I O F S U M M A R Y

* Function
  - Easily review results of batch jobs
  - Control disposition of sysout data
  - Control attributes of sysout data
  - Manage all aspects of user's batch jobs

* Performance
  - Excellent response due to design
  - No system degredation
  - Minimum I/O

* Security
  - No unauthorized access to spool or checkpoint

* Environments
  - All versions of JES2
  - All TSO terminal types
  - All versions of SPF

Slide 28

IOF provides the function that TSO users need to completely manage all aspects of their batch jobs. Users can easily review the results of their jobs and then control the attributes and disposition of all sysout data.

The excellent response time provided by IOF is a direct result of its design. And of course good response time is only achieved by minimizing the hardware resources required to perform each function. IOF reads data only when it is required, and not because it might be required in the future. This means that IOF will actually reduce the total hardware resources consumed by the average TSO user.

IOF does not compromise the security of spool data since it requires no unauthorized access to the spool or checkpoint data sets.

IOF supports all versions of JES2, TSO terminal types, and all versions of SPF.

SHARE

SHARE SESSION REPORT

61	B401	MVS Under VM THrough Conversion & Beyond	200
SHARE NO.	SESSION NO.	SESSION TITLE	ATTENDANCE

MVS New Users	Tom Arnold	UAD
PROJECT	SESSION CHAIRMAN	INST. CODE

United Airlines, 5350 S. Valentia Way Englewood, Co 80111 (303) 779-2157

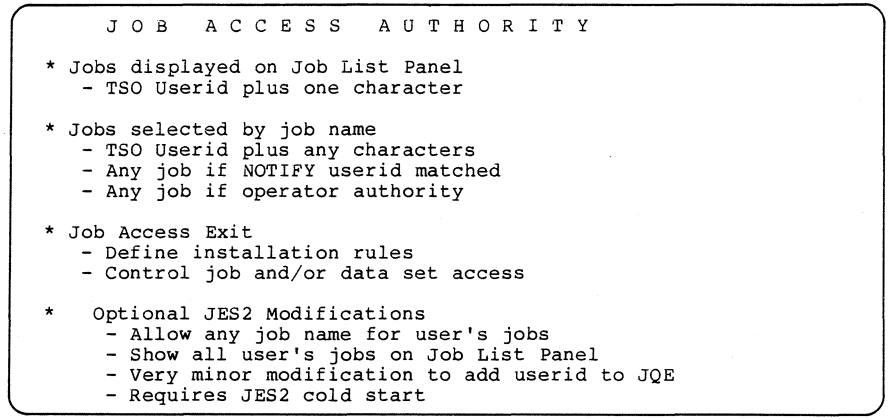
SESSION CHAIRMAN'S COMPANY, ADDRESS, AND PHONE NUMBER

MVS New Users Project

MVS and VM Usage at United Airlines - Denver

Tom Arnold (UAD)
United Airlines - DENKR
5350 S. Valentia Way
Englewood, Colorado 80111

SHARE 61 - Session B401
MVS Under VM Through Conversion and Beyond
Wednesday, August 24, 1983



Slide 26

IOF allows the installation a great deal of flexibility in defining the access rules for reviewing output jobs. First we will discuss the default rules supplied with the IOF distribution.

If the user enters "I" on the SPF Primary Option Menu or "IOF" at the TSO READY level the Job Selection Menu will display all of the jobs in the system whose names consist of the user's TSO userid plus exactly one character.

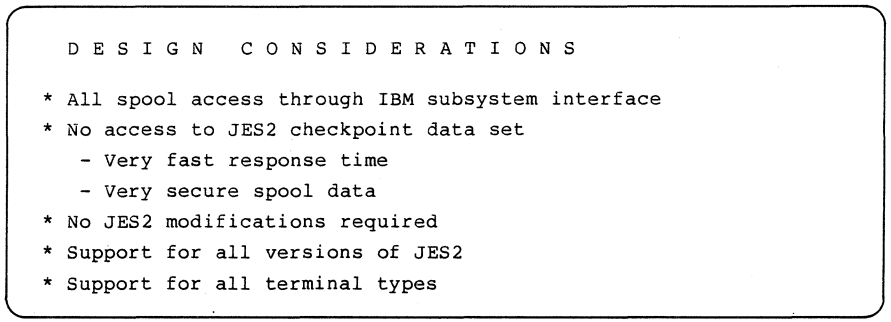
If the user enters "I.jobname" on the SPF Primary Option Menu or "IOF jobname" at the TSO READY level IOF will invoke the Job Summary Display for that specific job if either:

- the job name begins with the user's TSO userid, or
- the NOTIFY parameter on the job's JOB statement specifies the user's userid, or
- the user has been granted TSO OPERATOR authority.

All of the above rules are actually implemented in the IOF Job Access Exit and can be easily modified or extended by the installation. This exit is designed to interface easily to the installations security system.

IOF provides an optional JES2 modification which allows users complete freedom in naming their batch jobs. Users can select meaningful names for their jobs and yet all of the jobs will appear on the IOF Job Selection Menu.

The modification simply adds the USERID to the JES2 Job Queue Element (JQE). By means of the Job Access Exit, the installation can select jobs for review based on this added field. This is a very minor JES2 modification but does require a JES2 cold start since it modifies the JES2 JQE control block. This modification is so simple and useful that it is eventually installed by many IOF installations. It is not required however in order to run IOF.



Slide 27

The primary design objective for IOF was very simple: use as many of the standard interfaces provided by IBM as possible. All spool access is through the IBM subsystem interface.

At no time does IOF allocate or open either the JES2 spool or checkpoint data sets. This is the major reason why IOF is a highly efficient processor that provides excellent response time. This also means that you can protect your spool and checkpoint data sets without affecting IOF operation.

An absolute design objective was that there were to be no JES2 modifications required to install IOF. The optional modifications mentioned earlier can be installed or not at the installation's discretion.

Meeting the design objectives above resulted in reduced JES2 control block dependencies and made it quite easy to support all versions of JES2.

Another major design objective was to provide all of the functions of IOF to non-3270 terminal users. This was a substantial commitment that required extensive effort at the basic design level and at each level of implementation and checkout.

1. Conversion to MVS with VM (Foil 1)

Conversion to MVS began in 1980 and the 3.8D IPO (Non-SE) was ordered from IBM. At that time VM/370 was already installed and up in production on a 4Meg 370/168. The VM system was used mainly for running multiple ACP (Airline Control Program, also called PARS) test systems. There was limited CMS usage (systems) as all applications and systems programmers were using SUPERWYLBUR on OS/MVT for program development, etc.

The MVS IPO was restored to "IPORES" and "IPOCAT" and maintenance applied with SMP4, which worked fairly well under MVT with the "NORECOVERY" option. When it was time to begin initial testing of MVS, a VM Directory entry was created for the MVS system. Procedures were put in place to share only those 3350 strings between VM and MVT's CPU's which were necessary to run a representative MVS system under VM. A method for submitting jobs from MVT/SUPERWYLBUR to MVS under VM was devised using one of the shared packs; there was no CTC (Channel to Channel Adapter) available at this time.

MVS was brought up under VM and test jobs were run. At times, ACP testers noticed response time impact when MVS was up and running so some tuning and priority adjusting was done. Jobs submitted for testing under MVS/VM were limited to quick running tests; specifically running SMP PTF Apply's, etc., ran far too long to be practical.

As testing progressed, the MVT operators became involved. Initial familiarization began with the installation of a 3277 terminal near their MVT operator consoles to be used for MVS operations. While MVS was up under VM, this 3277 was "attached" to the MVS Virtual Machine as a real 3277. As time allowed, each operator would run through a familiarization procedure set up by our Procedures & Training group. This "hands-on" experience was supplemented by the IBM MVS Operator ISP (Individual Study Program).

Eventually, MVS was kept up in test mode under VM continually. Applications and Systems programmers could submit jobs at anytime from SUPERWYLBUR but could not monitor their execution or retrieve print. The jobs would run under MVS/VM and go through the VM Spool facility to be printed by VM.

When initial testing was complete, exhaustive system testing began. A production MVS system was built from the IPO system and, after being checked out under VM, was brought up in native mode as time permitted, by the OS operators. An additional SUPERWYLBUR procedure was established to allow users to either submit "small" jobs for MVS under VM or "big" jobs for MVS in native mode on the 3033. Also, selected re-runnable production jobs were scheduled to run for benchmark purposes. Jobs that ran under MVS with no errors were marked as "OK for MVS". Those that did not, such as the S0C4 abends in COBOL programs, etc., were assigned to an applications programmer for correction. We did not run every production job under MVS before cutover; some errors were found thereafter and were fixed in the same method. Strong management support at this time prevented a fallback to MVT as only a "handful" of jobs needed to be modified.

2. After Conversion

Once cutover was complete, MVT was retired. When "CPU swaps" dictated by the online reservation system caused MVS to "steal" VM's 370/168, MVS performance suffered and, of course, VM wasn't up at all. This condition occasionally exists today and, as a result, can cause the same problems. However, VM and MVS have grown considerably.

a. Current Environment (Foil 2)

VM and MVS today have come a long way in the past three years. Both systems usually run on 16 Megabyte IBM 9083 processors and talk to each other through the CTC link. VM's RSCS routes jobs and JES2 commands to MVS and receives output back from MVS; JES2/NJE receives the jobs and schedules them for running and routes the output back to RSCS or, if the CMS user selects, to be printed by MVS.

CMS users can also use their 3278-5 terminals to "dial" to any of the ACP test systems running under VM at the time. The programmer updates his or her program under CMS, routes it to MVS for assembly/compile updating an ACP object library (shared with VM), and then retrieves the listing back. Now, through CMS EXEC's the programmer gets his program to the desired ACP test system and then tests his program in the ACP virtual machine, with the terminal just used for CMS acting as a "reservations agent set".

Currently nearly all applications programmers use CMS instead of SUPERWYLBUR for program development and, as a result, any loss of the VM real machine will effect their productivity. MVS Systems Personnel use TSO and/or SUPERWYLBUR for MVS maintenance and then bring up the "IPO" system, which is our MVS "backup/test" system now, under VM. This procedure is used to check out PTF applications, I/O SYSGEN's, etc. before putting into production MVS. The impact on other virtual machines is not what it once was and, with VM/HPO, a limited MVS system runs quite well.

MVS Production now consists of a massive VTAM/NCCF/SNA network, critical ACP support jobs, 50-70 SUPERWYLBUR users during the day, and limited TSO. Nineteen initiators are available at all times, with some dedicated to the Cross Domain Network Data Transfer Facility (CDNDTF) so they can be used immediately when files are to be transmitted between centers.

b. MVS or VM or Both???

From the above, you can see that VM and MVS are both important systems at UAD and loss of either would effect productivity, schedules, etc. As mentioned before, however, occasionally the online reservations system may need to "steal" either the MVS or VM real CPU. Fortunately due to advances in hardware reliability this rarely occurs, but when it does a decision needs to be made:

"Should we take MVS' machine and leave VM up? Or should we take VM's machine and leave MVS up? Or should we take MVS' machine and bring up production MVS up under VM?"

Three questions. Needless to say choosing any one would cause an impact on at least some users. In the rare cases where this decision needs to be made different answers are received from effected groups. The VM/CMS users of course believe their system must not be brought down; MVS batch and SUPERWYLBUR users feel the same. The route usually taken depends on how behind the MVS backlog of critical ACP support is. If all the processing power of the machine is needed to meet a particular deadline then VM will give up the machine for native MVS. If this is not the case, then MVS production will be brought up under VM.

c. MVS Production under VM

Before giving MVS' CPU to the other system, MVS is brought down as clean as possible and non-rerunnable jobs cancelled as necessary. (The state of jobs running on MVS will also affect the decision to bring down the system.) Then the MVS hardware is switched to the VM CPU and the MVS VM logon procedure is performed. Tapes and printers are "attached" as needed to the virtual MVS machine as well as the 3705 for NJE/SNA/TSO/SUPERWYLBUR. The system is IPL'd and brought up with only 4-5 initiators for running critical jobs and others as needed. Once the system is up with limited batch work, operators tend to "forget" they are running under VM except for the longer-running batch jobs and occasional longer response times to MVS commands entered. CMS users and ACP testers note an impact too but agree it is better than none at all.

Once the original problem is resolved which resulted in the MVS/VM decision, MVS can be "migrated" back to native on that CPU. The MVS under VM system is quiesced, initiators and printers drained, and, when "ALL AVAILABLE FUNCTIONS COMPLETE", MVS is brought down and logged off of VM. Hardware is switched away from the VM machine to MVS' new machine, and MVS is brought up once more by itself.

3. Conclusion

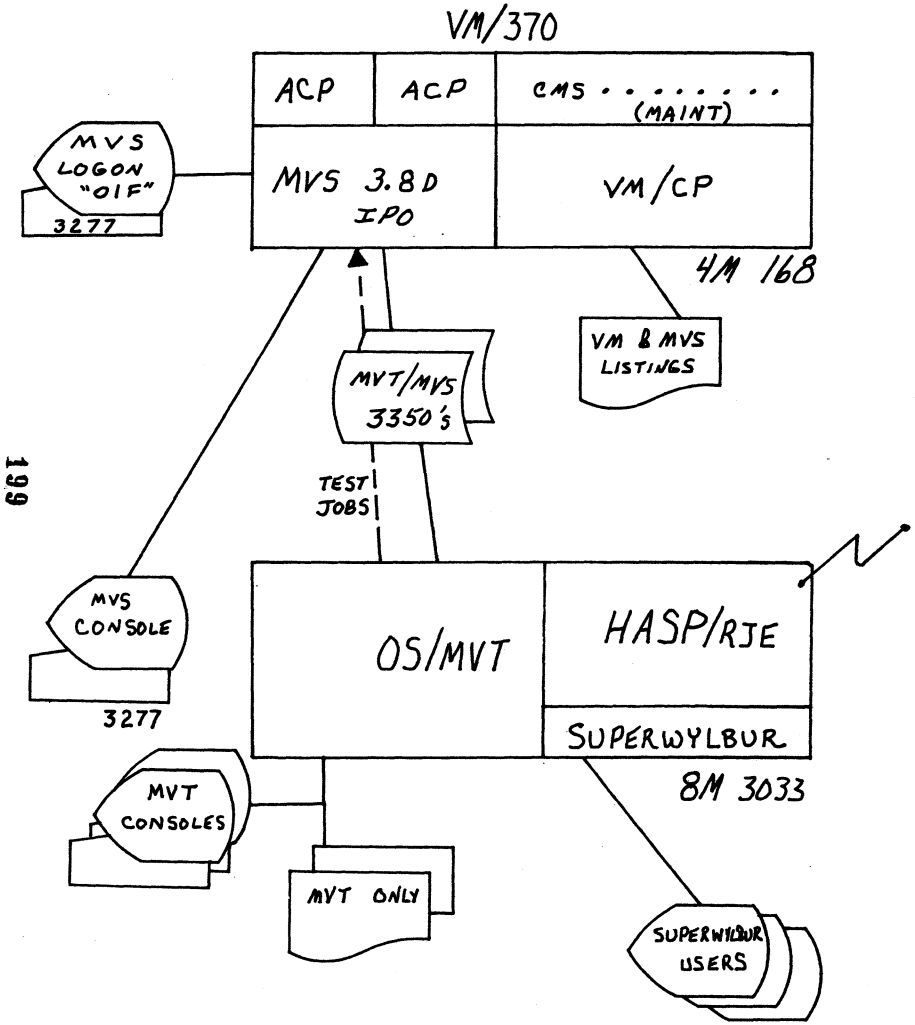
Admittedly, UAL Denver's operation is not typical. It does not reflect the medium or small shop's need to address the MVS under VM "all the time" option. The VM/HPO - MVS option offered by IBM addresses this problem and should be considered by these installations. My report today was to emphasize that VM and MVS are both critical systems to our operation and only described "emergency" procedures for running MVS under VM. Smaller shops, particularly those converting to MVS with one CPU, should use VM for conversion. OS/MVT and VS1 run nicely under VM and this environment provides facilities for testing and installing MVS without the need for fully dedicated time, except when timing and other considerations are applicable.

The single CPU shop needs to review several items before "burying" VM after conversion to MVS. Among these are:

1. Should CMS be our program development and interactive user package, in lieu of TSO or...?
2. If we "bury" VM, then all future MVS maintenance/checkout will have to be done on dedicated time.
3. Can we afford the overhead of VM and other users on our production MVS system? Do the benefits gained from this environment outweigh the results of this degradation?
4. If we keep VM and MVS, then we need to maintain two SCP's, which means more systems programmers, more things to go wrong...
5. With the decreasing cost of hardware accompanied with increasing capacity, can we live with this environment until we can convince upper management we need two CPU's?
6. Can my DASD configuration support VM and MVS (paging, minidisks, production and test libraries, etc.)

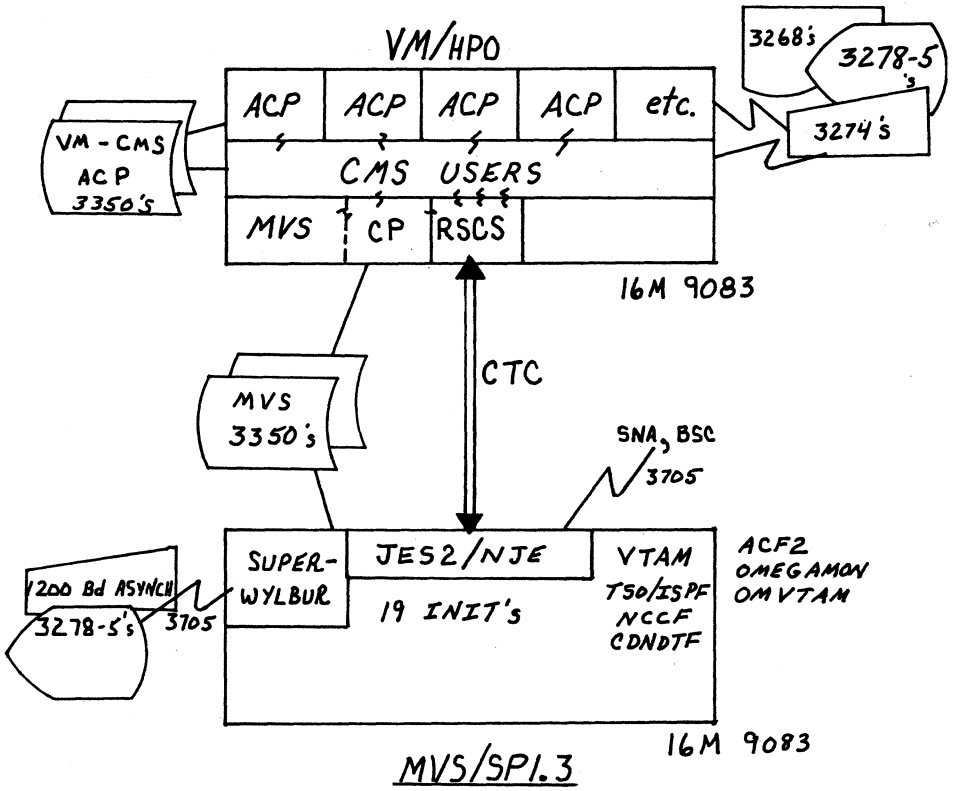
There are other concerns but before throwing VM out as only a conversion tool, users should take a close look at the advantages and disadvantages of both options, with a look 3-5 years hence, before making the decision.

B401 MVS & VM at UAD at MVS Conversion



FOIL 1

MVS & VM at UAD B401



FOIL 2