SLAC – PUB – 3869
January 16, 1986
N

Managing a Large VM Directory

Installation Code: SLA

VM Systems Management Project 0608

T.Y. JOHNSTON

Stanford Linear Accelerator Center

Stanford University, Stanford, California, 94305

ABSTRACT

A discussion of some changes to improve CP directory performance and of some tools for managing the CP directory.

Presented at SHARE 65, New Orleans, La., August 11-16, 1985

⋆ Work supported by the Department of Energy, contract DE – AC03 – 76SF00515.

1. INTRODUCTION

When the CP directory was designed in the late '60s directories of several thousand entries were not envisioned. In addition IBM has not provided tools other than DIRMAINT for managing the directory. Although, we have found DIRMAINT a necessary facility we wanted our users to manage their own minidisks.

SLAC does not have a massive CP directory. We have under 2,500 accounts and under 4,000 minidisks. However, we have a fairly high level of directory change. I will discuss what we have done to both improve performance and better manage the directory.

1. Performance Improvements
2. The performance changes group into 4 general areas:
3. Changes in the directory structure and access.
4. Changes in the functional interface to the directory.
5. Reduction in the size of the directory.
6. Miscellaneous performance enhancements.

1.1 Changes in the Directory Structure and Access

The CP directory as distributed has no structure or understanding of any structure. Consequently, to logon an account, spool a device to a userid, etc. it is necessary to search the entire first level of the directory to find the account (or to find out that it does not exist). This is a linear search. There are 170 directory entries per page so to find out that an account does not exist for a 2,500 entry directory takes 15 page reads. In addition there were insufficient consistency checks. CP's pointer to the active directory may not match the allocation map on the volume. Because of this the active directory can be overwritten and there is insufficient consistency checking to detect and gracefully recover.

SLAC runs DIRMAINT and consequently runs with a alphabetically sorted directory. With this in mind the following extensions were made:

1. An index of the lowest userid in each page was built in memory during initialization and at directory switch. This index was accessed to determine the page to read to find any account. This changed the number of page reads to find an account from a linear function to being a constant of one.
2. All pages were time-stamped so as to correlate that a consistent copy of the directory is processed. A mismatch forces switch processing.

The effect of this change was to lower the directory I/O on our system by a factor of 4.

1.2 Changes in the Functional Interface to the Directory

To add or delete a minidisk on the standard system one must re-create the entire object directory from the source using the DIRECT command. We felt that this was overkill and caused a significant amount of unnecessary I/O. Consequently, we modified the directory so that the UDEVBLOK pages were not full and extended UPDATE-IN-PLACE to support the addition and deletion of minidisks. We went from averaging many DIRECT commands per hour during prime shift to a few per day. Since the directory consists of tens of thousands of card images this both reduced DASD contention and gave the requestor much better response.

1.3 Reduction in the Size of the Directory

The directory contains huge amounts of redundant information. Everybody has an 00C, 00D, 00E, 190, and 19E that are almost the same. By the use of an IPLER these five cards can be removed from the directory in 99+they define their virtual printer different from our standard configuration. This change (besides its other benefits) enabled us to reduce the size of the source directory from over 30,000 cards down to about 20,000. There are other features of an IPLER that make it advantageous in addition to reducing the directory size.

1.4 Miscellaneous:

DIRMAINT does internal sorting for such functions as creating a new minidisk. We replaced the use of CMS SORT with XEDIT SORT.

2. DIRECTORY MANAGEMENT

When we were planning our conversion to VM a few years ago one of our goals was to allow the users to manage their own disk space rather than centralizing the function. This had been the practice under OS and worked quite well.

The users at the laboratory are broken up into groups with a group czar for each one. The groups are not arbitrary, but reflect the organization of the laboratory. Thus the personnel department and accounting departments are groups as are the various experimental groups. The group is given a space allocation by the computer center and manages that space. The computer center is only involved in changing the groups allocation when a group needs more space (there are almost never any cases when a group says it needs less space). The following functions are available:

1. Release a permanent disk.
2. Transfer a disk to another user.
3. Change a disk address.
4. Change size of a minidisk.
5. Add a permanent minidisk.
6. Obtain a temporary disk.
7. List status of temporary disks.
8. List group disk czars.
9. Show group disk space.
10. Set a user's password for him.

In the discussion of the various commands I will indicate who has the authority to issue the various commands.

There are a set of commands to manipulate permanent minidisks. These include the ability to obtain, release, transfer, change the address of, or change the size of a minidisk. The authorities are what you would expect. Only czars can give out space, but anyone can get rid of his own, or change its address, or give it to somebody else.

The temporary disks are pre-defined minidisks which are transferred to any user on request. They have a nominal life of 24 hours (they are removed from the users directory after 24 hours), but are not actually freed until the user detaches the disk (this can be caused by logoff or system restart). There are two major advantages to these disks over regular T disk:

1. The disks can be shared since they are in the user's directory. This allows one to put data on the disk and share it with either other users or one's own batch work.
2. The disks survive across system crashes. There is the ability to 'renew' a disk after the 24 hours is up so that it will not be lost.

In addition to the disk functions there are a few query commands and a tool to allow the czar to set a user's password for him. This allows a czar to set a password for a user that has forgotten his password. Why would one want this type of function? We have many visiting physicists who are not very familiar with VM. The physicists also work strange and long hours. Our service desk is a 9AM to 5PM weekday function. Most users know how to get hold of their czar day or night and thus can get 24 hour service. This is not a service that they are apt to abuse, since the czar controls the keys to their computing kingdom.

3. SUMMARY

I have described two different parts of managing a large directory: making it more responsive to user requests and decentralizing the routine functions. The minidisk management functions are part of our SPACE machine which is available on the Waterloo Mods Tape. The indexed directory is also available on the Waterloo Tape.