Subject: MCR-10048, Add Volume Pool Support to Hierarchy Backups

Author: Eric Swenson

Date: August 18, 2018

Introduction

Volume backups support a notion of a volume pool, where tape volumes can be added and tape volumes allocated from this pool when new tapes are needed. Hierarchy backups, on the other hand, do not support this and require operators to keep track of tapes used for hierarchy backups.

This MCR proposes adding volume pool support to the hierarchy backup commands, such that if a volume pool is configured during a hierarchy backup (incremental, catchup, or complete) tapes will be allocated from this pool, rather than being prompted for. For sites that run 24x7 and require hierarchy backups, the current manual intervention required to supply tape names and keep track of the "next" tape is cumbersome. Adding volume pool support, such as already standard with the volume backup daemons, to hierarchy backup daemons, will greatly ease the burden for site operators and administrative staff.

The volume backup system requires a volume pool. An administer creates the volume pool (with the manage_volume_pool (mvp) command and adds tape volumes to this pool. When the volume backup daemons need a tape, they attempt to allocate a tape from the pool. If no free volume is available from the pool, the volume backup daemons prompt the operator for a volume name.

With volume backup, the liveness of tapes is maintained automatically. When a set of tapes is used during incremental volume backup, a subset of those tapes are automatically freed when the volume dump does a consolidate volume backup (because the segments on those tapes are also backed up in consolidated volume backup tapes). The volume backup daemons know how to "free" those tapes for use. Similarly, when a complete volume backup is performed, the volume backup can automatically "free" tapes in incremental and consolidated volume backups when all date is known to have been backed up in the complete volume backup tapes. Thus, the volume backup subsystem manages the lifecycle of incremental, consolidated, and complete volume backup tapes automatically.

No such support exists for the hierarchy backup daemons. What has been dumped to incremental, catchup (term used for hierarchy backups that corresponds to "consolidated" term used in volume backups), and complete dump tapes is not managed by the hierarchy backup software, and must be manually managed by operators and system maintenance personnel. With the hierarchy backup daemons, the operator or system administrators are forced to keep track of the available volumes by hand. This is error prone and cumbersome. But adding support for keeping track of hierarchy backup tapes is a large effort. This MCR does not propose to add this support.

In the case of volume backups, incremental, consolidated, and complete volume backups all use the same volume pool. The volume pool is stored in the segment >udd>Daemon>Volume_Dumper.volumes. As previously implied, incremental tapes are

automatically freed when they are no longer needed. The same is true for consolidated backup tapes. This allows automatic reuse of tapes that are no longer needed.

This MCR proposes adding a very simple extension to the current hierarchy backup system. When an incremental, catchup, or complete dump is initiated, a new control argument (-volume_pool, -vp) will enable the specification of a volume pool (same kind of pool as already supported by the manage_volume_pool (mvp) command and identical to the kind of volume pool already supported by the volume backup daemons).

The -volume_pool (-vp) control argument will be optional. If not specified, the current behavior of prompting for tape volumes will continue to occur. When specified, the next required parameter is a volume pool path, which must exist. When the hierarchy backup daemons require a new tape, if the -volume_pool was specified on the command line, and if there are any free tapes in the volume pool (as determined by manage_volume_pool_$allocate), then that tape is used rather than prompting the operator for a tape. If no free volume in the pool is found, the operator is queried for a tape volume, as if no -volume_pool control argument had been supplied.

Because incremental and catchup hierarchy backups are performed by Backup.SysDaemon, and complete hierarchy backups are performed by Dumper.SysDaemon, and because each type of hierarchy backup is invoked by a different operator command (“x inc”, “x cat”, or “x comp”) it seems reasonable that the new volume pool support require a parameter on the hierarchy backup command invocation command line to specify the appropriate volume pool path. This allows the site administrator to decide whether to use different pools for incremental, catchup, and complete backups, or whether to use a common pool. Consequently, no default path will be used — the operator (or >sc1>admin.ec) will be required to specify a pool path. The site administrator can customize >sc1>admin.ec as needed to specify an appropriate volume pool.

The author did consider a more complex system, where the backup invocation command line could specify a prefix to discriminate among volume names within a common volume pool, but seems overly complicated. Allowing the site administrator to use a common pool or separate pools seems like a more flexible solution. If a site administrator wants to segregate the volume (names) among incremental, catchup, and complete hierarchy backups, different pools can be used. If a common naming convention is preferred, a common volume pool can be specified.

Proposed Changes

This MCR proposes to add two new control arguments to backup_dump and related commands that initiate incremental, catchup, and complete hierarchy backups. These control arguments are:

-volume_pool (-vp)

This control argument requires a following argument — the pathname to a pre-existing volume pool. Such a pool may be created with the manage_volume_pool (mvp) command. The control argument processing uses the manage_volume_pool_$set_pool_path entry to attempt to establish a volume pool. If this is not successful (for any reason) an error is reported to the operator and the hierarchy backup attempt is aborted.

-no_volume_pool (-no_vp)

This control argument is provided to nullify a previous -volume_pool (-vp) control argument on the command line. Also, in keeping with the current practice and function of the hierarchy backup commands, since any option (such as -volume_pool) is kept in per-process static storage, the option is considered “sticky”. Subsequent invocation of the hierarchy backup commands in the process will continue to use any previously-provided -volume_pool control argument. Thus, the -no_volume_pool parameter is required to disable this feature. This supports the existing convention in the hierarchy backup commands of remembering options and overriding those options, as desired, in subsequent command lines.

With the new control argument in place, the actual choice of a volume to use in an incremental backup is made by the entry bk_output. Normally, this routine prompts for a tape and requires operator intervention to provide a tape volume. This MCR proposes to change the behavior of bk_output such that if a volume pool has been previously specified (via the -volume_pool control argument), then rather than prompt for the tape volume, the code will automatically attempt to allocate a tape volume from the specified volume pool. It does this via a call to manager_volume_pool_$allocate. If this call is successful, then the allocated tape volume will be used, and no prompt will occur. If the call is not successful (such as when the pool contains no free tapes, or any other error occurs accessing the pool), a message is displayed to the operator and the tape volume is prompted for.

If, after an unsuccessful attempt to allocate a tape from the volume pool occurs, an administrator or operator adds new tapes to the volume pool, subsequent attempts to acquire new tapes will again use the pool, and only resort to prompting if this is unsuccessful.

As mentioned previously, if different tape (names) are desired for incremental, catchup, and complete hierarchy backups, then the site can establish a different volume pool for each and populate those pools with appropriately-named tape volumes. If, on the other hand, the site desires to have one pool for incremental, catchup, and complete hierarchy backup volumes, this is also possible — by specifying a common volume pool path for all hierarchy backup daemons.

The proposed functionality, along with site operators keeping the volume pool populated with free tapes will allow 24x7 operation of Multics — including the running of hierarchy backups.

Adding the volume pool functionality to hierarchy backups does not, on its own, allow completely unattended backups on emulated systems. This is because tapes require “authentication” by the operator (to prevent overwriting tapes) if the label on those tapes doesn’t match the requested tapes. In the emulated system, new tapes are blank and have no label. When such blank tapes are mounted, operator authentication is required.

This means that even with the new -volume_pool support, an operator may still have to authenticate a tape volume when automatically requested by the hierarchy backup daemons. However, even with tape authentication enabled (by default), the proposed changes improve the tape volume management situation. It is easy to authenticate tapes (operator enters “x auth tapX_NN ***”, for example), but much harder to remember what the correct next tape should be, and ensuring that the wrong tape name is not entered accidentally.

Tape authentication can be enabled/disabled/configured through system installation parameters. The use of RCPRM can also mitigate the situation, because only those tapes registered with RCPRM and accessible by the hierarchy backup daemons can be requested for mounting. With RCPRM enabled and appropriate safeguards in place, tape authentication can

be disabled, if necessary, thus not being required at all and allowing fully operator-less hierarchy tape backups. Another way to support completely unattended hierarchy backups is to pre-label tapes before adding them to the volume pool. In this case, the label will match the tape volume name, and no authentication will be required.

Note that this tape authentication issue applies to volume dumping as well as hierarchy dumping. However, having the tape volume name automatically chosen by the system is a significant advantage — even if tape authentication is required. There have been countless times when the author, prior to having this volume pool feature for hierarchy backups, has accidentally entered the wrong tape volume name — usually an already written-to and still useful backup tape. The volume pool will prevent such mistakes.

This MCR does not propose changing the >t>admin.ec exec_com to use the -volume_pool control argument for the “x inc”, “x cat”, and “x comp” commands. Documentation will be provided in the release notes on how to make such a change if a site administrator desires to and the new control arguments will be documented in the appropriate info segments.

Alternatives Explored

We considered allowing the -volume_pool (-vp) control argument not require a path parameter, but, as an option, if no path is specified, use a “default” pool path. This would likely be the UserId.volumes segment in the home directory of the logged-in user (Backup or Dumper). However, we decided that it is better to not provide such a default and require explicit paths to be specified by the site. This allows explicit control and no accidental use of undesired volume pools. The fact that incremental and catchup hierarchy backups are both performed by Backup.SysDaemon helped support the notion that using a default volume pool might “do the wrong thing” in some cases.

We also explored the idea of allowing a single volume pool to be used, but allowing a volume name prefix to be specified on the various hierarchy backup daemon command lines. The idea would be that only tape volumes matching the specified prefix would be used from the volume pool. We discarded this approach as too complicated. It is more general to allow different pools (or a common pool) to be used. If different pools are used, then differently-prefixed volume names can be added to the respective pools. If a common volume name convention is desired, a common pool can be used.

Implementation

The undocumented manage_volume_pool_$set_pool_path and manage_volume_pool_$allocate entries provide the necessary support for the features proposed in this MCR.

bk_ss_$ maintains all the per-process static state of the hierarchy backup daemons. It is shared by all hierarchy backup commands. It will be updated to add a new ptr field — volume_pool_ptr. This will be set by bk_ss_.cds to null and thus the default for all hierarchy backup commands will be to not use a volume pool.

A common command line parameter parsing routing is used by all hierarchy backup commands. It is bk_arg_reader_.pl1. This argument parser will be augmented to support the two new control arguments specified earlier — -volume_pool and -no_volume_pool. The first

of these will attempt to invoke the manage_volume_pool_$set_pool_path entry on the specified volume pool segment name. If this is successful, the pointer to the volume pool will be stored in bk_ss_$volume_pool_ptr. If this is not successful, an error message will be displayed to the user/operator and the command aborted.

bk_output controls what tapes are chosen and the writing of incremental backup information to those tapes. In the current implementation, when a tape volume is filled, the operator (or user) is prompted for the next tape. This MCR proposes modifying that behavior to allocate a tape from the volume pool if one was specified when the backup command was invoked. By default, bk_output continue to prompt for tape volumes. It uses manage_volume_pool_$allocate to allocate a tape from the volume pool if -volume_pool was specified on the command line. If this fails (no free tape is available or there is an issue with the volume pool), then the normal prompting of a tape volume is used.

Again, no automatic freeing of tape volumes in the volume pool will be supported by the hierarchy backup commands. This function is left to the site administrator to perform manually with the manage_volume_pool (mvp) command.

The following two undocumented manage_volume_pool_ entries will be used in the proposed implementation:

```
dcl     manage_volume_pool_$set_pool_path
        entry (entry options (variable), char (*), ptr, fixed bin (35));
dcl     manage_volume_pool_$allocate
        entry (ptr, entry options (variable), char (*), char (*), char (*), fixed bin (35));
```

The following is the bk_output logic:

```
    if bk_ss_$volume_pool_ptr = null () then
               call query_for_tape(type, label, Squit_the_dump);
          else do;
               requested = "*";
               comment = bk_ss_$myname;
               volname = "";
               call manage_volume_pool_$allocate(bk_ss_$volume_pool_ptr, error_rnt, requested,
comment, volname, code);
               if code ^= 0 then do;
                    call com_err_(code, "bk_output", "Unable to allocate tape from volume pool");
                   call query_for_tape(type, label, Squit_the_dump);
                    end;
               else
                    call ioa_ ("^a: Allocated tape ^a from volume pool", bk_ss_$myname, volname);

              label = volname;
          end;
```

Operation

With the current proposal, a site administrator desiring to use volume pools with the hierarchy backup daemons would first use the manage_volume_pool (mvp) command to create one or more pools (shared among hierarchy backup daemons or distinct subsets of the daemons). Then, the administrator would populate the volume pool with tape volume names. Finally, the hierarchy backup daemons would be invoked, specifying the -volume_pool control argument and the path to the already-created volume pool segment.

While this MCR is not proposing to make any changes to the admin exec_com (>t>admin.ec or, as installed, >sc1>admin.ec) a site administrator can choose to update the local copy of

>sc1>admin.ec to alter the “x inc”, “x cat”, and “x comp” commands to include the -volume_pool control argument in the command invocations of the various commands used for incremental, catchup, and complete hierarchy backups.

The hierarchy backup daemons will prompt for tapes when there are no free volumes in the appropriate volume pool. Site administrators can “keep ahead of the game” by examining the volume pools and making sure free volumes are not running low (see manage_volume_pool documentation for listing and updating volume pools). As long as free volumes are available, the hierarchy backup daemons will not prompt for tapes, but use tapes from the volume pools.

The site administrator can manually mark tapes in the pool as “free” when it is known that those tapes are no longer needed. In so doing, the site can limit the tapes in the pool. For example, when a complete hierarchy backup is done, the incremental and catchup tapes subsumed by the complete backup can be considered no longer useful and can be marked “free” in the volume pool (manually by the administrator).

Manage Volume Pool Issues

While examining the source for the undocumented manage_volume_pool_$ entries, a race condition was discovered in the setting up of a cleanup handler. Each subroutine entry point (as well as the main command entry point for the manage_volume_pool command) invoked a setup() procedure and afterwards established a cleanup handler, which invoked the finish() procedure. However, faults occurring in the setup() function, followed by stack cleanup (such as when the “release” command is executed) would leave state not-cleaned-up. We decided to fix these issues by splitting up the setup() procedure into two parts (setup1 and setup2) where setup1 would be called first, followed by the establishment of the cleanup handler, followed by setup2.

Testing of the Change

The changes described above to the hierarchy backup software, with the exception of the cleanup handler-related fixes in manage_volume_pool_ have been running for several months on GHM. This site runs 24x7 and runs incremental hierarchy (and volume) backups constantly. The site administrator (the author of this MCR) also uses the backup_dump command regularly. There have been no issues with the changes during this time.

The new manage_volume_pool_ changes will be put into place soon and the final software will be used for some time before the next Multics release (12.6g) is created. The author intends to test out the manage_volume_pool command, as well as the hierarchy backup commands with both the -volume_path, -no_volume_path, and neither option present to make sure no issues arise.

Testing will also include having no volume pool segment created, having no free volumes in the pool, as well as the normal situation where the pool is populated. This condition has been tested for several months now already.

Bug Reference

• URL of Multics Change Ticket: http://multics-trac.swenson.org/ticket/134

Documentation

Info segments for the backup_dump, start_dump, catchup_dump, and complete_dump commands will be updated to include documentation on the -volume_pool (-vp) and -no_volume_pool (-no_vp) control arguments.

The release notes for the next release of Multics will include a description of the new functionality.

If I get inspired, I'll create errata for the GB64 (Multics Administration, Maintenance, and Operations Commands), AM81 (System Maintenance Procedures), and GB61 (Operator's Guide) manuals that include appropriate material for the new functionality. GB64 documents the backup_dump and related hierarchy backup commands. AM81 is where hierarchy backups are discussed in detail. GB61provides some documentation for operators to initiate hierarchy backups.

Date	Revision	Author	Comment
2018-08-18	1.0	Eric Swenson	Initial version of MCR.