

Systems Managed Storage - Getting there is half the fun

Henry Steinhauer, Hewitt Associates

INTRODUCTION

It is not the scope of this paper to give a detail view of Systems Managed Storage (SMS)*. There are other papers that address themselves to this topic very well. We do not need another one. What is not plentiful, at this point in time, are papers dealing with the success of using the SMS concept with more than just the IBM family of products.

The goal of this paper is to present actual user experience of implementation from a user who still has to work at the shop where the installation was done. I could not afford the consultants role of recommendation and then leave. I had to stay.

Part one - Why we have the mixture we have.

When the view of SMS was introduced by IBM it was stated that any vendors products could be used for the boxes that IBM was using for given functions. These boxes are defined as: Active Data Management (the functions in DFP/MVS*); Data Movement and Conversion (DFDSS*); Inactive Data Management (DFHSM*); Resource Protection (RACF*); Data Sorting (DFSORT*). Of prime importance to us was Data Movement and Conversion function and Inactive Data

Management or the use of HSM* or FDR*. Next in order of importance was Active Data Management and the continued elimination of X37 abends. This had been accomplished for us by a product called STOPX*. We had problems with jobs not requesting enough space and then abending with the common X37 abends. The fact that the users were asking for large amounts of data one time and then very little the next did not remove the impression in their mind that we could not manage our DASD pools to prevent them from having Space Abends. It did not help that the primary allocation could be obtained in as many as 5 extents and thus leave them with only 11 more extents before a dreaded abend. We solved this perception problem with StopX* and did not want to leave this feature behind. The success of doing this was presented in a prior CMG paper. (CMG89)

We created a report during the period of watching free space and working to prevent space abends that is created 3 times a day for management reading. It may seem strange for a detail operations report that is created at 8am, 1pm and 4pm showing free space, Job Class Queue depths and printer lines waiting for print to be sent through PROFS* to upper management but it is true. The nature of our Data Processing center has users communicating with our DP executive about these topics.

He has found it prudent to be kept informed.

The tri-daily reports required learning more about the DASD allocation that happen in our installation. We have now established triggers for each of these reports for the DASD space numbers. When the DASD values fall below the triggers then action is required. Not only is global free space or percent free used but also the number of datasets of a certain size that could be created. This helps us to gauge the speed at which datasets are created during the day.

We do our DASD charge processing by reading the VTOCs on a regular schedule. Since we already have the information further processing of the VTOC information is cheap. From this we were able to use UNIVARIATE descriptive statistics to build a model of our DASD farm. With 90,000+ datasets it is important to look for a big picture and not get tied up with an application by application view. 80% of our datasets are less than 1 cylinder. This allowed splitting our pools into large and small divisions. The need for large amounts of free space is only needed for the large pool. The small pool can work just fine with 1 and 2 cylinder holes everywhere. This fragmentation would be death for the large pool though.

The large pool has a 90% size of 40 cylinders. With 3,000 datasets in the large pool that means 300 datasets have a size larger than 40 cylinders.

Combined with the rules for Inactive Data Management where we let these datasets stay for 5 days from last reference this implies that we need to have room for 60 datasets to be created in any one day. Since an allocation can be split into 5 extents to meet the demands for a primary space request and given the fact that the 99% size is 122 cylinders then the 40 cylinder size for our Rule Of Thumb value is valid. We have taken the rules one step further by requiring twice the number of datasets at 8 AM, going to 60 dataset areas by 4 PM. If the limits fall below those targets we have to take action. This action has been either to add volumes to the pool or to have a search and destroy mission for really large datasets that 99% of the time have been in error anyway. These targets give us an early warning on DASD problems before our users begin to report problems with their jobs.

The way that we look at the actual free space on volumes allows us to look at each free space extent. We use a function of FDR* to give a free space map. This allows us to review all the storage pools in less than 10 minutes. For each free space extent we find the number of 90% cylinder areas that would fit. Adding up for a storage pool gives us the actual value to compare the targets. Also adding up the whole cylinder extents gives us an indicator of total space available in the storage pool. That also is used as a trigger for action.

Up to this point our DASD

pooling was just a General Storage split into Large and Small pools. It is common for the Small pool to have 3,000 - 4,000 datasets per volume. When the 95% dataset is only 1 track in size then a lot of datasets can be held on one volume. The Large pool has the normal amount of datasets in the 100-200 range with a 95% size being at 40 cylinders. Other common Pools are: Public, CICS* Test, CICS* Prod Datacom*, CICS Prod Non-Datacom.

Our next step was to start to split up the General Storage pool. This pool had grown to just more than 70 3380K volumes. Daily DASD maintenance jobs were consuming more of the critical overnight window. A split was needed.

Part Two - Splitting of the pools

We began to split out a pool that we called Temporary. This pool has datasets that last 3-5 days from date of last reference. Thus datasets can be used in one job and then used in another job without having to be concerned with keeping that dataset forever. There is no need for archive in this pool. When their time is up then they are scratched. Our naming convention is to use a second node of T0 or P0 for 3 days and T1 or P1 for 5 days. The T and P means Test or Prod. Our original thought was to divide these into their own pools. The problem with the division was explored in the original split between small

and large. If the pool is too small then the normal swing in allocations can cause many problems. By combining Test and Prod temporary datasets, the pool was large enough to absorb the swing.

As the datasets left the general storage pool we would remove volumes from the general storage pool and add them to the temporary pool. We started with 3 3380K and now have 8 volumes. The rate of change has stabilized and that pool is now fixed. We do not expect changes in this pool. Our General Storage pool was able to be reduced by more than 8 volumes. This is explained by the more stable population of keeping the temporary datasets in their own pool. The percent used goes between 35% to 80% depending on where we are in the processing cycle.

Next created a TSO pool. Started with 2 volumes. This pool was all the datasets that started with TSO IDs. This pool is now stable with 3 3380 triple(K). The percent used rides between 65-75%. Again this was able to be taken out of the General Storage pool.

Next to go was a Test pool. These are identified to us as a second node of T and not a temporary dataset. This was divided into large and small pools. We had learned that we could not combine the 1 track allocations with the 40 cylinder allocations. The amount of free space on a volume is much higher for those volumes that are in the large pool than those that are

in the small pool. The Large pool swings from 70-80 percent used. The small pool swings from 75-85 percent used.

Part Three - Problems

It was at this point that a few problems started hitting hard. The crux of the problem was the difference between the way DFSMS handled dataset allocation and non-DFSMS allocation.

The first problem was size requested. The pool can appear to have lots of space and yet users can begin to get JCL errors saying there is no room on any of the available volumes in the defined pool. This problem is hard to see. There is no message written to SYSLOG or recorded anywhere external to the job that has the problem. The message is only displayed in the messages for the job where the allocation messages are displayed.

When we first began to have the problem we thought it was just a typical user problem. When we looked deeper we saw that it was a problem with the number of free extents and the size of the largest free extent. When a pool does not have enough large extents there is a good chance that you will experience this type of problem.

One of the first ways we used to work around the problem was to change the split allowing datasets into the SMS managed pool and keeping them out. This was done by primary space

allocation size. We did not like using this rule for separation but it did reduce the JCL problems. It also gave us some working time to look for other solutions.

The purpose of highlighting it in this paper is to alert you to possible hidden problems. The message number is in the IGD172 family. Be on the look out for the problem. Keep in touch with your users and check for unusual reports of JCL errors.

Another problem we encountered was with GDGs. When SMS was first introduced it was clear that GDG processing was not going to remain the same. Our problem is in the restore process. By default, when a GDG is restored it is left in a status state called 'deferred'. This means that if a job references a GDG as +0 that it does not reference this GDG level even if it is the current level. Also that if a range of relative numbers is referenced that this certain GDG number will not be allowed to be in that relative range. If the GDGs are numbered 191, 192, 193, 194 and number 193 is in deferred status because of a restore then a reference to -1 would reference 192 and not 193 as expected. When the GDG was archived it was in 'active' status. When it was restored it was made into 'deferred' status. The only way it can change from 'deferred' to 'active' is to issue an IDCAMS command to ALTER the status with the real number (193 in our example).

It is possible to flag that when a GDG is restored that it should be restored as 'active'

but that is not the default restore status. We were alerted to this problem quickly by the users. When we talked with out vendor we found out they had just that week heard of that problem and had a fix for us. We also were able to scan the VTOCs looking for all GDGs and altering them to be in 'active' status.

Part Four - Recommendations

A good working relationship with your primary users is critical to the success of bringing control to the DASD farm. We have done this by having weekly status meetings with the primary support group of the primary user. This has kept open lines of communication and helped to reinforce that we are not holding back on them or trying to make their processing harder. There have been times where we have implemented a new rule for datasets and then found out that just because it made good logic to us it caused them problems that no one expected. An example is not to allow renaming datasets from TSO ids to production ids that we found the primary user had a part of their monthly processing cycle renaming files as part of their own application recovery methods. This caused them to create a number of changes to their production processing very quickly. They understood and agreed to the concept once it had been explained from our view point. They had just never looked at it with that view point.

The good working relationships

also allowed us to pin point the problems with the GDG processing. We were able to respond to the problem call and find a fix without having to move backwards in the plan.

We are still working on finishing the conversion to SMS. We have been able to divide the DASD daily jobs into job sets with one set per pool. This has allowed more parallel processing to occur in the DASD area. Before we had one set for the large pool and another set for the small pool. Now we are able to have a separate set of jobs for each pool. This allows us to have more parallel processing during the critical overnight window.

Knowing the profile of your DASD Farm is very critical. This knowledge can be gained by knowing the archive rules for your datasets, knowing the descriptive statistics for the datasets in each pool and by keeping a close eye on the behavior of your DASD Farm.

Trademark Notice

*SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA

*SMS, DFP/MVS, DFPSORT, DFHSM, DFDSS, RACF, PROFS are a registered trademark of IBM.

*STOPX is registered trademark of Empact Software.

Author

Henry Steinhauer
Hewitt Associates
100 Half Day Road
Lincolnshire, Il 60069
(708) 295-5000