Section VI -- Operations Management

- Requires Operations management effort

To maximize performance on a multi-user VMS system requires an active management effort over the usage of the system. Beyond the technical functions (tuning, doing backups, etc.) this management effort must influence the behavior of users by:

- managing the workload, when and where it is performed

- promoting efficient, effective usage practices

- promoting development and/or acquisition of efficient and effective software

Management in this sense requires a sensitive, business-like approach. The purpose of the computer is to effectively fill an organization's need for computation; its users should not have to function at the convenience of the datacenter. On the other hand, the computer represents a significant cost, for which results should be maximized. And that maximization will not occur if users are not sensitive to the cost of the computer resources they are consuming.

Ultimately, the computer's purpose is to increase people productivity. Fortunately, improving the computer's productivity does not usually have to be done at the expense of the productivity of its users. When this is not the case or when the cost of changing to a more productive mode is significant, a cost benefit analysis is required. The productivity improvement may not be worth the cost of implementing it.

The usual problem in VAX/VMS environments is that the user community is not sensitive to the costs of machine usage. The individual user is not aware of the impact of his usage on other users; he only feels the impact of other users, which he blames on the machine and its keepers. The data center manager is charged with the job of making users happy, but isn't allowed to directly control their activities. Yet the only way to better performance is to alter those activities.

Implications

In a business organization the driving force is (or should be) money. Users who are not sensitive (either directly or by virtue of the sensitivity of their direct management) to the costs of computer usage will make no efforts to make their usage more efficient. Nor will there be any reason perceived by the users, and therefore no pressure to obtain, improved software.

good point

THERE WILL BE NO SIGNIFIGANT PERFORMANCE IMPROVEMENT IN OPEN VAX ENVIRONMENTS UNLESS THERE IS A RESOURCE CHARGEBACK SYSTEM COUPLED WITH A MANAGEMENT COMMITMENT TO HOLDING USER GROUPS RESPONSIBLE FOR COST JUSTIFICATION AND MINIMIZATION.

Chargeback systems must differentiate charges in a manner designed to promote desired behavioral changes. Charge the most for what is demanded the most. Do not attempt to set charges by any accounting justification (except the grand total, which in internal cost centers should equal total costs.)

Workload Management

Among Multiple CPUs

VMS provides mechanisms to automatically distribute the workload among CPUs. These mechanisms represent, at best, an inadequate solution.

- allocation is based on number of processes
- require entire storage system to be open to all CPUs, which introduces large performance costs.

If the user environment is understood -- the demands for resources, the timing of those demands, and the relative priorities and importance of the different applications -- and frequent monitoring of that load is performed, much more intelligent and effective load balancing can be effected by specific allocation.

For performance, a mix of activity types should be maintained on each CPU -- I/O intensive, CPU intensive, highly interactive, etc.

Smaller numbers of large CPUs are easier to manage, provide smoother operation, and are more cost effective. Two 8600's will be a much better solution than ten 750's.

Use cluster technology for redundancy.

Load Balancing By Time

Computers can do exactly the same things at 2AM as at 2PM. Despite this, much of the work that is normally being done at 2PM need not be done then, nor does it need human interaction while it is running.

Batch work is anything that doesn't need interactive, human inputs as it proceeds (or, if it currently does, could be set up so those inputs are specified before it started).

There is nothing more wasteful of <u>people</u> time and productivity than sitting in front of a screen watching the cursor blink. If it can be run in batch, it should be if any significant amount of elapsed time will be required for it to complete.

If it isn't needed now, why is it being run interactively? A range of priorities must be available for batch work.

Users will not use batch if computer usage is free, the costs meaningless, or there is not a significant discount for batch. If batch is not used, the overall usage of your VAX will never exceed 20%.

Assuming all work that should be run in batch mode is being run in batch mode, peak interactive loads must be analyzed and efforts made to rearrange work patterns.

Lunch hours are generally lightly used. Set a lower charging rate for this period.

Can staggered work shifts be arranged?

Why does the payroll group only use the machine in the afternoon?

Should some user groups work a multi-shift operation?

Why is all the usage from dept. X on Mondays and Tuesdays?

Friday the load is always light. Should a reduced charge be instituted?

User Practices -- Production Users

In General

Users whose resource usage is noticeably out of line with other users doing similar work should be addressed. If the motivation toward better performance is provided by a chargeback scheme, users who are offerred assistance toward more productive habits will often accept it.

Users whose performance is better than average should be identified and considered for roles as trainers.

Some specific points regarding good usage habits:

Disk drives are more expensive than filing cabinets as storage devices. Printed material (ie., that which is not to be processed in the future) should not be stored on line. It is cheaper to photocopy a document then re-print it from the computer. If the costs of filing cabinets becomes too large, investigate microfilm. For access control, consider computer based index systems.

Users working at DCL level must understand and use command procedures and logical names. The less a user has to type, the fewer mistakes will be made, and the less VAX and people time will be wasted.

However, any DCL procedure used extensively should be replaced with a program. DCL is extremely inefficient.

Applications which require user parameter and input data specification, then proceed to run in batch mode for a considerable period of time based on these inputs, should have front ends which check the parameters as thoroughly as possible. Every effort must be made to eliminate wasted CPU and people time due to undetected, but detectable, specification errors. For batch mode operation, an interactive procedure should be used as the front end which accepts and checks the inputs and then creates and submits the batch procedure.

File directory organization is critical, as well as appropriate use of file system commands.

User Practices -- Software Developers

Though the basis for this discussion is the computer resource usage of programmers, the real subject is their ultimate productivity.

To write music for an orchestra one does not use an orchestra. Similarly, it does not necessarily follow that to design and write software one uses a computer.

Computers, for this purpose as for any other, should be used only to the extent they can effect a productivity increase greater than their cost. The question which should be raised is whether the interactive use of computers by developers results in an increase in productivity, and how much.

In fact, it appears that the interactive use of computers by developers leads to a signifigant decrease in productivity. To review this, let us detail the types of tools programmers tend to use on a computer:

Compilers and Linkers

As well as being necessary to make use of the programmer's end product, these tools provide useful diagnostic information. However, they also consume large amounts of computer resources.

Note that they are essentially batch functions, and their most important development product is a printed listing, of which the most valuable diagnostic tool is a cross-reference listing.

Editors

Editors can be used for 3 different functions:

-As a display and diagnostic tool. Editors are frequently used in place of source and cross-reference listings as an "electronic desktop". Unfortunately, 80 columns by 24 lines is not a large enough view area for effective review, editor "page turning" is nowhere near as fast as hand page turning, and editor searches are hardly as effective as consulting a good cross-reference listing.

-As a composition worksheet (code entry): Some people can type -- even programs -- as fast and as effortlessly as they can write. For most, however, using a keyboard is a distraction and causes a loss of flexibility. Loss of effective coincident documentation and code readability is the common result when code is composed on the screen. Moreover, 80 X 24 is hardly a sufficient reference area, and discourages proper cross checking and validation during the composition process.

Even with large area displays (windowed graphic displays) the area is still too small, operation is not as flexible as an actual desktop, and page searching and turning is more tedious.

Note that programs are line-oriented material with (for most serious work) lengths of hundreds or thousands of lines. Line editors are designed for this type of material. Text editors, with no line number referencing and preservation, are not and can not be as productive.

Code Management Utilities and Libraries

These are effective coordination and organizational tools and may be critical for any large project. However, these are not diagnostic tools.

Testing

Clearly needed, but requires care and can be wasteful of machine and people resources. Short, one step tests are wasteful as compared to intensive, wide ranging sessions. If done interactively, hard copy output is required.

Debuggers

For the vast majority of problems, these are nowhere near as productive as good, solid bench analysis. Data values and flow can usually be tracked more effectively with well placed debug code insertions and hard copy test results. (If in MACRO, the debugger is useful for this limited purpose.)

Requires large amounts of resources and tedious operator setup and control. Critical reading of source listing requires no machine resources or setup time.

Implications

There are no effective interactive programming aids on the VAX. Programming is primarily a paper and pencil job. To be productive, a programmer must have clerical support for code entry and editing and know how to use it.

Only line editors should be employed. For any material of serious length, EDT screen mode is not productive. SOS is an effective line editor. Changes (including editing commands) should be made on paper, then executed as a clerical task.

There is no substitute for source listings with cross reference. All diagnostic work should be done against such listings, not with a screen. Thorough bench running of code is an absolute must both before testing and as THE debugging technique. Cross reference listings are not merely for reference -- they must be analyzed as part of the bench testing procedure.

The debugger is strictly a last resort. It should rarely be used with high level languages and, particuarly when used with MACRO, used primarily as a means to display storage contents. Code tracking should be done by bench running.

Testing requires hard copy and is not a "find an error, fix it, recompile, relink, test again" procedure. Each testing session should be as exhaustive and wide ranging as possible given the current state of the software. Upon completion, the programmer must return to paper and pencil mode, diagnose and compose fixes for all the errors encountered prior to testing again.

The physical appearance of code -- spacing, indentation, separation and comments -- is an excellent indicator of its quality. Readability is as important as any other factor. Readability is decreased equally by a lack of these style elements as by their overuse, such as large blank areas, verbose commentary and unnecessarily long variable names.