Section V -- Data Gathering

VMS Usage Data -- MONITOR

Data gathering must be a major activity for the performance analyst. To manage performance, the analyst must know what each application is doing, when it is doing it, and how much resource usage is occurring.

MONITOR is helpful for getting snapshots of total system behavior, but has limitations beyond that purpose.

It is a snapshot. Acquiring data over large time spans and extracting information from the result is tedious.

Be certain of the representativeness of any MONITOR data before acting on it.

Not terribly helpful for learning about individual user behavior and not helpful for reviewing usage by program.

MONITOR PAGE

This display show the various statistics concerning the activity of the memory management system. Remember that two adjustments to the values displayed must be made to make them comparable with the desired values discussed in the section on tuning:

-The rates reported by MONITOR are a rate per clock second. The statistic discussed in this manual is the rate per CPU second. To convert, the rate reported by MONITOR must be divided by one minus the percentage of time the system was idle during the period the MONITOR display is based on.

-The rates reported by MONITOR include faults due to image activation, which must be excluded to get a meaningful picture of actual virtual memory transactions.

The statistics:

Page Fault Rate

This is the total number of page faults for all working sets -- the sum of hard faults plus soft faults. Since, under normal circumstances, there are very few hard faults relative to the number of soft faults, this value can be used as the soft fault rate.

SOFT = TOTAL - HARD

Page Read Rate, Page Write Rate

These values reflect the total number of pages transferred from or to disk as a result of hard faults (read) or swapper writes of modified page list pages (write). The ratio of these values to the corresponding Page I/O Rates indicate the page clustering level achieved by the memory system.

Clustering on the write side should approach the value set for MPW_WRTCLUSTER for the system. Levels considerably below this indicate one of three problems:

-The page file is too small, and because there tends to be little free space in it, swapper is unable to find large contiguous free areas within it.

-The page file itself is fragmented.

-There is significant mapped section file activity.

Clustering on the read side is seldom very high -- usually about 5 to 15 pages per I/O. Values significantly higher than this may indicate:

-Most read I/O's are image activation I/O's for large images (where good clustering is often achieved).

-(If the hard fault rate is high) a large, poorly organized, image is flushing itself in and out of memory because the clustering procedure is bringing many unneeded

pages into memory.

-A program is using a huge amount of virtual memory in a very non-local fashion. This is a very serious problem, as performance for all software will be harmed. This can be further diagnosed by looking for a particular process incurring a high hard fault rate.

Page Read I/O Rate

This is the hard fault rate. HARD

Page Write I/O Rate

This is the rate at which the swapper is doing I/O's to write pages from the Modified Page List to the page file(s) and mapped section files. A high value (> .5/sec (780/785/8200); proportionate values for other CPU's) indicates either MPW_HILIMIT is too small, or (more commonly) some application is either using memory badly, or mapped section files are in use.

Free List / Modified List / Demand Zero / Global Valid Fault Rate

Rates of various types of faults. Note there there is double counting among the categories (eg., a global page may be found in the free list) so the sum of the numbers is not meaningful.

System Fault Rate

See the tuning section for a discussion of the significance of this statistic.

MONITOR IO

A useful collection of statistics showing I/O rates and the important memory system rates in the same display.

Direct I/O Rate

Direct I/O's are I/O's to disk and tape devices. Typically they represent normal data transfer I/O's, but some functioning commands for special equipment (SIMAX control transfers, terminal port maintanence commands, etc.) are also counted as DIO's.

In most situations, this value, plus the I/O's from the memory management system (page read and write I/O's, swapping I/O's) is the total disk access rate for the system.

Buffered I/O Rate

Buffered I/O's are terminal I/O's, mailbox I/O's and Disk ACP control functions (file system transactions such as creates, deletes, opens, etc.). This statistic is not terribly useful to gauge terminal I/O activity levels, as the number of bytes transfered must also be known to properly interpret this value.

Mailbox I/O Rate

The only resource consumed by Mailbox I/O's is CPU time. Intensive inter-process communication by means of mailbox I/O's is a very wasteful technique, but can only be fixed by changing the offending software. That software will show up as very CPU intensive.

Logical Name Translation Rate

An activity which only consumes CPU time -- software which does excessive translations will show up as an excessive user of CPU time.

Window Turn / File Open Rate

See MONITOR FCP.

Inswap Rate

See the discussion on tuning procedure.

MONITOR LOCK

A summary of the activity of the lock manager. This activity represents a very significant use of CPU time and should be minimized. Much of it is generated by the XQP, and can only be controlled by reducing file system transactions. RMS also uses the lock manager for shared files; see Section IV for a discussion of better approaches to handle shared files.

If any other software is making significant usage of the lock manager, the solution is to re-design the software utilizing better procedures.

The sum of the first 3 lines of the display is the total activity. The balance of the display is used to debug specific application problems with the lock manager.

MONITOR DLOCK

This display shows how lock requests are moving in a VAX cluster. Because communication within the cluster is an expensive process, the goal is to (after minimizing the number of lock requests in general) to minimize the number that are handled remotely.

This is done by making sure each disk is used primarily by one CPU only, that files are not opened on a shared basis by processes on multiple CPU's, and that no applications use the lock manager intensively on an inter-cpu basis.

The MONITOR SCS display can be used in conjunction with the DLOCK display to determine which cluster member is originating the incoming lock requests or which is managing the resource for which outgoing lock requests are being originated on this system.

MONITOR DISK

This display shows the number of I/O's per disk drive. These number should be compared to the values shown in the table of disk drives in Section IV to determine if there is any level of contention. (These values include memory management I/O activity.)

If the values reported approach or exceed the average maximums, there is a contention problem which may need to be addressed. (The severity of the contention problem can be assessed by a MONITOR DISK/ITEM=QUE to determine the number of requests typically waiting to be processed.)

In a cluster where multiple CPUs are accessing a given disk the rates from all the involved CPU's need to be summed. MONITOR will do this if presented with appropriate recording files from each CPU.

MONITOR FCP

This display summarizes the activity of the XQP.

FCP Call Rate

The total rate of file system transactions, less window turns. High rates will be accompanied by high CPU usage (see below). The only solution to an excessive activity level is to fix the offending application software.

Allocation / Create / File Lookup / File Open Rate

The various types of functions.

Disk Read / Disk Write Rate

The level of actual I/O activity due to file system transactions. If large, adjusting ACP cache sizes may reduce the level, but be very cautious in doing so. Large levels are generally due to excessive call rates and are best addressed through improvements in appli-

cation software.

Volume Lock Wait Rate

This will go up as XQP activity increases and go down as it decreases. The value is an indication of how frequently a file system transaction must be delayed because a previously initiated transaction is in progress.

CPU Tick Rate

This is the percentage of the CPU which is being used to perform XQP functions.

This value does not include the cost of cluster communication if the functions involve a disk not "owned" by the CPU performing the function.

File System Page Fault Rate

If the ACP cache sizes are increased, and this value increases noticeably, and there is not a very large amount of excess memory in the system (the size of the Free List is generally near FREELIM) the cache size increase should probably be reversed, as it may be saving direct I/O's at the expense of hard faults (which require more resources).

Window Turn Rate

See Section IV for a definition.

Excessive values (> .50 on a 780/785/8200) indicate a disk fragmentation problem. A disk re-organization should be performed, and application software should be modified to perform contiguous best try file allocations.

Erase Rate

Erase operations are performed either when file space is allocated on a volume with High-water Marking turned on, or when a DELETE/ERASE function is performed.

If this value is non-zero, and applications don't normally do DELETE/ERASE or files are not marked for erase on delete, look for a volume which has highwater marking on. This is an expensive operation -- erase on delete should not be done unless security considerations require it.

MONITOR FILE_SYSTEM_CACHE

This displays the performance of the ACP caches -- how frequently data needed by the XQP is found in a cache, thus obviating the need for an I/O to fetch the data (but possible resulting in a hard fault I/O, anyway).

Naively, one would wish to get as high a percentage of cache hits as possible, but the more astute analysis is to think of the number of I/O's which result. This is done by taking the attempt rate and multiplying it by one minus the hit rate. Thus, if the Directory Data cache is achieving only a 60% hit rate, but the attempt rate is .1, the I/O rate for directory data is only .04 per second -- one I/O every 25 seconds. Devoting more memory to this cache would not be a good investment -- in fact, making it smaller would seem to be indicated.

MONITOR SCS

This displays the level of inter cluster communication activity. To maximize performance, the objective is to minimize the communication between CPUs in the cluster. The statistics of interest in this regard are:

ITEM=M_SEND / ITEM=M_RECEIVE

The rate at which messages are sent or received to initiate lock manager requests or MSCP disk data transfers.

The objective is to minimize this activity -- this display serves as an alert to its occurance and a determination of which CPU's are involved.

ITEM=REQUEST_DATA / ITEM=SEND_DATA

A measure of the rate of remote access requests for I/O's from locally served MSCP disks. This activity must be minimized.

ITEM=KB_MAP

A measure of the volume of data exchanged among the cluster. This is not very significant, as the overhead due to cluster activity is proportional to number of communications, not their size. However, when this statistic is summed across CPU's, it reveals the total traffic rate on the CI and to the HSC's.

MONITOR MODE

Great significance is usually given to this display, but in reality, it offers little insight into system behavior (except to summarize idle time).

Interrupt Stack

Proportion of CPU time spent "on the interrupt stack" -- e.g., responding to interrupts. This value typically represents 50 to 75% of the overhead on the system (e.g., CPU time consumed, but not charged to any process). It is caused primarily by terminal I/O, DECNET activity, and cluster activity. (This value understates the total overhead because not all the uncharged activity is spent on the interrupt stack; some occurs in Kernal mode.)

Kernel Mode

Time spent performing most system service, such as lock manager requests, I/O set up logical name translation, memory management activity, image activation, process creation, etc. If applications must perform a large volume of I/O, this value will be large; if the applications are primarily computational (and DECNET, cluster and terminal I/O activity

are low, the memory management system is well set up, there is not excessive image activation or process creation, etc.) this value will be small.

There are no values which are "too low" or "too high" -- the value is entirely due to the type of activity on the system. On many systems this component of activity is largely due to the resource consumption of the lock manager and/or excessive hard faulting, often due to process trimming.

Executive Mode

The primary user of executive mode is RMS (excluding the time it uses to perform the actual I/O to a device).

Supervisor Mode

The primary user of supervisor mode is DCL, though some application software (e.g., Oracle) does use it. Absent an application like Oracle, values of this activity exceeding 2% indicate heavy DCL activity which should be identified and the functionality replaced by non-interpretive code. (If the supervisor mode activity is 2%, one can be sure that the DCL routine is also causing high levels of kernel mode time due to image activations and executive mode time due to RMS usage.)

User Mode

Time spent executing user instructions. Again, there is no significance if this value is large or small -- it stems from the type of system activity.

Compatibility Mode

Time spent in compatibility mode. Software which executes in this mode is often inefficient, but this is due to a poor usage of VMS facilities as opposed to compatibility mode itself being inefficient.

Idle Time

Time spent in the NULL process -- available CPU time.

MONITOR STATES

This is primarily a diagnostic display and has little performance analysis application other than to guage the length of the queue of processes waiting for the CPU in heavily loaded systems.

MONITOR POOL

Used to guage levels of nonpaged pool requirements -- see the tuning section. (The SHOW MEMORY command produces a much more useable report in this regard.)

Accounting File Subsystem

VMS includes a powerful monitoring tool in addition to MONITOR. The accounting subsystem collects data about resource consumption detailed by the entity -- user, program, queue, terminal port, etc. which caused the consumption. Unlike MONITOR, the data collected is by no means a snapshot -- it is the complete record of system activity. That completeness makes it suitable for accounting purposes, but also makes it a powerful analytical tool.

This is the most important type of data, as without knowing who or what is responsible for excessive resource consumption, there is nothing one can do to remedy the problem.

The problem with this data source is to be able to flexibly analyze the rather large mass of data one ends up collecting. This is complicated by the unusual nature of the data -- rather than being a series of single point observations it is a set of data about overlapping and highly variable time spans. Moreover,

there are unusual relationships among the individual observations by virtue of the VMS sub-process facility. No general purpose tool is equipped to interpret data of this kind for analytic purposes.

VMS Usage Data -- Other Tools

Any other data gathering tool must be chosen and used with care. Avoid any that use GETJPI to scan the system frequently. One can collect much data on what VMS is doing -- much of it is "so what" data; knowing it serves no purpose other than satisfying academic curiosity. Some data gathering tool draw inappropriate conclusions:

"free memory"

"%I/O busy"

Response Time Measurement

May be useful to know based on measurement tools rather than hearsay or informal observation:

- identifies times where problems exist (ie., overloads)
- determines compliance with service delivery levels if they have been set

The problem is determining what to measure, as any particular test will vary differently across different loads from another test across the same loads. A test mechanism must be chosen which carefully emulates (or is) the actual application for which response time is critical.

Response time is composed of:

Network delay
Measuring this is not a VAX performance issue

CPU and I/O processing time
Any task takes some CPU time and, possibly, I/O. Even on a totally empty system there is a non-zero response time. Many times complaints concerning response are really a reaction to inefficient software.

CPU and I/O contention delays
With multi-user loads on computers -- which can only do one thing at a time -- there will invariably be times when there are queues for one at a time resources. The point of measuring response time is to measure this component.

The test tool must emulate the application software -- similiar use of CPU, ratio of CPU to I/O, and similar type of I/O usage. Nothing is better than a representative run of the actual software.

A regularly scheduled batch job, running at a similar priority as normal users, with a record of its elapsed time being made, is generally satisfactory as a measurement procedure. If there is a lot of terminal input in the actual application this may not be the case. The fact that output will be to a disk file is usually not significant:

- priority boost not all that significant on larger systems
- although the disk may be faster, the real issue is comparison between runs

The data recording and reporting should show worst cases, averages, and variances.

The exercise is pointless unless its purpose is to follow up the measurements with efforts to explain and correct times of poor performance.

Data from Users (QUALITATIVE)

Gathering all the statistics in the world is of little value if one does not explain why the statistics are what they are.

It is not enough to know statistically what a program or user is doing -- if one is to make improvements one must know actually what it is doing.

To get this information requires tact -- one must be able to communicate with users on a helpful, non-adversarial, non-judgemental basis. It also requires stealth, defined in this case as:

```
SET PROCESS/PRIVILEGE=READALL
```

