JANE MARINSKY

A step-by-step guide to file system management, including data striping, data mirroring and logging file systems with Solstice DiskSuite

Apps in Action

Using Solstice DiskSuite

by Kailash Jayaswal

Now that disks are becoming cheaper and users are demanding ever larger file systems, spanning multiple disks, online resizing, data mirroring and striping have surfaced as hot topics in systems management. How can you accomplish these for your users? Solstice DiskSuite effectively tackles these and other tasks. It is included with the Solaris Server CD set and requires no licenses. This article describes some features of the Solstice DiskSuite and ways to implement it.

With DiskSuite you can stripe data, mirror data and log file systems. Striping allows you to create a file system with two or more disk partitions that are concatenated or striped. Striped file systems provide improved I/O rates as data is split into chunks and written to separate disks.

Mirroring distributes multiple copies of your data on different disks. If a disk with one copy fails, the file system can continue working with other copies of the data, residing on other disks.

UFS (UNIX File System) logging is useful for large file systems or file systems where data is constantly being modified. As a general rule, it is unnecessary to log file systems with mostly read activity. A file system with UFS logging has a master device (containing a new or existing file system) and a logging device. All file system writes are safely recorded in the logging device, which could be a disk partition or metadevice. The host system is then sent a message that the write operation to the file system has been completed. The host system goes ahead with the next operation. Later, the write to the logging device is rolled forward to the actual file system on the master device.

Because synchronous writes to the disk are reduced, local directory operations are faster. In addition, because the logging device stores only complete logs and all partially recorded logs are discarded, the file system does not need to be checked during reboots. This speeds the bootup process.

In addition to striping, mirroring and logging, DiskSuite handles hot spares and shared disksets. It also supports software RAID Level 5. Simply put, a hot spare is a disk partition configured to automatically replace a failed disk partition that is a component of a submirror or RAID 5 device.

A shared diskset is set of disks connected to and shared by two host computers. If one of the hosts fails, data on the diskset is still available through the other host. The disks in the shared diskset can be used for hot spares or file systems by both host computers. However, one of the two hosts is the implicit owner of the shared diskset. Besides using the shared diskset, each host has one or more local disks that are available exclusively to the host.

A RAID 5 device consists of three or more disk partitions (called components or columns). As with mirroring, a RAID 5 device continues functioning despite failure of a single component. RAID 5 offers some of the benefits of mirroring but at a lower cost and, like striping, provides improved I/O rates. DiskSuite 4.0 is the first version to support RAID 5 configurations.

OS Requirements, Setup

To use Solstice DiskSuite 4.0, you need Solaris 2.3 or later running on Sun SPARC or Solaris 2.4 or later running on an Intel Corp. X86-based system. If you intend to use the UFS logging feature of DiskSuite 4.0, you need Solaris 2.4 or later. The diskset feature of DiskSuite 4.0 is supported on SPARC systems running Solaris 2.4 or a later and is not supported on X86. (Editor's note: Version 4.1 of DiskSuite is now available.)

As with most Solaris software from Sun, `pkgadd` is used to install DiskSuite 4.0.

```
#cd /cdrom/disksuite_4_0
# pkgadd -d `pwd`
```

The following packages are available for installation: SUNWabmd (DiskSuite Answerbook); SUNWmd (DiskSuite software, metadevice drivers, binaries, man pages and so on); and SUNWmdg (DiskSuite Tool, GUI interface). In addition, two DiskSuite start-up files are installed:

```
/etc/init.d/SUNWmd.init
/etc/init.d/SUNWmd.sync
```

Creating State Database Replicas

DiskSuite devices are called metadevices, and the disk partitions are usually referred to as components. Before using DiskSuite you need to create at least three copies of the metadevice state database. The metadevice state database keeps DiskSuite operating and holds information about metamirrors, submirrors, concatenations, stripes, hot spares, error conditions and so on.

State database copies (517 Kb) or replicas can exist on either of the following:

- An unused partition that will be used to store only the metadevice state database.
- A disk partition that doesn't have a file system or swap and will later be used as a DiskSuite metadevice or logging device, with the exception of an OS file system (`root`, `usr`, `swap`).

To protect DiskSuite against controller or disk failure, put only one copy of the replica on each disk and make sure that the disks are attached to different controllers.

Once you have decided which partitions (or components) will contain the state database, you can use the `metadb` command to create the state database:

```
#metadb -a -f /dev/dsk/c1t3d0s4 \
   /dev/dsk/c0t3d0s6 /dev/c0t1d0s5
```

The above one-line command places one copy of the state database on each of the three specified partitions. The `-a` option is used to create a replica of the state database. The `-f` option forces the creation if no prior state databases exist. The following command is used to check the status of the state database:

```
#metadb -i
```

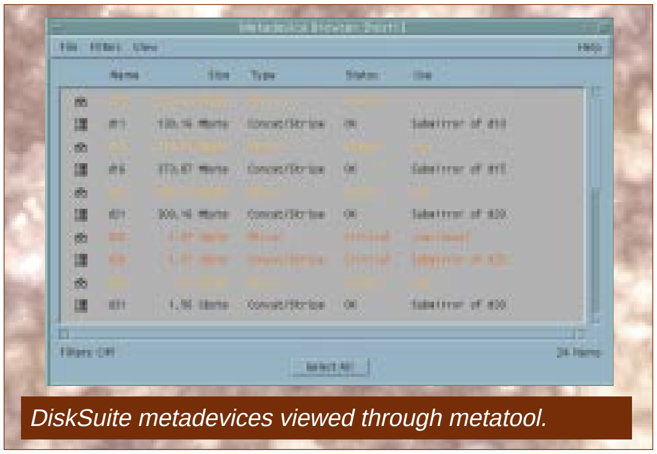
Concatenated Metadevices

Concatenated metadevices enable you to create a file system comprising multiple disk partitions that are accessed sequentially. One way to set up a metadevice is to edit the `/etc/opt/SUNWmd/md.tab` file and run `metainit`. For example, the following line in `/etc/opt/SUNWmd/md.tab` defines a concatenated metadevice, `d10`, made up of three components:

```
/dev/md/dsk/d10 3 1 /dev/dsk/c0t2d0s3 1 \
   /dev/dsk/c1t2d0s3 1 /dev/dsk/c1t4d0s0
```

The default metadevice names run from `d0` to `d127` and are located in the `/dev/md/dsk` and `/dev/md/rdsk` directories. The number `3`, in the example above, tells the system that the metadevice has three rows of components, and the number `1` means each row comprises one component. The `metainit` command is run to start using the metadevice:

```
# /usr/opt/SUNWmd/sbin/metainit d10
```



DiskSuite metadevices viewed through metatool.

In a striped metadevice, the partitions (or components) are accessed in an interlaced manner, not sequentially. The default interlace size in DiskSuite 4.0 is 16 KB for each component. Metadevices with components on different controllers offer better I/O rates than components on the same controller. In fact, striping partitions on the same disk may hurt performance.

To create a striped metadevice, `d11`, comprising three partitions, namely `/dev/dsk/c1t2d0s3`, `/dev/dsk/c2t2d0s3` and `/dev/dsk/c2t4d0s3`, with an interlace size of 32 KB, add the following line in `/etc/opt/SUNWmd/md.tab` and run `metainit` specifying `d11` to start using the metadevice:

```
/dev/md/dsk/d11  1  3  /dev/dsk/c1t2d0s3 \
   /dev/dsk/c2t2d0s3 /dev/dsk/c2t4d0s3 -i 32k
```

Metadevice `d11` has one row of three components.

Concatenated Stripes

Concatenated stripes provide better performance than simple concatenations. Also, concatenating multiple stripes provides a way to expand striped metadevices. To create a concatenation of two stripes, where each stripe has two partitions, enter the following one-line command in `/etc/opt/SUNWmd/md.tab` and run `metainit` to begin using the metadevice:

```
/dev/md/dsk/d15  2   2  /dev/dsk/c1t0d0s3 \
   /dev/dsk/c2t0d0s0 -i 32k  2 \
    /dev/dsk/c1t4d0s3 /dev/dsk/c2t4d0s0 -i 16k
```

Metadevice, `d15`, first fills up `/dev/dsk/c1t0d0s3` and `/dev/dsk/c2t0d0s0` with an interlace size of 32 KB. It then starts filling `/dev/dsk/c1t4d0s3` and `/dev/dsk/c2t4d0s0` with an interlace size of 16 KB.

There are a few caveats when creating a stripe or concatenation: First, all components must have the same disk geometry, that is, the same number of sectors per track, same number of tracks per cylinder and so on. Second, if you have components of different sizes in a stripe, the usable size of each component will be the size of the smallest one.

The performance of a stripe depends on the interlace value. If the I/O size is greater than the interlace size, data is written to, or read from, multiple disks. This improves I/O performance.

Creating and Using File Systems

By now you are probably wondering what operations can be performed on metadevices? Most of the standard UNIX utilities for file systems built on simple disk partitions are valid for file systems built on metadevices. Once you have initialized a metadevice using `metainit`, you can run `newfs` to create a new file system. For example, use the command `fsck` to check a file system and the `mount` and `umount` commands to mount and unmount. In addition, you can use the `ufsdump` and `ufsrestore` commands to back up and restore entire file systems. Metadevices may have an entry in `/etc/vfstab`:

```
/dev/md/dsk/d50  /dev/md/rdsk/d50 /app ufs 6 yes -
```

It is easy to convert a file system on a partition to a metadevice file system, and the process preserves data on the file system.

The following example converts a file system on `/dev/dsk/c1t2d0s7` mounted on `/app` to a simple metadevice `/dev/md/dsk/d20` with the above partition as its only component.

Step 1. Edit `/etc/opt/SUNWmd/md.tab` to add:

```
/dev/md/dsk/d20 1 1 /dev/dsk/c1t2d0s7
```

Step 2. If possible, unmount `/app` with `#umount /app`

Step 3. Initialize the metadevice by running `#metainit d20`. If the file system cannot be unmounted in Step 2, use `metainit -f`.

Step 4. Edit `/etc/vfstab` to change the line

```
/dev/dsk/c1t2d0s7  /dev/rdsk/c1t2d0s7  /app ufs 6 yes -
```

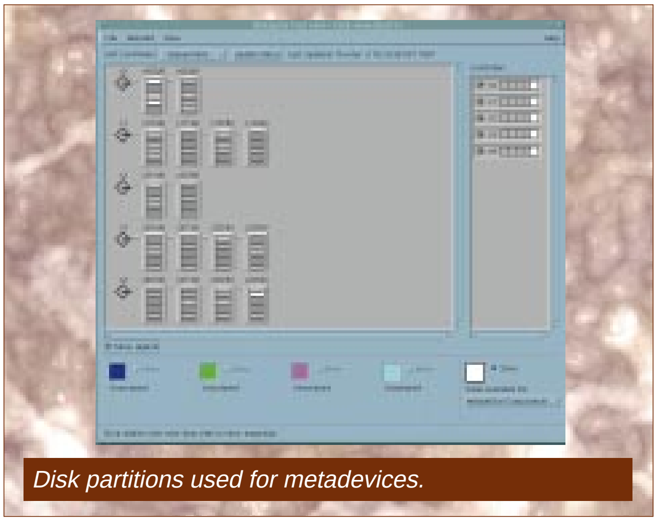
to read

```
/dev/md/dsk/d20  /dev/md/rdsk/d20  /app ufs 6 yes -
```

Note that only the first two fields have to be changed.

Step 5. If the file system was unmounted in Step 2, mount it; otherwise, reboot the machine to start using the metadevice.

Mounted or unmounted metadevice file systems can be easily expanded. For example, a device called `d8` could be concatenated with another empty partition, say, `/dev/dsk/c0t1d0s6`, by using the `metattach` and `growfs` commands. First, unmount the file system (if possible) on `d8` and edit the `/etc/opt/SUNWmd/md.tab` file to include the new partition for `d8`. Use the `metattach` command to concatenate the new partition to the end of the existing metadevice. For example,



Disk partitions used for metadevices.

```
#metattach d8 /dev/dsk/c0t1d0s6
```

Use the `growfs` command so that the file system occupies the entire metadevice. The `growfs` command will not affect data on the file system it expands. You can use the `-M` option of `growfs` if the device cannot be unmounted.

Mirroring Data

DiskSuite provides a way to replicate data by using mirrored metadevices called metamirrors. A metamirror has multiple submirrors, and each submirror contains one copy of the data. Mirroring data offers several advantages:

- Uninterrupted access to file system data despite hardware failure of the disk containing a copy of the data.
- The ability to perform online backups.
- Faster reads.

DiskSuite 4.0 allows up to three submirrors in a metamirror. Therefore, you can have up to three copies of the file system data in three different places.

The following six-step example creates a metamirror, named `d10`, with data currently on partition `/dev/dsk/c1t1d0s3` (which would become the first submirror, `d11`) and a new partition `/dev/dsk/c2t4d0s3` (which would become the second submirror, `d12`).

Step 1. If possible, unmount the partition `/dev/dsk/c1t1d0s3`.

Step 2. Edit `/etc/opt/SUNWmd/md.tab` to define `d10`, `d11` and `d12`:

```
/dev/md/dsk/d10 -m /dev/md/dsk/d11
/dev/md/dsk/d11 1 1 /dev/dsk/c1t1d0s3
/dev/md/dsk/d12 1 1 /dev/dsk/c2t4d0s3
```

Step 3. Initialize both submirrors and metamirror:

```
# metainit d12
# metainit d11
# metainit d10
```

If the file system was not unmounted, use the `-f` option of `metainit` for submirror `d11`.

Step 4. Edit `/etc/vfstab` to be able to use the new meta mirror `/dev/md/dsk/d10` instead of using the partition `/dev/dsk/c1t1d0s3`.

Step 5. If the device was not unmounted in Step 1, reboot the system; otherwise, simply mount it.

Step 6. Run `metattach` to copy data from the first sub mirror to the second:

```
#metattach d10 d12
```

The submirror `d12` must be of equal or greater size than the first submirror, `d11`, to accommodate the data in `d11`.

To add or remove copies of data, you simply attach or detach submirrors to or from a metamirror. For example, a submirror containing a copy of the data can be taken offline for backups. Before taking a submirror offline, you have to lock writes to a file system using the `lockfs` command. Then use `lockfs -i` to unlock the file system and allow writes to continue.

There are numerous read and write options you can set for metamirrors. Reads can be made in a round-robin manner from all submirrors (default); divided among submirrors based on logical disk block address (geometric reads); or all reads can be directed from one submirror. Writes to submirrors can be done serially or in parallel. Parallel is the default write mode.

RAID 5 Devices

DiskSuite 4.0 supports RAID Level 5. A RAID 5 device contains three or more physical partitions. Despite hardware failure of a single component, data on the file system will continue to be available. A single sector on any component contains either a sector's worth of data or parity information for data located on corresponding sectors on other components.

Existing components can be replaced and new ones can be easily added to a RAID device. To guard against data loss caused by disk failures, a hot spare pool can be assigned for online replacement of failed components.

To define a RAID 5 metadevice, edit `/etc/opt/SUNWmd/md.tab` and add an entry like the following:

```
/dev/md/dsk/d60 -r /dev/dsk/c0t0d0s4 \
   /dev/dsk/c1t2d0s6 /dev/dsk/c2t0d0s3 -i 32k
```

Initialize it by running `metainit`. This example puts the three specified partitions in the RAID metadevice, `d60`, with an interlace size of 32 KB for striping data and parity regions on the three components.

Hot Spare Pools

A hot spare is a partition that can replace a failed component of a submirror or RAID device. As soon as a hot spare component is brought into a submirror or RAID metadevice, data on the hot spare component is built from data on another submirror or other RAID components. There is no downtime caused by this online, automatic replacement.

Once you have decided which partitions to use as hot spares, these partitions can be grouped into one or more hot spare pools in the `/etc/opt/SUNWmd/md.tab` file. The following lines define two hot spare pools, namely `hsp001` and `hsp002`, each with two components:

```
hsp001 /dev/dsk/c0t2d0s7 /dev/dsk/c1t2d0s3
hsp002 /dev/dsk/c1t2d0s3 /dev/dsk/c0t2d0s7
```

Each hot spare pool can be initialized and associated with a submirror or RAID device. If a component fails, the system searches the hot spares in the associated hot spare pool for a partition that is of an equal or larger size than the failed component. If one is found, the submirror or RAID device copies data to it. DiskSuite offers utilities to dynamically add, delete, replace and enable hot spare components in a hot spare pool.

DiskSuite has a wide range of features for those administrators who provide users with file systems spanning several disks, mirrored file systems, RAID 5, hot spares or dynamically expandable online file systems. With it, you can offer users improved I/O rates, increased capacity and better protection against disk failure.

Kailash Jayaswal is employed by Astralray Engineering, San Jose, CA. He can be reached by email at `jayaswal@astralray.com` or by phone at (408) 871-8000.