Online Backups
Using the VxVM Snapshot Facility

By John S. Howard - Enterprise Engineering

Sun BluePrints™ OnLine - September 2000

http://www.sun.com/blueprints
Introduction

Complete and accurate backups are crucial to every datacenter. However, the uptime and availability requirements of a datacenter are often opposed to the typical backup needs of making disk volumes quiescent for the minutes or hours necessary to perform a backup.

By using the snapshot facility of the Veritas Volume Manager software we can effect an online backup of disk volumes. This online backup still requires the volume to be made quiescent. However, the snapshot facility requires the volume to be quiescent for only seconds or minutes rather than hours.

The Snapshot Procedure

The procedure overview for performing an online backup is as follows:

1. Create and attach a mirror to be used for the snapshot.

   As this step is attaching a mirror to an existing volume, a data synchronization to the snapshot mirror will be performed. Depending on the size of the volume and system load, this synchronization may be very consumptive of time and system resources, so it is recommended that this step be performed during off-peak system utilization. The snapshot mirror is a true mirror, for as long as the snapshot mirror is attached any changes or updates to the volume will automatically be made to the snapshot mirror as well.
2. When you are ready to perform the online backup, the snapshot mirror is broken off and created as a volume.

   This step is typically very brief and the volume should be made as quiescent as possible to minimize the possibility of data inconsistency on the snapshot volume.

3. The consistency of the snapshot volume is verified.

4. The backup of the snapshot volume is performed and any clean up is done.

   In the example that follows, version 8 of the Solaris™ Operating Environment and version 3.0.4 of the Veritas Volume Manager (VxVM) software are used. It is also assumed that VxVM has been correctly installed and root’s PATH has been correctly set. We will be demonstrating the snapshot procedure on the u01 volume of the drepdg diskgroup. The u01 volume is a 6GB RAID1+0 volume containing a ufs filesystem that is mounted on /u01. In addition, we will be adding disk c5t1d0 to the drepdg diskgroup to be used to hold the snapshot volume.

   The detailed procedure is as follows:

1. Initialize the snapshot disk and add it to the drepdg diskgroup with the name of snapdisk.

2. Create and attach the snapshot mirror on snapdisk:

   ```
   racex# vxdisksetup -i c5t1d0
   racex# vxdg -g drepdg adddisk snapdisk k=c5t1d0
   racex# vxassist -g drepdg snapstart u01 \alloc="snapdisk"
   ```

   When the vxassist snapstart is completed (when the mirror synchronization is complete), the snapshot mirror will be put in a SNAPDONE state. For example:

   ```
   racex# vxprint -ht -g drepdg u01-03
   PL NAME VOLUME KSTATE STATE LENGTH LAYOUT NCOL/ WID MODE
   SD NAME PLEX DISK DISKOFFS LENGTH [COL/]OFF DEVICE
   MODE
   SV NAME PLEX VOLNAME NVOLLAYR LENGTH [COL/]OFF AM/NM
   MODE
   pl u01-03 u01 ENABLED SNAPDONE 12585752 CONCAT - WO
   sd snapdisk-02 u01-03 snapdisk 12585752 12585752 0 c5t1d0
   ENA
   ```
The time to synchronize the snapshot mirror to the volume is dependant on the size of the volume, system load, and system resources such as number of CPUs and amount of RAM. For the 6GB u01 volume on a moderately loaded 4 CPU Ultra Enterprise™ 420 used in this example, the snapstart step took approximately 10 minutes.

3. Make the u01 volume as quiescent as possible, then break the snapshot mirror off of the u01 volume and create a temporary snapshot volume named snapshotu01:

```
racerx# /usr/sbin/vxassist -g drepdg snapshot u01 snapshotu01
```

The time to break off the snapshot mirror and create it’s associated temporary volume is typically very brief, but the length of time is dependant upon the size and number of IO operations pending to the volume (essentially, the size and number of updates that need to be flushed to the mirror before it is detached). For the 6GB u01 volume we are using in our example, the snapshot step took an average of 0.6 seconds to complete.

The vxassist command creates the temporary volume as specified, in this example the temporary volume is named snapshotu01. The temporary volume is accessible and may be manipulated as any other VxVM volume. However, given that The temporary volume contains a point in time snapshot of our data, it is recommended to only use the snapshot volume as described in this article.

As with any other VxVM volume, access from the Solaris Operating Environment to the temporary volume is done through the /dev device tree. Given our example, a snapshot volume name of snapshotu01 in the diskgroup drepdg, the raw device path to the temporary volume is /dev/vx/rdsk/drepdg/snapshotu01.

4. Next, verify the consistency of the snapshot volume with the appropriate tool.
Because the u01 volume was a ufs filesystem, we will use fsck to verify the filesystem’s consistency:

```
racerx# fsck /dev/vx/rdsk/drepdg/snapshotu01
** /dev/vx/rdsk/drepdg/snapshotu01
** Last Mounted on /u01
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
5 files, 4196385 used, 1996269 free (13 frags, 249532 blocks, 0.0% fragmentation)
```
5. Perform the backup.

Here ufsdump is used for the sake of example. The backup utility in use at your
datacenter (i.e., Solaris Backup Utility or Veritas NetBackup) should be used to back
the snapshot volume up to tape:

```
racerx# ufsdump 0uf /dev/rmt/0 /dev/vx/dsk/drepdg/snapshotu01
DUMP: Writing 32 Kilobyte records
DUMP: Date of this level 0 dump: Tue 01 Aug 2000 02:35:10 PM PDT
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/vx/rdsk/drepdg/snapshotu01 to /dev/null.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 8405470 blocks (4104.23MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 8405438 blocks (4104.22MB) on 1 volume at 24469 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Tue 01 Aug 2000 02:35:10 PM PDT
```

If your backup utility has the capability of verifying the quality and consistency of
the data written to tape, that verification should be performed now.

6. Finally, destroy the temporary snapshot volume:

```
racerx# /usr/sbin/vxedit -rf rm snapshotu01
```

The c5t1d0 disk may now be used to snapshot another volume in this diskgroup by
starting over with the vxassist snapstart command. Alternatively, the c5t1d0
disk may be removed from the drepdg diskgroup and added to another diskgroup
to be used for snapshots within that diskgroup.

Comments

As with all backup procedures, this procedure should be thoroughly tested on a
regular basis. In addition to scheduled testing of the backup procedures, scheduled
testing of the restoration procedures is also essential.

Regular backups of the root “/” filesystem are often neglected or overlooked
because of the inability to take the entire system down to get a clean, consistent
backup of /. It is useful to note that this procedure may also be used to perform a
backup of a VxVM encapsulated boot disk during off-peak system usage.

4 Online Backups Using the VxVM Snapshot Facility • September 2000
Conclusion

This article has presented a procedure to effect an online backup of a disk volume using the snapshot facility of Veritas’ Volume Manager software. Complete and accurate backups performed in a timely fashion are crucial to every datacenter. This article has presented a procedure utilizing the snapshot facility of the Veritas Volume Manager software which enables the System Administrator to perform timely, complete and accurate online backups with minimal impact to the user or applications.

Author’s Bio: John S. Howard

John S. Howard is currently a Staff Engineer in the Enterprise Engineering group at Sun Microsystems in San Diego, California. He has worked as a software engineer and systems administrator for the past 19 years. Prior to Enterprise Engineering, John worked in Enterprise Services as an Area System Support Engineer for five years. As an ASSE, he was responsible for developing and performing Reliability, Accessibility, and Serviceability (RAS) studies of customer datacenters and the development of proactive Enterprise RAS Services. Prior to Sun, John held engineering positions at: The Chicago Board of Trade Clearing Corporation, Datalogics Inc, and Rand McNally. Throughout his career he has developed: pagination and publishing software, loose-leaf publishing systems, extensive SGML systems development, database publishing systems, text editors and WYSIWIG systems, and device drivers.