Experiences with the DMF Advanced Tape MSP

Birgit Hellwich and *Mathilde Romberg,* Zentralinstitut fuer Angewandte Mathematik (ZAM), Forschungszentrum Juelich GmbH (KFA), Juelich, Germany

ABSTRACT: *KFA started in May 95 to use the advanced tape MSP under DMF 2.2.6. Approximately 200,000 files with about 500 GB of data were moved from the old MSP to the new one. Practical experiences gained within the process are given and critical steps are described in this paper.*

Introduction

KFA is one of 16 German national research laboratories. Its main fields are structure of matter and materials research, information technology, life science, environmental precaution research, and energy technology. Among other tasks, the Central Institute for Applied Mathematics (ZAM) manages KFA's central computer systems and communication networks. The facilities comprise an IBM ES/9000, an Intel Paragon XP/S 10, an IBM SP2, and two Cray Y-MPs as well as FDDI, HiPPI, ATM, and other networks. The Cray systems are a Y-MP8/864 and a Y-MP M94/4512 each dedicated to a different group of users.

KFA is using the data migration facility since 1989 to primarily enlarge the limited disk space. On each of the two Cray systems DMF is used on one file system, /arch, where users should store their large datasets. Because of limited tape capacity available for the Crays at that time (part of a STK silo with 200MB 3480 cartridges) we chose to have only one copy of a dataset residing in DMF. This was at first implemented through a local modification, later - after its introduction - the configuration parameter DEFCOP was used. In addition disk quotas were established on the file systems counting online and offline data. This is still implemented through a local modification. The data migration facility is a reliable product. During all those years we only lost data from two tapes due to media problems.

KFA is a relatively small DMF site with about 500 GByte of data altogether on both machines which reside on 960 3490E tapes (800MByte capacity) managed by *dmtpmsp* (status as of May 1st). The available disk space for the /arch file system is 10 GByte per System. On average about 3 Gbytes of data are written to tape and 150 datasets are recalled a day. The oldest datasets are 1290 days old, the average age is 517 days.

Currently KFA is running UNICOS 8.0.4 with the PRIV_SU version of Multilevel Security and DMF 2.2.6. One STK ACS 4400 tape silo is dedicated to the Cray systems. It is connected via a SUN Workstation running ACLS V3.0.

Characteristics of *dmatmsp*

The advanced tape MSP introduced with DMF 2.2 is very valuable for us as it removes some shortcomings of *dmtpmsp*. Its major advantages are the capability of absolute block positioning and end-of-tape processing, so that it can exploit data compression, and that it has better recovery features in case of media problems. This leads to a more efficient usage of the tape media, especially when data compression is used and when there is not enough data to fill up a whole tape at write time.

The improved way of processing allows *dmatmsp* to be more flexible. The data is grouped in a hierarchy of blocks, chunks, and zones. A file fitting on one tape consists of one chunk, if it is spanned across tape boundaries it has multiple chunks. A zone is a logical group of chunks. It contains as many chunks as are written to one tape in one operation. The target zone size specified in the volume database is the amount of data which triggers *dmatmsp* to write data to tape. The zone size is the tape size at maximum. Each block, the logical I/O unit, contains in its header the zone and the chunk number it belongs to, as well as its position within the chunk. The tape catalogue database knows about the location of each file chunk by recording volume, zone and chunk number.

When writing to tape the medium is always positioned to the next available zone using absolute block positioning. A tape is always filled up with data until EOT. Thereby the media are fully utilized. This is in contrast to *dmtpmsp* which only puts the predefined tape capacity of data at maximum to tape in one operation leaving the remainder of the tape empty. A disadvantage of the advanced MSP might be that it needs more tape mounts than the old MSP. This is caused by the smaller target zone size (50 MByte for non-ER90 tapes) in comparison to a whole tape. Therefore *dmatmsp* reduces the number of mounts by leaving tapes mounted on the drive for another minute before dismounting them. This allows for a possible next operation to start without remounting.

When a file is recalled the tape is positioned to the zone containing the file (the tapes are zone addressable). From the beginning of the zone it moves forward to the first chunk of the file. This feature allows fast recalls of datasets even from high capacity tapes. It also enables DMF to recall all data of a damaged tape except the chunk, in worst case the zone, containing bad blocks. Thereby less data will be lost in case of media problem than with tapes managed by *dmtpmsp* .

With *dmatmsp* a set of new commands is introduced. They are mainly to support the advanced MSP and add integrity checking as well as backup tools:

dmvoladm Administrative services for the tape volume database – relates to *dmvdbgen* and *dmtpmerge* of the old MSP
dmcatadm Administrative services for the tape catalog database – relates to *dmdbval*
dmatread Copies files from MSP tape to disk, can be used for both tape MSPs
dmcopy Copies a byte-range of a file from MSP tape to disk without recalling the entire file
dmatsnf Verifies the integrity of (old and new) MSP volumes
dmatvfy Tool to correlate the contents of the *dmatmsp* database with the DMF database
dmsnap Takes snapshots of all data migration databases for backup purposes without interrupting DMF operations
dmastat Reports DMF activity statistics; it needs the LOG_ACTIVITY parameter to be set in the data migration configuration file
dmatcvt Utility to convert the databases of the old MSP to the ones of the advanced MSP

Migrating to the Advanced Tape MSP

In May '95 we evaluated the advanced tape MSP to exploit the data compression capability of the 3490E tape controller. Because of SFN #1155 from April 10, 95, which says that there is a risk of loosing data of multireel files when converting the MSP databases, it was decided to build the advanced MSP from scratch. This decision implied that the data had to be moved from one MSP to the other by hand.

We wanted to test *dmatmsp* with only a few selected userids first and then enlarge the group of users stepwise.

I. First Trial to Configure

First of all the DMF configuration needs to be extended by the new MSP. In addition the *archmed* value in the UDB, which defines the MSP to be used, has to be specified for all users who should use the new MSP. The *archmed* value and the DEFCOP value (giving the number of copies of a dataset residing in DMF) are used to determine the index of the MSP in the *dmf_config* file which manages the data for the particular user. The default value for *archmed* is 0, which selects the first MSP (*cart1* in our case). For *archmed* set to 1 another MSP has to be selected via its index in the DMF configuration file. The DMF Administrator's Guide (SG-2135) gives the default algorithm for the index calculation for the MSPs to be used in form of the following table:

DEFCOP	*archmed*			
	0	1	2	*n*
1	1,0	3,0	5,0	2*n*+1,0
2	1,2	3,4	5,6	2*n*+1,2*n*+2

This algorithm is not suitable for a site using DEFCOP=1 and wanting to use two MSPs configured by

```
MSP_NAMES cart1 cart2.
```

Obviously, the algorithm assumes that there is a MSP defined with index 3 in case a user has the *archmed* value 1 (see the marked field in the figure above). We verified during our tests that the second MSP could never be referenced . The algorithm is designed only for sites using DEFCOP=2. To be able to address *cart2* one solution is to add an additional (dummy) MSP to the DMF configuration:

```
MSP_NAMES cart1 dummy cart2.
```

This definition allows to use both MSPs, *cart1* and *cart2*, but results in error messages at startup saying '*VOL database /usr/dm/msp/dummy/tpvol does not exist*' and, after creating one and adding a tape to it '*Only 1 free migration tapes remain in VOL database ...*'.

II. Second Trial

An alternative solution is documented in the DMF Administrator's Guide: Modify the function *dmmfunc.c* to implement a site specific index algorithm. The *dmmfunc* source is available to all DMF sites. It is used to implement the calculation of the primary MSP index. Unfortunately, the manual does not document the dependencies of the routine, i.e. which parts of DMF are using it. So we modified it by simply setting the primary index to *archmed*+1 in case of DEFCOP=1 leaving the rest unchanged. This worked for all automatic operations but failed in case a user explicitly requests two copies of a file with a *dmput* command not knowing just one is allowed. In this case the user gets the prompt without any error or warning message but his/her request is not performed, there is not even one copy of the file in DMF. Therefore we extended the modification by specifying *archmed*+1 for the primary index in both cases, 1 and 2 copies. From our point of view it is not a good idea that one has to modify a routine being left blind about the consequences, detailed documentation and a user exit would be far better.

The following shows a part of the DMF configuration file we are running with:

```
DEFCOP=1
MSP_NAMES cart1 cart2
cart1 MSP_TYPE tape COMMAND dmtpmsp
cart2 MSP_TYPE tape COMMAND dmatmsp
NTAPE DEVICE_TYPE 3490e \
      MOUNT_OPTIONS -l\ sl\ -i\ on
```

III. Privilege Setup

The configuration above still requires additional prerequisites as we experienced when first wanting to access tapes in the new MSP. With the mount options also used for tapes of *cart1* we are not able to access the standard label tapes in *cart2*: The tape message says
'Flaw Tape Msg: reason=1, msg=process_mount_tape: mount tape error:vsn=...' and
MSP Tape Info: vsn=(NULL), bs=65536, type=NTAPE, label=blp, ...
For some unknown reason *dmatmsp* always wants to mount its tapes with *labelbypass* label. So, *root* needs to have the *labelbypass* permbit to be able to access tapes in *cart2*. In addition *root* also needs the *tape-manage* permbit defined in the UDB, otherwise *dmatmsp* cannot position to an absolute block address on a tape:
'Flaw Tape Msg: reason=6, msg=position_tape_to_zone: Could not position to absolute address'.

After providing all these prerequisites, *root* privileges and index calculation algorithm, we started to use *dmatmsp* for a few selected users. All other users can access their migrated data and new files of the selected users are migrated by the advanced MSP while they still can access their old data managed by *dmtpmsp*.

IV. Moving the Data

With the intention to replace *dmtpmsp* by *dmatmsp* step by step we set *archmed* to 1 for all users in the UDB. The new MSP then manages all the new data on the /arch file system which needs to be migrated. Access to old datasets is still possible via *dmtpmsp*. This is transparent to the users.
dmselect offers the functionality to select datasets from given MSPs according to a minimal or maximal size, age and/or ownership. The chosen datasets can be passed on to the *dmmove* command which then internally recalls the data, migrates it to the specified MSP, changes the DMF database, and removes (hard-deletes) the entry in the "old" MSP tape database. For recalling the data *dmmove* uses space on the file system specified by the MOVE_FS configuration parameter. The working space on the file system can be limited by specifying a maximum of Mbytes with the –s parameter.

For KFA the appropriate selection criterion was the userid. We started with the top users, occupying tenth of GBytes and ended with the numerous having just a few Megabytes. For the top users the dataset age was added as selection criterion, which helped to restrict the amount of data to be moved in one step.
dmmove recalled all the selected datasets in a very time consuming process: The data of a single user was spread over many different tapes in the old MSP which all must be mounted and read. This was really an endurance test for the tape equipment. Moving all the data from *cart1* to *cart2* took us about three month during working hours having four tape drives reserved on each machine.
In our opinion it would be good to have an option to move the data tapewise, this would shorten the move process considerably.

During the process we had the problem that the MOVE_FS filesystem was flooded with recalled datasets. The recall of datasets was aborted when the file system ran out of space. We didn't loose any data because *dmmove* proceeded writing to tape until the working space was cleared. To prevent this situation we used the –s option to restrict the working space in the following time.

As the data was moved on a per user basis we now have the "old" data of a user grouped all together on the same tapes. It is advantageous if a user recalls multiple datasets with a single *dmget* command. Otherwise you see a mounted tape being winded forward and backward over a long period. With the old MSP this burden was normally spread across several tapes.
Having a *dmselect* option to move data on a per tape basis would have shortened the move process considerably. Related to SFN #1155 the database convert is working for all single tape files so that only multi-reel files must be moved using *dmmove*. This would be the best thing to do for most sites.

Early Experience

Beside all the small problems we ran into we are quite satisfied with the new MSP. There are now about 700 GByte managed in *dmatmsp* together on both systems (status as of September 10, 95). These data occupy about 750 tapes. Most of the tapes are filled with about 1 GByte. The compression factor is not as high as expected because the majority of files contain dense binary data.

Like in DMF 2.1 statistic data is gathered in *dmdact.** and *mspact.** day files in the DMF spool directory, if the LOG_ACTIVITY configuration parameter is set. Now the *dmastat* command is offered to format the collected data. This gives a good overview of the MSP activity as shown in the table below. It is an extract of the data gathered on one system over a period of 20 days. The average time to position in case of recalling data varies a lot from just a few seconds up to several minutes, on average it is about a minute.
In the recall statistic given the column containing the 'Average Data Xfer Rate' is omitted due to lack of space. It varies between 0.1 and 3 MB/sec.

Until now we don't experience serious problems with *dmatmsp*. The only problem occurring several times is an unrecovered tape error saying 'Backward at BOT'. This is already documented in SPR 93378 and fixed in DMF 2.3.

```
                         Migration Statistics
                         ____________________

                               MSP cart2
                               _________

          Number                          Data
            Of                          Xfer Rate
Volume   Requests   Amount Migr.        (MB/sec)   Mounts
______   ________   ____________       _________   ______

118885         41    1915.805 MB         2.122          2
118886        112    1667.807 MB         2.121          5
118887        258    1671.187 MB         2.006          3
118888        296    1231.000 MB         1.813          2
118889         23    1266.255 MB         2.425          3
118890          4     998.051 MB         1.574          2
118891          3     997.342 MB         2.528          1
  .
  .
           ________  ____________

Total          5817  39494.015 MB
```

```
                            Recall Statistics
                            _________________

                               MSP cart2
                               _________

                                  Average
        Num.                      Time to   Mount
        Of       Total           Position   Time
Volume  Req.  Amount Rec.          (Secs)  (Mins)  Mnts
______  ____  ___________   ...__________  ______  ____

118834     2  131.132 MB         29.820   0.543      2
118878     1  147.386 MB         37.717   0.542      1
118849    11  513.159 MB         68.222   0.633      9
118705   169    0.828 MB        114.832   0.955      3
118715     5    5.697 MB         19.259   0.784      2
118714    11   21.476 MB         80.155   0.768      4
118778     9   67.169 MB        102.742   0.631      6
118804     5  291.081 MB         52.843   0.680      5
  .
  .
        ____  ____________

Total   1248  17794.740 MB
```

Conclusion

dmatmsp fulfils our expectations: It has a far better tape usage than the old tape MSP and supports backup and verification tools. It ran reliably up to now. The statistics formatted with *dmastat* give a good overview of the MSP activity.

Looking at the configuration there are some disadvantages. Better documentation of the prerequisites for using the MSP is needed: The *labelbypass* and *tape-manage* permbits for the userid *root* are required. An explanation for what reason these privileges are needed should be given. Furthermore the default index calculation algorithm is only suitable for sites using two dataset copies residing in DMF. Other sites have to modify the routine *dmmfunc.c* and they are left blind about the consequences. A far better approach would be to implement a well documented user exit.

Offering the convert utility *dmatcvt* is very useful as the process of starting with *dmatmsp* from scratch and moving all the data "by hand" is very time consuming. It is only acceptable for small DMF sites.