The Cuddletech Veritas Volume Manager Series

Exploring Layered Volumes in VxVM 3.X

Ben Rockwood
CDLTK
Cuddletech

benr@cuddletech.com

VxVM 3.x adds the powerful capability to create "layered volumes", the ability to create a volume inside a volume. Sadly, practical coverage of creating such volumes has been sparse. In this tutorial we discuse the topic at length and walk thru several examples of creating layered volumes from start to nish with a focus on RAID1+0.

Introduction

A greatly powerful, and often requested, function of Veritas Volume Manager if often the least well understood. This being the usage of Layered Volumes. Working with RAID0+1 (Stripping plus Mirroring), RAID0 (Stripping), and RAID5 are well understood concepts for most. But when it comes to the popular alternative of RAID1+0 (Mirroring plus Stripping) we nd a knowledge rift due to poor documentation on the part of Veritas. Well, in this document we're going to x that. But rst, what is RAID1+0?

In a "typical" RAID0+1 volume con guration we take several disks and then create a stripe across thoughs disks (the RAID0 part). Then once complete we do this again on a separate set of disks, and then attach that new stripe to the rst creating a mirror (the +1 part). We then have a RAID0+1 volume thats ready to have a lesystem put on it. The point of interest with this setup is that we're actually mirroring a complete stripe (and therefore ALL the disks in that stripe) to another stripe (and therefore ALL of it's disks). The problem here is that if for some reason we need to re-sync the volume we'd need to re-sync a full stripe to a full stripe (very timely) which is a nearly tragic proposition if your talking about 50G+. A far more ef cient setup would be to mirror each disk to each disk... in other words, to mirror a bunch of disks on a one-to-one basis, and then build a stripe on top of these mirrors. In this case if we need to re-sync due to a disk failure we can simply sync the failed disk to its mirror, instead of the full stripe. This is the power of RAID1+0; the difference between mirroring the stripes (0+1) and stripping the mirrors (1+0).

If the terms seem to confuse you, try this for size:

Bingo! Done! Just create your file system and your ready to go! You can see from the step through that really your are just creating a bunch of volumes, telling Veritas Volume Manager to just think of the volumes as subdisk objects (via vxedit), and then building another volume with the subvolumes as subdisks. The only trick along the way is to make sure to start the subvolumes before you you convert

root@sun7:/usr/sbin# vxassist -g testdg convert stripeprovol layout=mirror-stripe

Pretty easy line huh? We're simply telling Veritas Volume Manager that we want it convert (actually, its a relayout) our "stripe-mirror" to a "mirror-stripe". The beauty of this is that there is NO CONVERSION TIME! Because the process is a simple change in the way that VxVM looks at the disks rather than the way that it positions the data there is no conversion time. Lets look at what our earlier built volume looks like now that we've converted it (vxprint -hrt):

```
v  stripeprovol -            ENABLED  ACTIVE   2097152  fsgen     -        SE-
LECT
pl stripeprovol-01 stripeprovol ENABLED ACTIVE 2097152  STRIPE    2/128    RW
sd testdg02-01  stripeprovol-01 testdg02 0     1048576  0/0       c1t3d0   ENA
sd testdg03-01  stripeprovol-01 testdg03 0     1048576  1/0       c1t3d1   ENA
pl stripeprovol-02 stripeprovol ENABLED ACTIVE 2097152  STRIPE    2/128    RW
sd testdg04-01  stripeprovol-02 testdg04 0     1048576  0/0       c1t3d2   ENA
sd testdg01-01  stripeprovol-02 testdg01 0     1048576  1/0       c1t3d30  ENA
```

Look alittle less intimidating? It should. Using vxassist we can convert back and forth seamlessly between stripe-mirror and mirror-stripe. The same can be done with concat-mirror and mirror-concat. The only catch is that you can't use "vxassist convert" to convert from stripe-mirror to concat-mirror (or any other concat to stripe combination), for that you'd need to do a full-blown vxrelayout.

I should also note that the vxassist conversion will work on any properly formed volume. This means that you can build a volume by hand (vxmake) as a RAID0+1 and then convert it to the RAID1+0. Or you could create a volume using vxassist as RAID1+0 and then convert it, work on it (tear apart the plexes, whatever) and then when finished just convert it back! This make everything really simple to work with.

And just to round this out, lets add take one last look at things by adding a Dirty Region Log to our volume:

```
root@sun7:/usr/sbin# vxassist -g testdg addlog stripeprovol

(vxprint -hrt)
v  stripeprovol -            ENABLED  ACTIVE   2097152  fsgen     -        SELECT
pl stripeprovol-01 stripeprovol ENABLED ACTIVE 2097152  STRIPE    2/128    RW
sd testdg02-01  stripeprovol-01 testdg02 0     1048576  0/0       c1t3d0   ENA
sd testdg03-01  stripeprovol-01 testdg03 0     1048576  1/0       c1t3d1   ENA
pl stripeprovol-02 stripeprovol ENABLED ACTIVE 2097152  STRIPE    2/128    RW
sd testdg04-01  stripeprovol-02 testdg04 0     1048576  0/0       c1t3d2   ENA
sd testdg01-01  stripeprovol-02 testdg01 0     1048576  1/0       c1t3d30  ENA
pl stripeprovol-03 stripeprovol ENABLED ACTIVE LOGONLY  CONCAT    -        RW
sd testdg02-02  stripeprovol-03 testdg02 1048576 33     LOG       c1t3d0   ENA
```

Now, let's convert out newly logged volume back into a stripe-mirror (StripePro):

```
root@sun7:/usr/sbin# vxassist -g testdg convert stripeprovol layout=stripe-mirror

(vxprint -hrt):
v  stripeprovol -            ENABLED  ACTIVE   2097152  fsgen     stripeprovol-
04 SELECT
pl stripeprovol-04 stripeprovol ENABLED ACTIVE 2097152  STRIPE    2/128    RW
sv stripeprovol-S01 stripeprovol-04 stripeprovol-L01 1 1048576 0/0 3/3     ENA
v2 stripeprovol-L01 -        ENABLED  ACTIVE   1048576  fsgen     -        SELECT
p2 stripeprovol-P01 stripeprovol-L01 ENABLED ACTIVE LOGONLY CONCAT -       RW
```

```
s2 testdg06-01  stripeprovol-P01 testdg06 0    33       LOG       c1t3d3   ENA
p2 stripeprovol-P02 stripeprovol-L01 ENABLED ACTIVE 1048576 CONCAT -       RW
s2 testdg02-03  stripeprovol-P02 testdg02 0    1048576  0         c1t3d0   ENA
p2 stripeprovol-P03 stripeprovol-L01 ENABLED ACTIVE 1048576 CONCAT -       RW
s2 testdg04-02  stripeprovol-P03 testdg04 0    1048576  0         c1t3d2   ENA
sv stripeprovol-S02 stripeprovol-04 stripeprovol-L02 1 1048576 1/0 3/3     ENA
v2 stripeprovol-L02 -        ENABLED  ACTIVE   1048576  fsgen     -        SELECT
p2 stripeprovol-P04 stripeprovol-L02 ENABLED ACTIVE LOGONLY CONCAT -       RW
s2 testdg06-02  stripeprovol-P04 testdg06 33   33       LOG       c1t3d3   ENA
p2 stripeprovol-P05 stripeprovol-L02 ENABLED ACTIVE 1048576 CONCAT -       RW
s2 testdg03-02  stripeprovol-P05 testdg03 0    1048576  0         c1t3d1   ENA
p2 stripeprovol-P06 stripeprovol-L02 ENABLED ACTIVE 1048576 CONCAT -       RW
s2 testdg01-02  stripeprovol-P06 testdg01 0    1048576  0         c1t3d30  ENA
```

Interesting, huh? Now lets convert back just for fun:

```
root@sun7:/usr/sbin# vxassist -g testdg convert stripeprovol layout=mirror-stripe

(vxprint -hrt):
v  stripeprovol -            ENABLED  ACTIVE   2097152  fsgen     -        SELECT
pl stripeprovol-01 stripeprovol ENABLED ACTIVE 2097152  STRIPE    2/128    RW
sd testdg02-04  stripeprovol-01 testdg02 0     1048576  0/0       c1t3d0   ENA
sd testdg03-01  stripeprovol-01 testdg03 0     1048576  1/0       c1t3d1   ENA
pl stripeprovol-02 stripeprovol ENABLED ACTIVE 2097152  STRIPE    2/128    RW
sd testdg04-01  stripeprovol-02 testdg04 0     1048576  0/0       c1t3d2   ENA
sd testdg01-01  stripeprovol-02 testdg01 0     1048576  1/0       c1t3d30  ENA
pl stripeprovol-04 stripeprovol ENABLED ACTIVE LOGONLY  CONCAT    -        RW
sd testdg06-01  stripeprovol-04 testdg06 0     33       LOG       c1t3d3   ENA
```

Notice, how the log (the same log) gets seeing in both subvolumes. This goes against all sense, but there it is. This is one of thoughs questions to save for the Veritas Engineers while waiting for phone support. But you can see how easily everything works together.

The Wrap Up

I hope that this tutorial has shown you both the power and the ease of implementing layered volumes under Veritas Volume Manager. This tutorial very well could have been merely a lesson in the usage of the "vxassist convert" tool, but hopefully by looking at the structure of the volumes and going through a vxmake of a volume you can get a solid grasp of the concepts behind volumes.

A precaution is in order (yes another one), earlier I had said that vxassist made conversion of properly formed volumes very easy. Notice closely that I said "properly formed volumes". Several cases have popped up on various mailing lists over the last couple months from people who had significant trouble with conversion to and from layered volumes because, for one reason or another, something got changed so that the volume was no longer proper and the volume was stuck in one state or another. Take, for example, a situation where you have a StripePro volume and a disk fails and you replace it. In this example you might end up replacing the disk and then accidental making one column longer than another; the volume is still in tact, and technically everything should still jive, but you eventually decide to convert back to mirror-stripe and the operation fails due to inconstancy. This is an example of things that can and do happen. Great care should be taken when dealing with layered volumes. Let me further

supplement this by saying that many admins don't consider layered vols ready for prime time. Even in the latest Release Notes for VxVM 3.1.1 there are still bugs and inconsistencies that make it too unreliable. Here are two two "Veritas Incident Numbers" found on pages 19 and 20 of the VXVM 3.1.1 Release Notes:

```
45668 Due to the current implementation to handle the resize of layered
 volumes, it is recommended not to grow or shrink layered volumes
 (stripe-mirror, concat-mirror, and so on) while resynchronization
 is going on.
 Internally, the Volume Manager converts the layout of layered volumes
 and updates the configuration database before it shrinks or grows their
 sizes.  This causes any ongoing operation, such as the resynchronization,
 to fail.
 [snip]
 Although this release supports layered volumes, creating volumes
 with mixed layout types is not recommended.  For example, adding a
 mirror use the vxassist mirror command specifying "layout=mirror-stripe
 nmirror=1" to an existing volume with stripe-mirror layout is not
 recommended. (Se Sun BugID 4339626)
none While doing relayout on a mirrored volume the vxassist command
 keeps the volume as mirrored even if the layout attribute is specified as
 stripe or nomirror.  For example, see the following commands:
  # vxassist make vol 1024 layout=mirror-stripe ncol=3
  # vxassist relayout vol layout=stripe ncol=2
 The volume vol is converted to a 2 column volume, but it is
 still mirrored even if the layout attribute is specified as stripe
 and nomirror.
```

Both the above incidents might seem like no-brainers, but these are real problems that any of us can run into. As a general rule, any time you make changes to the volume you should consider converting back to RAID0+1, you'd be better off stuck in 0+1 than in 1+0. As with anything in Veritas Volume Manager (Veritas products in general really) you need to be very careful and look at things from every possible angle before making changes. When real data is on the line the amount of time it takes to think and rethink things through is nothing compared to the time it'll take to restore if you make a mistake.

```
For More Information See:
 - The Veritas Volume Manager Documentation (Included with Product)
 - The Veritas Volume Manager Support Site (www.veritas.com, click support)

For Help contact:
 - The Veritas-Vx Mailing List
 (http://mailman.eng.auburn.edu/mailman/listinfo/veritas-vx)
```


```
Special Thanx to:
 - Doug Hughes (doug@Eng.Auburn.EDU), one of the nets greatest Vx gurus,
 for giving the tutorial a final look over and sharing his thoughts.
```

Warning

Disclaimer: This documentation is provided without warranty of any kind. If you blast your data due to something found in this tutorial its not my or Cuddletech's fault. All methods found here should be tested in a sandbox environment where you can experiment and become comfortable BEFORE being used on real data or in production environments. Veritas Corp had nothing to do creation of this doc and has no connections of any kind to Cuddletech.com.