SLAC-PUB-2782
July 1981
(N)

MOVING TO A TOTAL VM ENVIRONMENT*

T.Y. Johnston
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

August 11, 1981

FUNCTION OF SLAC

The Stanford Linear Accelerator Center is a single purpose laboratory operated by Stanford University for the Department of Energy. Its mission is to do research in High Energy (particle) physics. This research involves the use of large and complex electronic detectors. Each of these detectors is a multi-million dollar device. A part of each detector is a computer for process control and data logging. Most detectors at SLAC now use VAX 11/780s for this purpose. Most detectors record digital data via this process control computer. Consequently, physics today is not bounded by the cost of analog to digital conversion as it was in the past, and the physicist is able to run larger experiments than were feasible a decade ago. Today a medium sized experiment will produce several hundred full reels of 6250 BPI tape whereas a large experiment is a couple of thousand reels. The raw data must first be transformed into physics events using data transformation programs. The physicists then uses subsets of the data to understand what went on. The subset may be anywhere from a few megabytes to 5 or 6 gigabytes of data (30 or 40 full reels of tape). This searching would be best solved interactively (if computers and I/O devices were fast enough). Instead what we find are very dynamic batch programs that are generally changed every run. The result is that on any day there are probably around 50 to 100 physicists interacting with a half dozen different experiments who are causing us to mount around 750 to 1000

---

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.

tapes a day. This has been the style of computing for the last decade. Our going to VM is part of our effort to change this style of computing and to make physics computing more effective.

COMPUTING HISTORY AT SLAC

The first major computer installed at SLAC was a 360/91 in October 1968. This system was run under HASP and the first interactive system on it was CRBE. CRBE was replaced with WYLBUR in 1970. In late 1973 and early 1974 2 370/168s were installed. ASP was brought up to control the triplex made up of the 2 168s and the 91.

During the fall and early winter of 1975 SLAC was a JES3 field test sight. Our intent was to convert to JES3 in the 1977/1978 time frame. During 1976 it became increasingly evident to us that we did not have a good way of getting to MVS. We had significant enhancements to ASP which would require integration into JES3 before putting JES3 into production. We could not spare a 168 from service so as to provide adequate test time and we did not have the option of acquiring another machine. At Share 47.5 in St Louis in December 1976 we heard for the first time of the Amdahl hypervisor which was later marketed as VMPE. This appeared to be the answer to our testing problem as it allowed us a way to get extended test time without taking a 168 out of batch service.

The intent was to use VM as a test vehicle so as to allow us to test MVS concurrent with SVS production. We like most other people "knew" that VM was much too expensive a system to run "production". We were, however, not enamored with the idea of going to MVS/JES3. It was going to be a lot of work and when we were through we weren't going to have much more function than we already had. We would have a more reliable system that would support new hardware, but little in the way of improved user capability. We felt that we needed to support many true interactive users and did not know how we would be able to accomplish that goal.

During the spring of 1977 another extremely important event occurred. A shiny-pated evangelist taught us that VM was not expensive. The running of guest operating systems was expensive, but who needed those massive systems? We ran tests and confirmed that in fact VM was not expensive. Many of us immediately bought into the idea of building a batch system on an interactive system. We had seen the marginal success of interactive systems

built on batch systems and felt that batch was basically an interactive session without a connected user. We also had considerable expertise in batch and batch scheduling.

In July 1978 we obtained a version of the Amdahl hypervisor. We modified it and SVS to support SVS. Prior to January 1979 VM development time was limited to a couple of hours a day three or four days a week. By January 1979 we were running VM 24 hours a day on one machine along with a hypervised SVS system as an ASP main that continued the OS batch service.
We now had a system where we could start development. More of the systems programmers started learning VM. Before many months had passed several of our more adventuresome users were also trying to use VM. By early 1980 we had several groups of users who were helping us to find the most important functions to add to the system. We limited the number of users that we would allow on VM because of capacity problems.

In February 1981 a 3081 was installed and we finally had the capacity to try and convert our user community from VS to VM. How were we going to get the user community moved over? I would like to discuss this in two parts. First, the functions that we felt were necessary before we could move the users over, and second, the strategy for moving the users.

NECESSARY FUNCTIONS

Some of the functions which will be discussed are unique to our type of a research laboratory, others are merely functions which we felt should be part of any system. Some of the more important changes are:

University of Maine support for the 370 PER hardware. Fundamental for systems programming and very useful for sophisticated users.

Batch support. We obtained INTEL's batch monitor and modified it extensively to provide priority queueing, and interfacing with other SLAC services such as SETUP.

Tape support including SETUP. This was our largest single effort. It includes SETUP scheduling (currently integrated with ASP setup scheduling), Multiple Console Support, Multiple volume support, tape usage and error statistics records, extensions to FILEDEF for NL and BLP tapes, and extensions to get output tapes tape marked on abnormal termination.

Symbolic debugging support. Another very large effort. Our world is a FORTRAN H Extended world and we need the ability to debug interactively. Several years ago SLAC modified FORTRAN H to put out SYM cards and built an interactive debugging system under SVS. We have now built an interactive debugging environment for the user under VM. This includes fully symbolic debugging of operational codes (not re-compiled and re-linkedited such as with the PL1 checker). It includes symbolic break and watch points and the ability to do free run. Some production runs are done under the debugger in free run mode. In case of failure a symbolic dump of interesting areas will be created.

Removal of GLOBAL. The limit of 8 items on a GLOBAL has been a very serious constraint to our physicists who are used to having many separate subroutine libraries included in their programs.

Miscellaneous changes. Minidisk backup service, extensions to CMS for multiple unit record devices, temporary minidisks (not tdisk) that live across IPLs and can be shared, primitive file sharing, and many small changes.

One of our goals in bringing up VM was to minimize the changes to both CP and CMS. We spent a very considerable effort in putting together a good maintenance system (based on the CTS system on the Waterloo tape). We do not have many modifications to CP and CMS. It takes us a couple of man weeks to bring up a new PLC for CP and CMS. The major new functions have been implemented in service virtual machines. These are both faster to implement and easier to debug and understand.

USER CONVERSION STRATEGY

SLAC is made up of 38 groups in 3 divisions - the Administrative Division, the Technical Division, and the Research Division. The Research Division accounts for over 95 percent of the computing work at SLAC and we have concentrated our conversion efforts so far on the Research Division. It is made up of 18 experimental groups. Each of these groups is quite autonomous and self contained. Each such group consists of from 10 to 50 physicists and support personnel working on one or more experiments. Certain of these groups account for very significant percentages of the laboratory's computing. In fact the top 10 groups account for 94% of the laboratory's cpu utilization.

With this utilization in mind our first set of objectives has been:

- o Get together a group of key users who were willing to work with us in initially exploring VM service.
- o Convert a couple of small but sophisticated groups to VM.
- o Assign personnel to help convert one of the large but diffuse groups over to VM.
- o As a part of all of these processes build up a set of tools that improve the usability of the system.
- o Work with other large groups to get them converted to VM.
- o Don't prevent anyone who wants to convert to VM from converting on their own.

CURRENT STATUS

Sufficient users have been converted to VM such that we were able to power down the 360/91. We have completed our first set of objectives and are now starting to plan for the next phase of the conversion. This phase will emphasize getting additional small groups over to VM. Certain of our groups have special problems which require solving before they can move. We will look at the solution of these specific problems.

We do not expect to complete the conversion to VM for several years. The major physics groups are both quite sophisticated and have sufficient resources to do most of the conversion work. The other groups around the laboratory are not nearly so self sufficient and it will take considerable effort on our part to help these people convert. That will take time.

SOME PERSONAL OBSERVATIONS

Why did we attempt this major change in our computing style? Why did we go through this effort and not go along with the main stream of IBM computing? Was it worth it? Would we do it again? What are the benefits of this change? How does that effect the inexperienced user versus the sophisticated user? Let me address these various issues by a series of personal observations.

1. We felt that the direction of IBM batch computing was not a direction at all. It provided the same old functions and pain and no improvement in the fundamental need to provide the user better computing.

2. The VM approach allowed us to change our environment towards a more productive world for both the systems programmer and the casual user.

3. I have observed the Systems Programmers in my installation become significantly more productive because they have tools to solve their problems and systems that don't stand in their way.

4. I have seen myself change from using WYLBUR (a very fine text editor) to using XEDIT and SCRIPT (which in many ways are quite inferior) because it was easy.

5. Would we do it again? Emphatically, yes! The only other choice would be to leave IBM for systems like VMS on the VAX's, but DEC does not provide us the computing power that we need.

6. When one talks about the benefits of VM/CMS versus OS they generally catagorize into two major areas: Removal of ugliness and ability to solve problems more naturally. We did not think that it was reasonable or effective to require our users to know both an interactive command language and a batch command language. Consequently, our approach of creating a powerful VM batch meant that the user could always work in a single command language. We felt that the transition from test to production should be simple and rapid.

7. Although I have used computers for over 2 decades, I am now as a manager, a casual computer user. What I have found is that I can accomplish a great deal of work with very little study. I have never read the XEDIT manual, but I can do most of the things that I need to do. I have spent about 20 minutes in learning SCRIPT, but I can create reasonable documents. These are the kinds of systems that we need! Ones where with a modicum of intelligence and little effort one can become effective, and where the expert can perform magic. I believe this world is better exemplified in VM/CMS in IBM than it is in MVS and its various JES systems.

CONCLUSIONS

In the last 6 months we have made a significant transition from a predominately OS installation to a predominately VM shop. We still have a long way to go. We have added a considerable number of necessary functions, but we have spent very little time in smoothing the rough edges of the system. We are only now starting to spend any considerable effort on improving the human interface to the system.

Since our laboratory does not have a long experience of VM usage, we have had to face the problem of user education as well as system conversion. Each user group has a computing guru. What has been mandatory has been to thoroughly train that individual before any group starts the conversion process. That person ends up a pathfinder for the others in the group. VM has a very different style of computing than what one does under OS. It is probably not smart to attempt to do things in VM just like you did them in OS. Further, the worst world to live in is both worlds. Trying to straddle the fence is generally painful and confusing, and one ends up with the worst of both worlds. We have also on occasion assigned persons from systems and user services to help the large groups convert. This has been very valuable.

Although, we are still in the world of maintaining multiple operating systems our maintenance efforts for the old system are minimal. We are moving towards a much more understandable world where the individual user is more effective. We believe that this is a direction that many organizations could find reasonable. The biggest impediment to this move is the lack of support within VM for many needed functions. The Waterloo library certainly helps out in this area, but there is clearly the need for many more software products to support the VM community. Our competition at SLAC is not from MVS but from VMS (the VAX 11/780 operating system). VMS is a much newer system than CMS and is in many ways a nicer system.

Several years ago we decided that the most important element to optimize in the user/computer relationship was the user. We have seen progress in this area with VM over our prior system. The sophisticated VM user can be very effective. Our job now is to try and make more and more of our users effective. This requires that we provide a variety of tools and capabilities. Some of these are:

- o User friendly systems that are readily adaptable to the computing needs of the individual.
- o A rich variety of functions to solve the varying computing needs of our users.
- o Road maps and tools that allow a user to take advantage of the system (rather than being taken advantage of by the system!)
- o Hardware as well as software that helps one rather than components that create impediments to our accomplishing our tasks.

VM/CMS provides the best tools available from IBM today for solving these problems. We have come a long way, but we still have a

long way to go. We believe that our approach is a step in the right direction. We can only hope that other steps will follow which will make our efforts a first step towards an increasingly more effective and satisfying environment rather than a minor aberration in a continuing journey towards regimentation and mediocrity.