SESSION REPORT

61	B591	Multiple VM Systems Sharing DASD	157
SHARE NO.	SESSION NO.	SESSION TITLE	ATTENDANCE

VM System Management Project	John Bevis	UF
PROJECT	SESSION CHAIRMAN	INST. CODE

NERDC, Room 107 SSRB, University of Florida, Gainesville FL 32611
SESSION CHAIRMAN'S COMPANY, ADDRESS, and PHONE NUMBER

Richard Alexander of Cornell University (CUN) and Donna Walker of the Central Intelligence Agency (CAD) presented two views of the problems and pitfalls of sharing DASD among multiple VM systems. Richard's presentation follows and Donna's foils may be found in Volume 2 of the proceedings

VM/370 AND DASD PLANNING

RICHARD ALEXANDER

CORNELL UNIVERSITY - CUN

SESSION B591/B662

SHARE 61

NEW YORK, NEW YORK

AUGUST 1983

Introduction

Cornell University is a long time user of VM/370. Since 1980 we have operated a multi-CPU configuration. Until recently we had three 4341s and one 370/168. Currently we have one 4341 and one 3081 Model D.

Our approach to backup paths is that we provide one secondary path to every device through a second control unit normally on a second channel. On the 3081 the second channel is on a different data server element (DSE) and a different channel set.

We use an Intel 3805 solid state drum with two control units and 36 mb of memory running in native mode (FB-4096). Both of these paging control units have two channel switches.

All of our disk control units have four channel switches. We have five IBM 3880s (10 control units or storage directors), two IBM 3830-2s, and two Memorex 3674s (equivalent to IBM 3830s). We have IBM 3380s, 3370s, 3375s, 3350s and Memorex 3650s (3350 equivalents) and 3675s (3330-11 equivalents). Some of the IBM 3350s and Memorex 3650s are running in 3330-11 compatibility mode.

Bibliography

There are three manuals I have found useful with descriptions about the ramifications of I/O and VM/370. The first is the "VM/370 Planning and System Generation Guide", SC19-6201-2. This manual is explicit about details of the entries in DMKRIO. A second manual, "VM/370 Operating Systems in a Virtual Machine", GC19-6212-1, contains an excellent discussion of virtual Reserve/Release support in CP. Finally, the most impressive hardware book I have found is one of the new IBM "Cross-System" Series called IBM 3880 Storage Control Models 1, 2, 3, and 4 Reference Manual", GA26-1661-6. This new series of IBM manuals has good depth and clarity.

Two areas poorly documented are 1) the IOCP restrictions about alternate paths for 308x processors and 2) the CP alternate path support which should be in the System Generation Guide mentioned above.

I/O Overview

I have limited the discussion of I/O to DASD devices. Other device addresses have a similar structure, but may vary in detail. The 370 I/O addressing scheme is reasonably straight-forward. Every device is connected to a control unit and every control unit is connected to a channel. All devices must have unique addresses to the system. Three nibbles (4 bits each) are used to address a device. Typically, the address 12B refers to a device on channel 1, control unit 20, and device B (base 16). Figure 1 depicts the address structure.

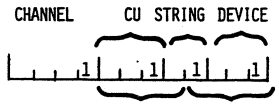
Unfortunately, there is additional complexity introduced by the head of string or "controller" function. Each control unit can be

connected to multiple heads of string which must be uniquely identified in the I/O address. Each head of string can be attached to two control units with what is called the string switch feature on normal DASD. The "string switch" feature per se is not available on the 3380s. However, if you invest in the dual head of string unit (3380 AA4), then the function of the string switch is included.

For the IBM 3380, each control unit can address two physical strings of devices. Therefore, there is one bit set aside for the head of string address. A physical string of 3380s has 16 logical addresses. Four bits are needed to address 16 devices. For other IBM DASD (3370, 3375, 3350, 3330), there can be four heads of string attached to a control unit. Two bits in the I/O address are needed to identify the head of string. Three bits are needed to address the 8 devices on the head of string.

Figure 1

I/O ADDRESS ANALYSIS
3370/3375/3330/3350



#CU	#STR	#DEVICE
8	4	8
00-F0	00	0-7
	01	8-F
	10	
	11	

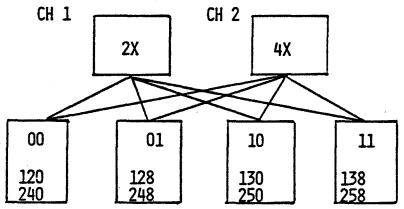
Typically a DASD control unit (a storage director in a 3880) will address 32 devices, although a control unit could address 8, 16, 24, or 32. A mistake that many system programmers make is to forget to code FEATURE=32-DEVICE on the RCTLUNIT macro in DMKRIO when the control unit has a 32 device capture range. CP's default for all control units is a capture range of 8. This mistake manifests itself most frequently in CP throwing away interrupts from devices. This is the most common reason for the "missing interrupt" problem.

A control unit which captures a range of 32 addresses will support 4 strings of 3370s, 3375s, 3350s, or 3330s. These disks come in strings of 8 devices. A physical string of 3380s represents 16 devices so a control unit will only support two 3380 strings. When a control unit has a 32 device capture range, its base address must be an even number, for example 20. If 20 is the base address, then the capture range is x20-x27, x28-x2F, x30-x37, and x38-x3F.

It has been my experience that the major confusion about I/O addresses is to remember that the capture range encompasses both x2x and x3x.

Figure 2

I/O ADDRESSING



VM and Alternate Paths

Basically there are two reasons to need more than one path to a device. These reasons are performance and redundancy.

To satisfy the redundancy requirement one can configure two physically separate paths to a device. These two paths can simply be generated in DMKRIO as two separate devices. If both paths are online when CP initializes, the lower numbered path will be used and the higher numbered path will be varyed offline. If the channel, control unit, or head of string fails, then the second path can be varied online and used. Of course, if a path to CP-owned volumes fails, an IPL will be necessary. This arrangement is static. Only one path will be used for a given IPL.

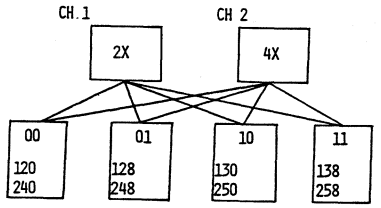
By using the ALTCU and ALTCH options on the RDEVICE and RCTLUNIT macros, you can tell CP that there is more than one path to a device. If this alternate path is declared, then CP will automatically use the second path if the first fails. No IPL is needed. More importantly, CP will use the second path whenever the first path is busy. The use of other paths in a dynamic fashion is what I mean by alternate path

support. Since Cornell does not use the ALTCH option, my discussion will center on the ALTCU option. The ramifications of both are identical.

In some operating systems, one may choose the algorithm to be used in dynamic path selection. In CP, there is only one supported by IBM. It simply uses the second path whenever the first is busy. You can take advantage of this algorithm by defining the primary path on one CPU to be the secondary path on the second CPU. This trick gives us control unit separation across processors and yet still allows use of the secondary path during busy periods.

Figure 3

I/O ADDRESSING




```
RDEVICE ADDRESS=(120,8)   ...ALTCU=240
RDEVICE ADDRESS=(128,8)   ...ALTCU=248
RDEVICE ADDRESS=(250,8)   ...ALTCU=130
RDEVICE ADDRESS=(258,8)   ...ALTCU=138
RCTLUNIT ADDRESS=120      ...FEATURE=32-DEVICE
RCTLUNIT ADDRESS=240      ...FEATURE=32-DEVICE
```

If the algorithm CP uses for alternate path selection does not fit your needs, there is a second one given in file MEMO ROT.CHAN on VMSHARE. The update to DMKIOS is straight-forward. The algorithm switches the primary and alternate control units of a device whenever DMKIOS receives a busy condition from an attempt at I/O initialization. Cornell has had this modification installed and noticed no particular performance improvement. We may reinstall the modification later if it seems that the DASD load characteristics at Cornell have changed.

A major weakness of the alternate path support is that there is only one activity counter in an RDEVBLOK. SMART and VMAP can only report activity by primary path address. A minimally acceptable solution is to have a counter for the primary path and another counter for all other path accesses. A much superior solution is to have an activity

counter for each path generated. Without activity by path it is very difficult to evaluate how alternate pathing effects performance.

Sharing Data on DASD

There are two ways to share data in read/write mode. The first is to share full volumes of DASD. The second allows sharing of minidisks on the same real system.

The 370 I/O architecture has two channel command words defined which allow two real systems to share a DASD volume. After a Reserve operation code (x'B4') is issued, a volume will present a busy condition to any other path attempting to access that volume. Data on the DASD volume is safe to update from the reserved path until a Release operation code (x'94') is issued on the reserved path. This is an extremely primitive sharing mechanism and has several drawbacks. The first problem is how to clear the reserved condition if the issuing system crashes and is not immediately restarted. The second drawback is that locking the entire volume is too coarse a mechanism to maintain satisfactory performance.

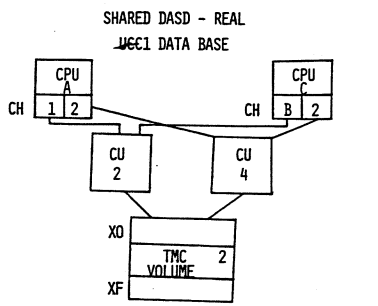
CP will allow any virtual machine to embed Reserve and Release operation codes in channel programs. The major restriction is that if an alternate path is defined to the volume, CP will quitely change all Reserve and Releases to Sense operation codes and never even wimper! Although this restriction is documented, there are more than a few people who have been caught short by it. The implementers of alternate path support in CP took the easy way out! By outlawing Reserve/Release they did not have to remember which path had a Reserve outstanding and force all I/O down that path. The mechanism is present in the I/O block to remember which real path was used, but this information is used only by error recovery.

The second way to share is to share minidisks. If the two (or more) users are on the same real system, a minidisk can be shared among them. By coding V on the MDISK directory statement, you will cause CP to use the so-called virtual Reserve/Release support. Figure 1 in the manual GC19-6212-1, Operating Systems in a Virtual Machine", details CP's actions in every possible software and hardware combination. The application programs are responsible for building Reserve and Release channel command words and imbedding them in the channel programs. CP will honor the Reserve and Release for other requests to access the minidisk on the same CPU. It will allow the Reserve and Release operation codes to execute on the real hardware. Other real systems with the same minidisk defined on the same volume will be synchronized for access as one would expect.

Cornell uses the tape management system marketed under the name of UCC1. We also use the VMSOFTWARE product, VMTAPE, which accesses the UCC1 database. We run VMTAPE on both real processors and MVS with UCC1 on one of the processors. Thus we have three virtual machines on two real systems accessing a common OS volume. We allow common access by attaching the shared volume to each virtual machine through a

different hardware path. UCC1 and VMTAPE use real RESERVE/RELEASE operation codes so we cannot use CP's alternate path support for this volume. A weakness of this approach is that, if we lose one of the paths to the system running UCC1 under MVS and VMTAPE, we can not operate the VMTAPE virtual machine until the path is repaired.

Figure 4



IOCP Restrictions

Before our 3081 arrived, we drew up an address configuration which continued Cornell's tradition of having control unit pairs backing up each other. Normally the control unit with address x0x backed up the control unit with address x2x while control units numbered xCx and xEx backed up each other. As I attempted to build an IOCDS with this configuration, I was told by IOCP that the second address had no control unit defined. Symptomatic of the general lack of consideration that VM/370 seems to have been given by the 308x developers, the IOCP generation will not allow alternate paths to devices through control units with the same number. This is a restriction from the MVS world, and as far as I can understand, has no basis in the hardware reality. The restriction takes away a feature that allows CP to have more flexiblity than MVS in the I/O addressing area. One day before the installation was to begin, our IBM field engineers readdressed all of our pairs of control units so that each pair has the same number.

An alternative which has been suggested to me through VMSHARE was to not tell IOCP that the volume addresses were alternates. It may be

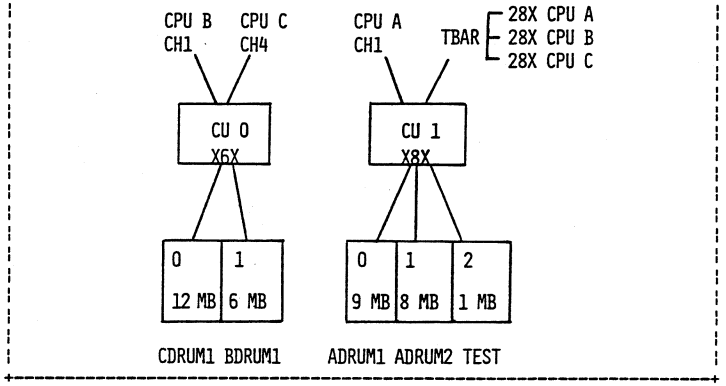
the case that the alternate path information is used only for dynamic path selection in XA mode. However, if you ever want IBM to look at an I/O problem you should do the IOCP generation by the book. We have subsequently had interface control checks on our 3375s connected to the 3081 requiring direct support of IBM San Jose engineering. We are very happy that we reconfigured our addresses to conform to the IOCP restrictions, senseless though they are.

Sharing Paging Devices

We use an Intel 3805 with two control units and 36 mb of memory running in native mode (FB-4096). Both of our paging control units have two channel switches. We have shared a paging control unit between two CPUs. The reason that this arrangement was acceptable is that one CPU had a low paging rate and the other a high one. The solid state drum was still a better paging device, even in this shared environment, than our best disk devices.

Each of our systems has unique CP-owned volume names. For example, ADRUM1 and ADRUM2 are declared in DMKSYS for the A4341 while the 3081's DMKSYS has the names CDRUM1 and CDRUM2. We need only to relabel a drum volume to change the system to which it will be connected after an IPL. This arrangement allows complete flexibility of where to use the drums as load shifts. One must not forget to do an IPL on both of the affected systems, i.e. the one gaining the drum and the one losing the drum, at the same time. If you forget to schedule a synchronized IPL, you may have an unscheduled IPL on either or both of the systems!

Figure 5



3375 and 3380 Device Construction

IBM 3375s and 3380s have multiple logical devices corresponding to unique device addresses on each physical unit. Both have two logical volumes per physical spindle. The 3375s can be given an second controller by acquiring a model D1. The 3380s can also have a second controller by acquiring a model AA4. With these second controllers, two data transfers on the string can occur simultaneously. However, if one logical volume on a physical spindle is busy, any attempt to access the other volume on the same spindle will also report back a busy. This is a subtle design feature (deficiency) with the 3375s. Although I would be happy for someone to correct me, I believe that the same feature is present with the 3380s. Cornell has addressed the problem , excuse the pun, by putting high-use data (like the CP nucleus, spool, and paging spaces,) on even numbered addresses. We reserve the odd numbered addresses for relatively low-use user data. After we gain some experience with the approach, I hope we will be able to report how well it performs.

Summary

The following chart trys to encapsulate the major concerns about an I/O configuration when you are using VM/370 in a complex environment.

Things to Remember

1. RCTLUNIT - Feature= 32-Device
2. RDEVBLOK Counters - 1 per real address.
   Real I/O path usage impossible to determine.
3. Sharing -- Reserve/Release -- Real & Virtual.
4. 3375s & 3380s with two addresses per spindle
   and two heads of string
5. In multi-CPU environments, watch SMART's
   CU/Device Busy statistics for contention
   between real systems.
6. IOCP Restrictions with ALTCU.
7. Channel Rotate Modification from VMSHARE.

SHARE

SHARE SESSION REPORT

61	B632	CMS IUCV		260
SHARE NO.	SESSION NO.	SESSION TITLE		ATTENDANCE
B/Systems CP Management			Stuart Bell	MTW
PROJECT			SESSION CHAIRMAN	INST. CODE

1820 Dolley Maddison Blvd., McLean, VA 22102 (703) 827-6366

SESSION CHAIRMAN'S COMPANY, ADDRESS, AND PHONE NUMBER

Using IUCV in CMS
SHARE 61

August 22, 1983

Samuel A. Thompson

IBM Corporation
PO Box 6
Endicott, NY 13760

ABSTRACT

The Inter-User Communication Vehicle (IUCV) was introduced in VM/SP Release 1 and has been enhanced in Releases 2 and 3 of VM/SP. IUCV is a general use communications vehicle which allows two programs executing in two different virtual machines to communicate with one another. The paper will give a general overview of IUCV and then concentrate on the changes which were made to IUCV in both the CP and CMS components of VM/SP Release 3.

1/E/LEJ/1