downtime and complexity of such an operation.

Solaris Volume Manager software preserves the correspondence between the volumes defined in its state database, and the disk partitions defined in the disk label (VTOC), at all times; disaster recovery is always possible by a standard method, without extra complications.

VxVM software also preserves partition information, but not under all circumstances. Examples of how partition information in the VTOC breaks with VxVM software include:

■ It is easy to grow `/var` using the VxVM graphical tool. This can be done by anyone at any time, to solve a disk space problem. However, this breaks the volume-partition relation as the `/var` volume is now a concatenation of two (not necessarily contiguous) subdisks.

■ When a disk breaks, the replacement disk is initialized. Slices 3 and 4 become the VxVM private and public region, and subdisks are allocated to be mirrored with the surviving disk. Partitions may be created by VxVM software for these

subdisks, but not necessarily in the same VTOC slots. At that point there may be a partition for the file system, but nobody knows where (the `vfstab` file that was saved before encapsulation is incorrect).

■ Shrinking the swap device to create, for example, to `/opt`, produces a volume without corresponding partition in the VTOC of the system disk.

The system administrator can, in each of the preceding circumstances, take actions to preserve the correspondence between VERITAS volumes and disk partitions. However, this is not the default behavior and requires specific procedures and commands, as well as a better than average understanding of the behavior of VxVM software. We recommend reading the Sun BluePrints book "Boot Disk Management — A Guide for the Solaris Operating Environment" by John S. Howard and David Deeths for more information about this topic.

There are two drawbacks to using Solaris Volume Manager software in combination with VxVM software:

■ First, using Solaris Volume Manager software and VxVM software on the same system has a cost: VxVM software requires a mandatory disk group, `rootdg`, which should not be used to contain data (a basic best practice of VxVM software). If the system disk is mirrored with VxVM software, `rootdg` is automatically populated with the system disk and its mirror. If the system disk is mirrored with Solaris Volume Manager software, two other disks must be put in `rootdg`. Since `rootdg` may not survive certain problems with VxVM software, these disks should only contain volatile data (like swap space or a scratch file system, analogous to `/tmp`). If there is no need for this, these two disks are an extra cost.

■ Second, Solaris Volume Manager software requires that a majority of the state databases be found at boot time (the quorum rule). When all data disks are under VxVM software, only two disks may be left under Solaris Volume Manager software. If one of these disks breaks, there is no state database quorum and the system will not boot without manual intervention. The intervention consists of removing the inaccessible state database copies (using the `metadb -d` command) and rebooting.

For this reason, and only in the two-disk configuration, we recommend that you disable the quorum rule in `/etc/system`. The system will then boot unattended, even with one disk. The following section describes this feature.

Runbook for Creating the Solaris Volume Manager Software State Database

In runbooks, the following disk names are assumed.

primary mirror	c0t0d0
secondary mirror	c0t1d0
backup system disk	c0t2d0
hot spare disk	c0t3d0

These names are placeholders for the real disk names. In most cases, the logical disk names will be different.

As a first step for each of the presented configurations, the Solaris Volume Manager software state database replicas must be created. Providing more than one replica protects the configuration against losing the definition of the volumes.

Solaris Volume Manager software has the following behavior with respect to loss of replicas:

- The system panics when more than half of the replicas break.
- The system does not boot unless more than half of the replicas are available. The system administrator must remove broken replicas in single-user mode to restore the majority.

Ideally, three replicas are created, each on a different disk, connected to a different host bus adapter. In this runbook, we assume that only the system disks are under control of Solaris Volume Manager software, and this ideal replica configuration is not always possible.

Be aware that the optimal replica configuration may be different from the one presented here, in the case that the data disks are also under control of Solaris Volume Manager software and can contain state database replicas.

Provide the following commands for replica creation for the two-disk configuration.

```
metadb -a -f -c 2 c0t0d0s7
metadb -a -c 2 c0t1d0s7
```

You can choose to force Solstice DiskSuite software to start if half of the state database replicas are available by setting the tunable `mirrored_root_flag` to 1 in `/etc/system`. The default value of this tunable is disabled, which requires that a majority of all replicas are available and are in sync before Solstice DiskSuite software will start. To enable this tunable, type the following:

```
echo set md:mirrored_root_flag=1 >> /etc/system
```

Caution – Enable the `mirrored_root_flag` only if the following prerequisites are met: 1) The configuration only has two disks. 2) Unattended reboots of a system with only two disks are a requirement and the risk of booting from a stale replica, and therefore, stale volume management system state, is acceptable. This means that in your setup, system availability is more important than data consistency and integrity.
Setting this tunable risks data corruption or data loss in the event of a transient disk failure, but allows unattended reboots. See the Solaris Volume Manager documentation on `docs.sun.com` for more information about state database replicas (metadbs).

Replica creation for the three-disk and four-disk configuration. (Do not create unnecessary multiple copies per disk.)

```
metadb -a -f  c0t0d0s7
metadb -a     c0t1d0s7
metadb -a     c0t2d0s7
```

Runbook for the Two-Disk Configuration

Copy the VTOC to the mirror disk. It is not necessary to make this disk bootable because the bootblock from the first submirror is copied to the second submirror during the resync operation. Use the `prtvtoc` command as follows.

```
prtvtoc /dev/rdsk/c0t0d0s2 | fmthard -s - /dev/rdsk/c0t1d0s2
```

Create the state database replicas as described in the previous runbook.

Create the volumes, as follows.

```
metainit -f d10 1 1 c0t0d0s0
metainit -f d11 1 1 c0t0d0s1
metainit -f d13 1 1 c0t0d0s3

metainit d20 1 1 c0t1d0s0
metainit d21 1 1 c0t1d0s1
metainit d23 1 1 c0t1d0s3

metainit d0 -m d10
metainit d1 -m d11
metainit d3 -m d13
```

Make the system boot from the mirror, as follows.

```
metaroot d0
```

Replace the remaining partitions with volumes in /etc/vfstab, as follows.

```
cp /etc/vfstab /etc/vfstab.tmp
 cat /etc/vfstab.tmp | \
     sed s,/dev/dsk/c0t0d0s1,/dev/md/dsk/d1, | \
     sed s,/dev/dsk/c0t0d0s3,/dev/md/dsk/d3, | \
     sed s,/dev/rdsk/c0t0d0s3,/dev/md/rdsk/d3, > /etc/vfstab
```

Document the physical device name of the two disks in the mirror, as shown in the underlined portion of the following example for the disk c0t1d0.

```
ls -l /dev/rdsk/c0t1d0s0
lrwxrwxrwx   1 root    other    68 Nov 30 15:43 /dev/rdsk/c0t1d0s0 -
> ../../devices/ssm@0,0/pci@18,700000/pci@2/SUNW,isptwo@4/
sd@1,0:a,raw
```

Halt the system, as follows.

```
init 0
```

Create an OBP alias for the bootable disks. Beware that for disks on certain controllers, the sd driver name must be replaced by disk. Verify this, using the OBP show-disks command before entering the nvalias commands.

```
ok nvalias bootdisk /ssm@0,0/pci@18,700000/pci@2/SUNW,isptwo@4/sd@0,0:a
ok nvalias mirrdisk /ssm@0,0/pci@18,700000/pci@2/SUNW,isptwo@4/sd@1,0:a
ok setenv boot-device bootdisk mirrdisk
ok boot
```

Attach the submirrors, as follows.

```
metattach d0 d20
metattach d1 d21
metattach d3 d23
```

Runbook for the Three-Disk Configuration

Copy the VTOC and the boot block to the contingency disk, as shown here.

```
prtvtoc /dev/rdsk/c0t0d0s2| fmthard -s - /dev/rdsk/c0t2d0s2
/usr/sbin/installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk
/dev/rdsk/c0t2d0s2
```

Create the state databases and volumes as described in the runbook for the two-disk configuration.

Perform a full backup of the file systems to the backup system disk and modify system files so that the disk becomes bootable from partitions.

```
newfs /dev/rdsk/c0t2d0s0
mount /dev/dsk/c0t2d0s0 /mnt
ufsdump 0f  -  /  |  (cd /mnt; ufsrestore rf - )
cat /etc/vfstab | \
    sed s,/dev/md/dsk/d0,/dev/dsk/c0t2d0s0, | \
    sed s,/dev/md/dsk/d1,/dev/dsk/c0t2d0s1, | \
    sed s,/dev/md/dsk/d3,/dev/dsk/c0t2d0s3, | \
    sed s,/dev/md/rdsk/d0,/dev/rdsk/c0t2d0s0, | \
    sed s,/dev/md/rdsk/d1,/dev/rdsk/c0t2d0s1, | \
    sed s,/dev/md/rdsk/d3,/dev/rdsk/c0t2d0s3,  > /mnt/etc/vfstab
  cat /etc/system | grep -v rootdev > /mnt/etc/system
  umount /mnt

  newfs /dev/rdsk/c0t2d0s3
  mount /dev/dsk/c0t2d0s3 /mnt
  ufsdump 0f  -  /var |  (cd /mnt; ufsrestore rf - )
  umount /mnt
```

Put the preceding commands in a script `/usr/local/system-backup`, make the script executable, and execute from `cron`, as shown here. Note that this is the most rudimentary implementation. You may want to refine this by using a combination of full and incremental `ufsdump` and more sophisticated update schemes.

```
10   1   *   *   0     /usr/local/system-backup
```

Create an OBP alias for the contingency disk, as described earlier.

Runbook for the Four-Disk Configuration

Copy the VTOC and boot block to the fourth disk. This disk must also be made bootable for the unlikely situation that the hot spare actually kicks in and later becomes the only surviving disk in the mirror.

```
prtvtoc /dev/rdsk/c0t0d0s2| fmthard -s - /dev/rdsk/c0t3d0s2
/usr/sbin/installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk
/dev/rdsk/c0t3d0s2
```

Create the state database replicas and volumes, configure the contingency disk as described in the previous runbook.

Configure the hot spare pools. Each mirror gets its own hot spare pool with a partition of the corresponding size, as follows. Putting all spare partitions in the same pool is simpler, but correct behavior then depends on the order of the partitions in the pool.

```
metahs -a hsp000 c0t3d0s0
metaparam -h hsp000 d10
metaparam -h hsp000 d20

metahs -a hsp001 c0t3d0s1
metaparam -h hsp001 d11
metaparam -h hsp001 d21

metahs -a hsp003 c0t3d0s3
metaparam -h hsp003 d13
metaparam -h hsp003 d23
```

Use of the SUNBEsdm Package

With this article, we provide the SUNBEsdm package. The package contains a script `sdmsetup` that greatly reduces the manual work to implement the runbooks mentioned in this article. The SUNBEsdm package is not supported by Sun, and is only provided, here, to demonstrate our example.

Take care to minimize the risk of running the script `sdmsetup`. This script does not modify your system, but rather generates the commands to be executed, allowing for inspection prior to execution.

The prerequisites for `sdmsetup` are as follows:

- Solstice DiskSuite 4.2.1 software must be installed.
- No state database or volume must already exist.
- A free slice must exist to be used as state database.
- A free slice must exist to be used as dump device (optional).
- All currently mounted file systems must correspond to entries in `/etc/vfstab`.
- All used disks must be of the same geometry.
- Slice 2 on all disks must cover the entire disk.

A README file in the package contains detailed instructions for the usage of the script.

About the Authors

Erik Vanden Meersch is a Technology Expert for Sun in Northern Europe, focusing on servers and Solaris. Erik is often involved in designing architectures for environments with high availability requirements. With Sun for five years, Erik entered the company as a Solaris instructor. He has a degree in electronics engineering, and his career, prior to joining Sun, includes research on algorithms for Very Large Scale Integration (VLSI) design, software engineering, and CAD support.

Kristien Hens has worked for Sun for three years. She is currently an instructor for Internal Technical Training, teaching mostly Sun Cluster courses. Before joining Sun, Kristien worked as a developer in the field of computational linguistics.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`