Introduction

This procedure is designed to direct the implementation of proper tuning on most general purpose, multiuser, VAX/VMS systems with 4 megabytes of memory or more. It does not necessarily apply to smaller systems (730's, 725's, or MicroVaxes) but may work on such systems in some cases.

In some special situations it may prove incorrect. The two most frequent cases are:

- The presence of a real time application requiring response not exceeding 15 milliseconds at all times. This condition can seldom be acheived with assurance under VMS with general work running, but any swapping activity can introduce occasional periods of dedicated system level activity in excess of 15 milliseconds.

- The presence of poorly written software, usually where there are unnecsssarily large numbers of processes which tend to be inactive except for very brief periods of activity every minute or so. Generally, this condition is improved if such processes are made non-swappable, but other aspects of this type of situation yield generally poor performance under VMS.

Initial Setup

Starting values for system parameters and some process defaults relating to memory management must be chosen. The following list describes the relevant parameters. Note that these numbers will be "wrong" as final values in most situations. They should not be installed unless the user intends to employ the tuning procedure described in the following pages.

The table also describes relationships among the parameters. These should continue to be observed when changing any particular value.

AUTHORIZE

WSDEFAULT - Set this to 200. If most working sets are generally under 250 pages, use a value of 100. If most working sets are generally over 1000 pages, use 300

WSQUOTA - Ideally, set to 2000 or more. Use a lower value (600-1000) if memory is tight and users needing large working sets are not to be serviced well during peak system loading periods.

- ~~large enough to suit the largest programs~~
- usually 2000 - 3000

WSEXTENT - Set to 3000 or more. (Parameter WSMAX must be equal or greater than largest WSEXTENT.)

- usually 3000 - 4000

PQL_DWSDEFAULT- Set these values to the same values for the corresponding parameters listed above.
PQL_DSWQUOTA
PQL_DWSEXTENT

} apply to subprocesses (is not the AUTHORIZE'S VALUES but PQL VALUES)

Batch Queues - Do not set explicit values for WSQUOTA, WSEXTENT or WSDEFAULT for any batch queue. Allow the values set via AUTHORIZE to be applied.

SYSGEN

1056 FREELIM - Set to 1000 + (100 * (# megabytes of memory - 5)). Do not exceed 2000 as an initial value (2500 on a VAX 8550/8600/8650)

1267 FREEGOAL - 1.2 times FREELIM, but never less than FREELIM + 200

2556 BORROWLIM - FREELIM + 1500 (780/785/8200)
FREELIM + 2500 (8550/8600/8650)

1000 DIFF

1750 DIFF

1556 GROWLIM - FREELIM + 500 (780/785/8200)
FREELIM + 750 (8550/8600/8650)

DORMANTWAIT - 240 seconds (ideal)
- 99999 (until feature fixed in VMS)

} broken - turn it off. IN Ver 4.XX

LONGWAIT - 20 seconds

SWPOUTPGCNT - 5000 (disable the feature) turns of trimming

MODIFIED PAGE STUFF

MPW_HILIMIT - .5 times FREELIM (as a starting point; relationship to FREELIM will not hold during tuning)

MPW_WAITLIMIT - same value as MPW_HILIMIT. Caution: if set lower than MPW_HILIMIT VMS will, sooner or later, crash. ← *

MPW_THRESH - Set greater than MPW_HILIMIT. Disables Modified Page List trimming.

MPW_LOLIMIT - .8 times MPW_HILIMIT, but not greater than MPW_HILIMIT - MPW_WRTCLUSTER.

MPW_WRTCLUSTER- 120 I/O SIZE TO WRITE TO PAGE FILE

AWSTIM ✓ - (780/785/8200) 200 milliseconds (8550/8600/8650) 100 milliseconds; set QUANTUM to 100 milliseconds also. } Interval Time

Dick OTTAWA'S

AWSMIN - 30 ?

PFRATH - (780/785/8200) 200 (8550/8600/8650) 400

PFRATL - (780/785/8200) 12 (8550/8600/8650) 20

Controls AWSA if in a memory bind, use large values for starting point.

WSINC - 150 200

WSDEC - 37 37

FOR OFF NORMAL WORK

The Tuning Procedure

The following procedure should be applied after the initial settings described above are implemented as soon as operational performance data is available. Generally a day or two of normal activity is sufficient. As any action step indicated in the procedure is taken, collect data again for a day or two before continuing.

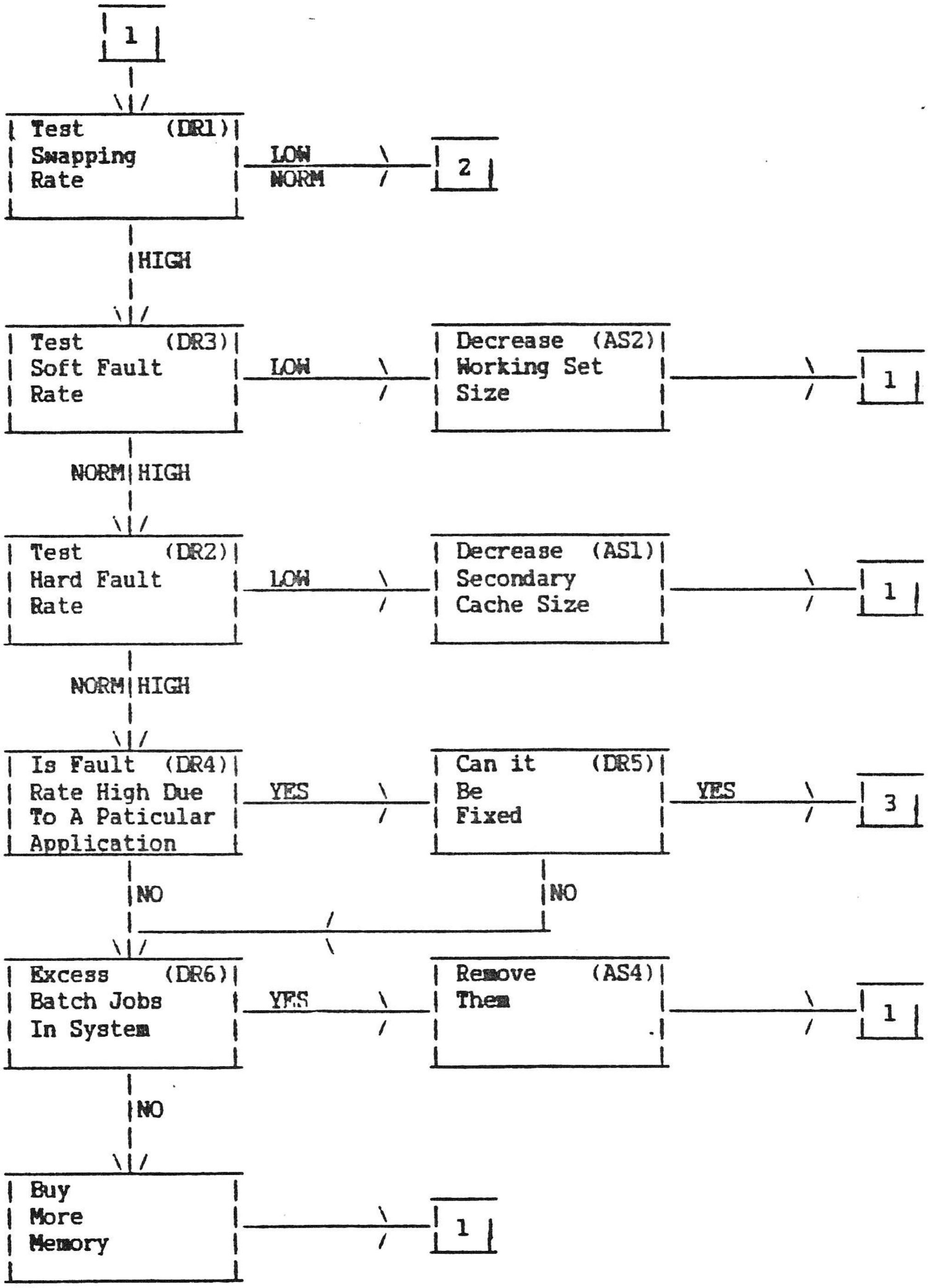
Each box in the flow diagram is either a decision rule ("DRx") or an action step ("ASx"). Each is fully explained in the text following the charts.

Decision rules which are described as being based on observed rates of certain system actions should be tested only against data from those periods of times when the systems is loaded at its normally heaviest sustained load of substantially ordinary work.

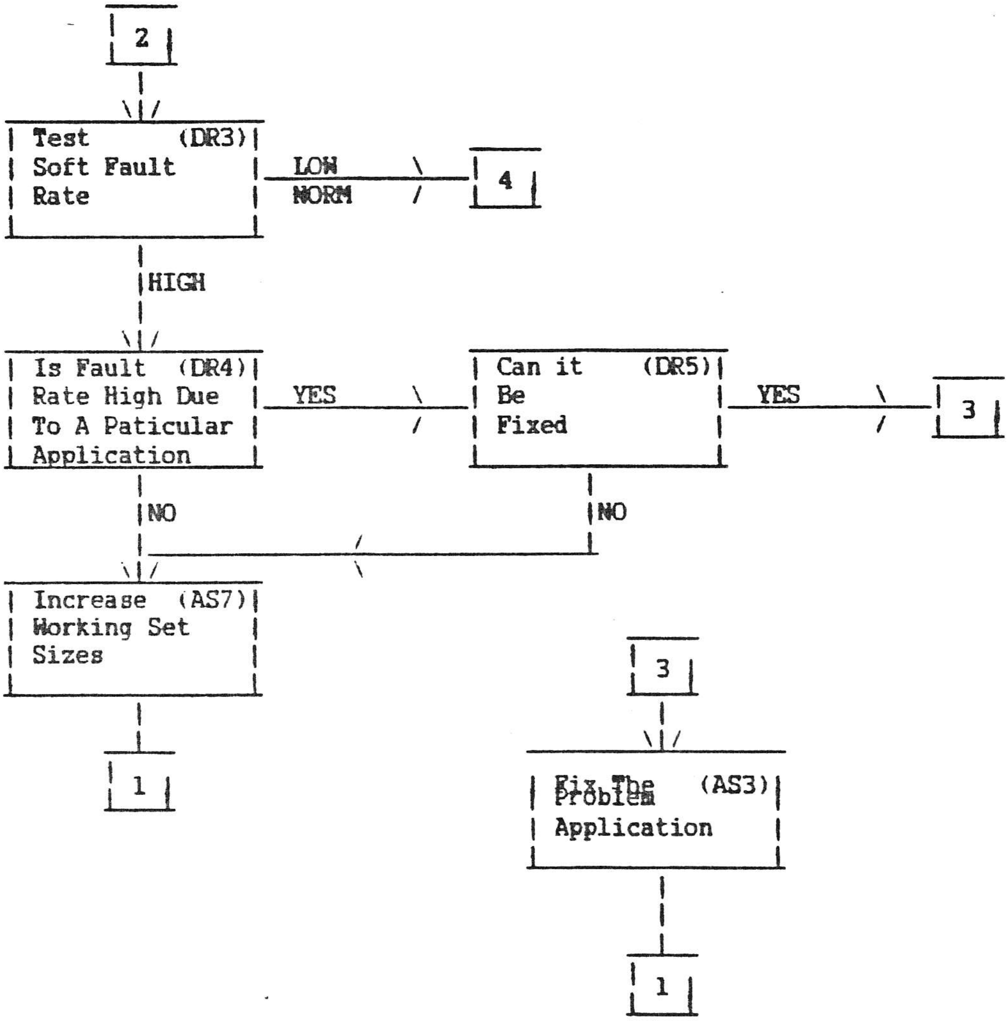
Changes to parameters should generally be an adjustment of about 10%, except in AS6, where 5% will do. Recognize that tuning is sufficiently precise if parameters are within 10% of their "optimum" values. Iterating these procedures to fine tune the system produces little, if any, performance improvement.

The procedure should normally be re-applied at 3 to 6 month intervals, or if there is a substantial change in workload or the nature of applications being run on the system.

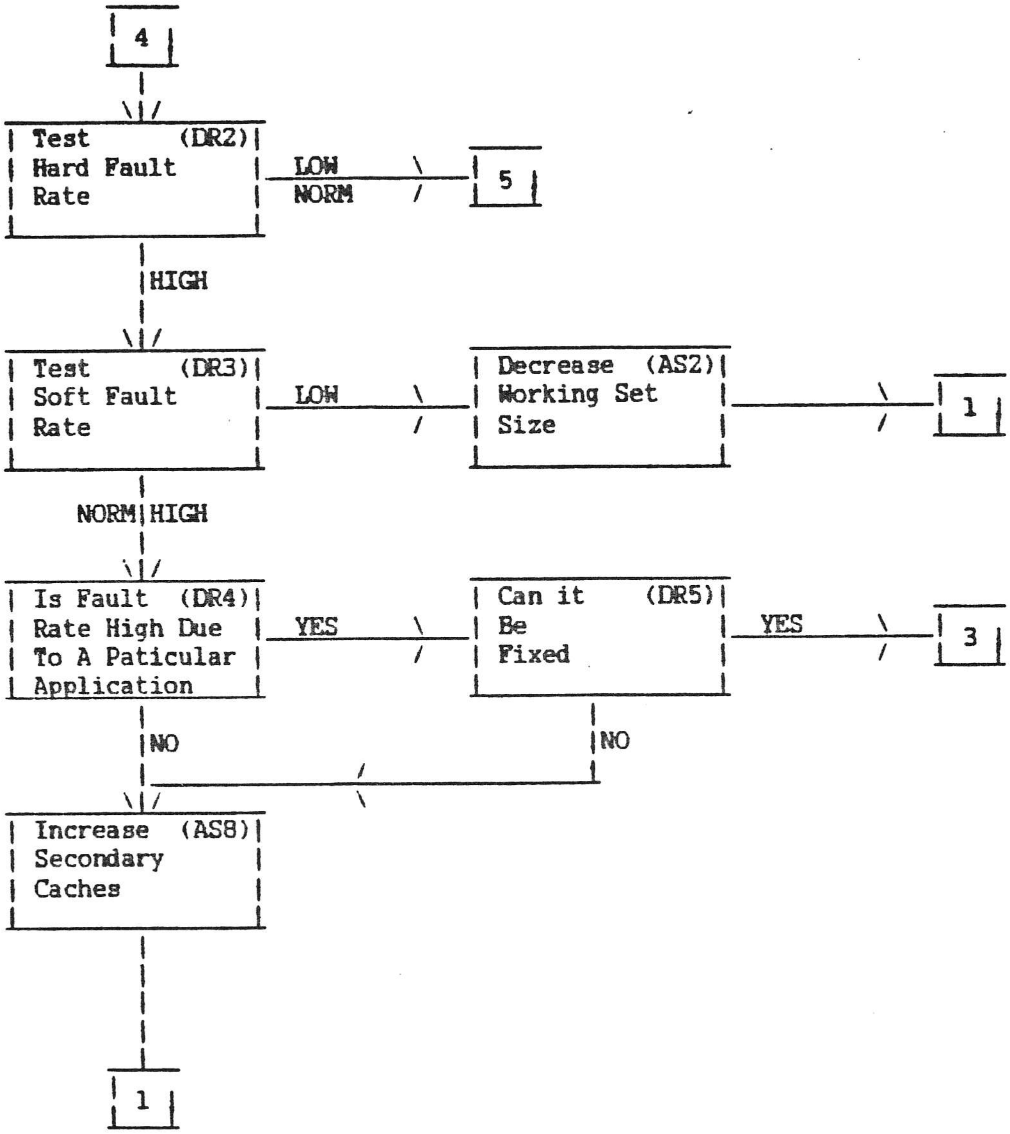
Flow Chart 1
Remedies for Excessive Swapping



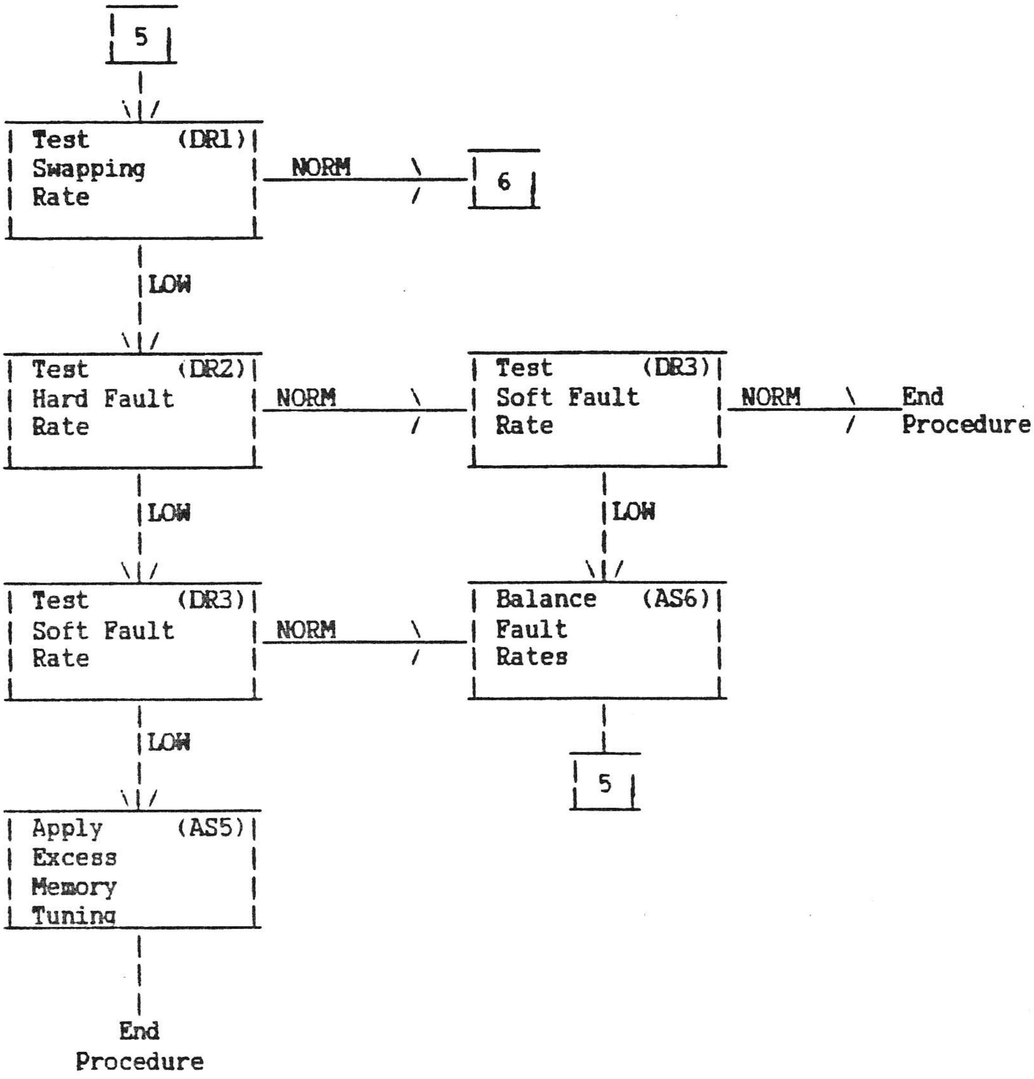
Flow Chart 2
Remedies for Excessive Soft Faulting



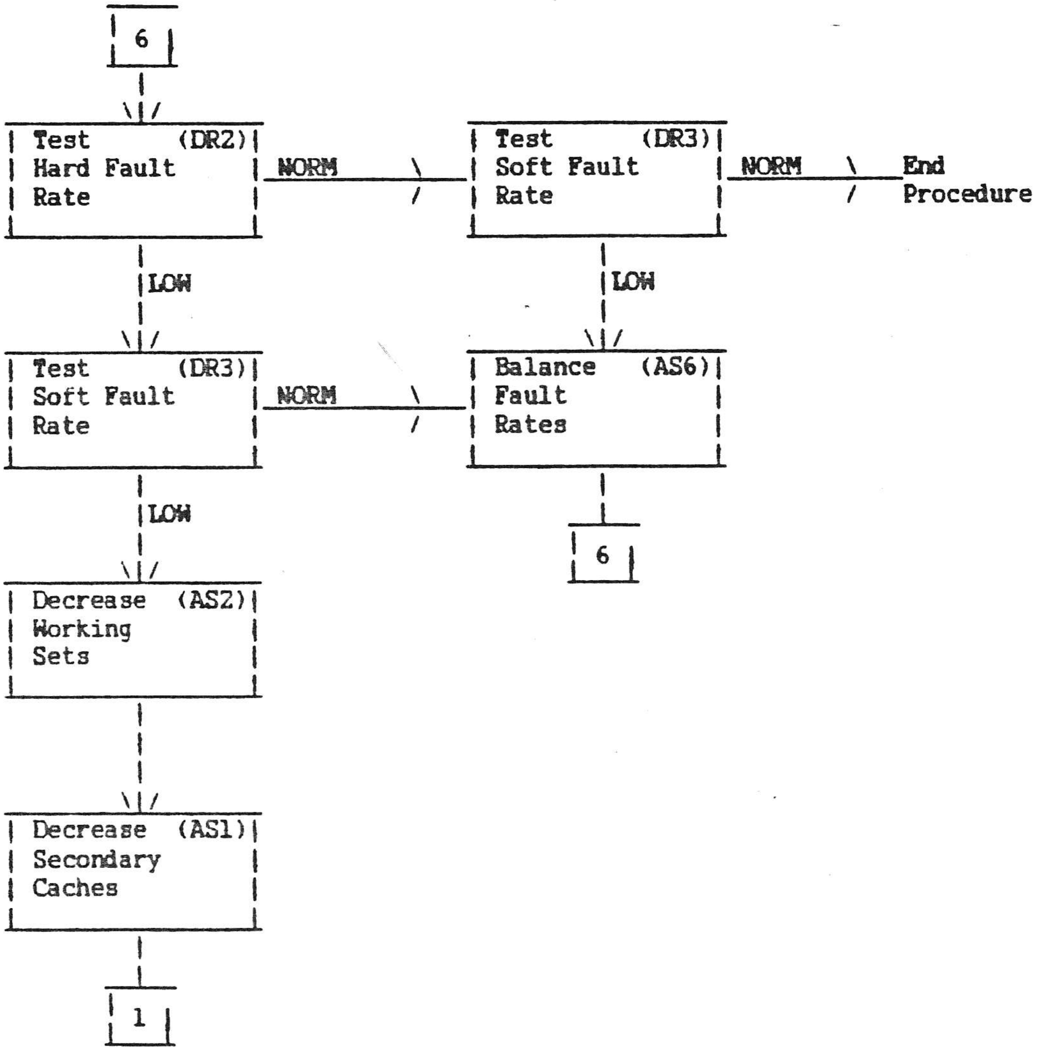
Flow Chart 3
Remedies for Excessive Hard Faulting



Flow Chart 4
Excessive Memory Check



Flow Chart 5
Balance Fault Rates


6
Test Hard Fault Rate (DR2)
NORM
Test Soft Fault Rate (DR3)
NORM
End Procedure
LOW
LOW
Test Soft Fault Rate (DR3)
NORM
Balance Fault Rates (AS6)
6
LOW
Decrease Working Sets (AS2)
Decrease Secondary Caches (AS1)
1

Decision Rules

DR1 Test Swapping Rate

A low swapping rate is evidenced by few or no swapped out processes during busy periods, and clearly inactive or dormant processes in memory. Dormant processes are usually batch jobs which are receiving no CPU time because higher priority processes are fully using the CPU. Of course, if the size of the Free List is always well in excess of FREELIM there is a low swapping rate.

SHOW PROC.

A high swapping rate is evidenced by either of the following:

-Inswap rate > .1/second (750) to .3/second (8650). The inswap rate is displayed by MONITOR IO

-Evidence of clearly active work being swapped.

DR2 Test Hard Fault Rate

The hard fault rate is calculated as the observed rate of Page File Read I/O's per CPU second less hard faults used for image activation. Rule of thumb: 4 hard faults per image activation. Accounting data is the only accurate source of image activation data; Monitor does not report it.

High rates are those in excess of:
2 per second (750/780/8200)
2.5 per second (785)
7 per second (8550/86x0 with local system disk)
5 per second (8550/86x0 with system disk on CI if total I/O load substantial)

Low rates are those less than:
.8 per second (750/780/8200)
1 per second (785)
2.5 per second (8550/8600/8650)

DR3 Test Soft Fault Rate

The soft fault rate is calculated as the observed rate of page faults per CPU second less faults used for image activation. Rule of thumb: 70 page faults per image activation.

High rates are those in excess of:
60 per second (750)
100 per second (780)
125 per second (785)
350 per second (8600)
500 per second (8550/8650)

* remember faults include image activation ∴ you should lower the observed value to compensate for this.

Low rates are those less than:
25 per second
40 per second (780)
50 per second (785)
125 per second (8600)
150 per second (8550/8650)

DR4 Individual User/Application Excessive Faulting

Any application or user which displays high hard or soft fault rates relative to the observed average for the system (following the above formulas) or which uses exceptionally large working sets relative to the nature of the application (e.g., an editor using 800 pages) or which employs extensive use of Mapped Section I/O represents a problem which should be dealt with by addressing the software or user practices rather than system tuning. These programs are no less "broken" than a program which uses two CPU minutes to copy a 100 block file. Especially suspicious is any program (regardless of its fault rate) running in a working set in excess of 1750 pages which is not doing so to make effective use of memory as a disk cache.

If an application or user is displaying high soft fault rates, it is either due to high image activation frequency (which is a performance problem which should be addressed, although is not a memory management problem, per se) or it is having its working set limited by WSQUOTA.

DR5 Can a Problem Program/User be Fixed?

- Is the software accessible? Can the user's practices be changed?

If a program is faulting heavily due to an insufficient WSQUOTA, is it possible to raise WSQUOTA? The answer may be no if such would lower overall system performance for other users, and the relative importance of the problem application is low enough that it should be forced to suffer poor performance, or run at times of lower system activity.

DR6 Excessive Batch Work in System?

If there are many batch jobs in the system, (and, they are not swapped out because they are getting at least a little CPU time occasionally,) they may be occupying lots of memory. Only that number of batch jobs sufficient to "soak up" otherwise unused CPU time should be allowed to run simultaneously.

Action Steps

AS1 Decrease Secondary Caches

Secondary caches are decreased by reducing FREELIM (to reduce the size of the Free List) and MPW_HILIMIT (to reduce the size of the Modified Page List). The relationship between the size of the two should be approximately equal to the ratio between observed Free List and Modified Page List faulting as reported by MONITOR.

If Page Write I/O exceeds .3/second (750) to 1./second (8550/8650), this may indicate that MPW_HILIMIT is set too low, but more often indicates either 1) an application is randomly accessing extensive regions of memory and should be repaired, or 2) the page file has become excessively fragmented because it is too small, or is allocated in a non-contiguous manner.

When changing FREELIM and MPW_HILIMIT, observe the relationships rules vis a vis GROWLIM, BORROWLIM, FREEGOAL, MPW_LOLIMIT, MPW_THRESH, and MPW_WAITLIMIT.

AS2 Decrease Working Set Sizes

Working set sizes are decreased by raising the values for PFRATH and PFRATL. Do not lower WSQUOTA or WSEXTENT.

Notice that these parameters are dynamic -- they may be changed without needing to re-boot VMS. In fact, VMS is extremely sensitive to these values -- the effect of changing the them takes hold within a few seconds after doing so.

AS3 Fix Broken Software

Steps which may be taken include:

-Raise WSQUOTA if working sets tend to run at the current WSQUOTA and hard fault and swapping rates could be allowed to get higher.

-Change software to have top down control flows and better modularity.

-Reorganize the order of routines within images and reduce the number of separate routines used during paticular processing phases.

-Redimension arrays.

-Consolidate data referenced by individual modules.

-Use explicit deterministic data overlaying techniques for very large data structures. Eliminate use of Mapped Section I/O.

-Reduce size and number of I/O buffers if I/O performance not impacted.

-Buy a competitive product that performs better.

AS4 Remove Excessive Batch Jobs

Reduce job limits on queues or stop them during prime shifts. An automatic procedure may be introduced to adjust these limits at the start and end of prime shifts.

AS5 Excessive Memory Tuning

If there is more memory than is needed on a VAX for the workload being run, there are two choices available:

-Sell the extra memory. This increases system reliability and increases the productivity relative to system cost. However, this may not be a feasible alternative due to organizational politics or red tape.

-A marginal performance improvement might be obtained by doing the following:

-Doubling AWSTIME

-Raising WSDEFAULT to 300

-Setting WSINC to 200, WSDEC to 47 Don't change WSDEC

-Increasing ~~ACP cache sizes~~ and SYSMWCNT (but undo this change if cache hit rates not increased.)

-Raise ACP_WINDOW to 10, increase BYTLIM for all users by 10%.

-Change RMS_DFMBC to 32 (if not already increased.)

-If after all of the above changes the Free List size remains consistantly well in excess of FREELIM, let working sets increase (AS7).

AS6 Balance Fault Rates

The ratio of hard faults to total faults should be roughly as follows:

1:40 -	batch oriented 750/780/785/8200
1:50 -	normal 750/780/785/8200
1:60 -	response critical 750/780/785/8200
1:60 -	85x0/86x0
1:80 -	85x0/86x0 with high I/O rates and page and swap files on the CI, or a response critical environment

Balance these rates to the above ratios by doing the following (changes in parameter values should be about 5%):

If hard faults too high, lower working set sizes (AS2) and increase secondary caches (AS8).

If hard faults too low, increase working set sizes (AS7), and decrease secondary caches (AS1).

AS7 Increase Working Set Sizes

Working set sizes are increased by lowering the values for PFRATH and PFRATL. Do not raise WSQUOTA or WSEXTENT.

AS8 Increase Secondary Cache Size

Secondary caches are increased by raising FREELIM (to raise the size of the Free List) and MPW_HILIMIT (to raise the size of the Modified Page List.) The relationship between the size of the two should be approximately equal to the ratio between observed Free List and Modified Page List faults as displayed by MONITOR PAGE.

However, MPW_HILIMT should never be set so large that, under normal heaviest sustained loading, the Modified Page List isn't reaching that size, and pages are written to the Page File, at least every few seconds.

One special check must be made after increasing the Modified List Size if there is any swapping activity in the system. Observe the MONITOR IO or PAGE display and note the frequency of Modified List dumps (ie., intervals where page file writes are non-zero.) If the Modified List is reduced to zero by the dump (indicated by the Modified List size dropping well below MPW_LOLIMIT) more often than once every few minutes and the overall dump frequency is low (1.5 times a minute or less), restore the Modified List to its previous smaller size and leave only the Free List larger.

When changing FREELIM and MPW_HILIMIT, observe the relationships rules vis a vis GROWLIM, BORROWLIM, FREEGOAL, MPW_LOLIMIT, MPW_THRESH, and MPW_WAITLIMIT.

Additional Tuning Steps

Observe the system fault rate -- if greater than 2/second (7x0/8200) or 4/second (85x0/86x0), raise SYSMWCNT. If less than .5/second (7x0/8200) or 1/second (85x0/86x0), lower SYSMWCNT, unless there is excessive memory in the system.

Set SRPCOUNT, IRPCOUNT, LRPCOUNT and NPAGEDYN to values normally observed using SHOW MEMORY/FULL during heavy processing. Keep SRPCOUNTV, IRPCOUNTV, LRPCOUNTV and NPAGEVIR at least double these values.

eats up lots of mem. - so be carefull.

controlls largest image that you can run.

Keep BALSETCNT, VIRTUALPAGECNT and WSMAX at reasonable values. See the SYSGEN Utility Guide for an analysis of the amount of memory these parameters cause to be permanently allocated to the system.

* if too low, could crash system.

* if too high, consumes some memory.

executable file open & don't have to close.

50 - 8550

Any program which is run frequently (>30 times per hour) should be installed "known". If the rate is over 120/hour it should be installed "header resident". If many users tend to use the program simultaneously, and it is activated frequently, it should be installed shared, if possible.

200 - 8550

don't have to load header into mem. because it will already be there

Images with low activation rates, but which tend to be active for long periods of time, should only be installed shared if 3 or more users tend to use the program simultaneously. The primary benefit of sharing is the possible reduction in hard faulting. Significant memory savings are seldom acheived.

Consider changing QUANTUM. Lowering QUANTUM may improve response time (at the cost of an increase in CPU overhead). Raising it may slightly reduce overhead, but make AWS less responsive. QUANTUM should be lower than 200 milliseconds for 85x0/86x0's.

* 750's - 30
8550's - 10

Quantum

This memory management procedure should, and generally will, be applied to the times of heaviest interactive use of the system, because these loads impose the severest memory demand on the system. As such, the system will be mis-tuned in regard to others times of the day, where batch loads might impose equally heavy CPU usage demands. A partial re-tuning of the system for evening and weekend may be accomplished by having a batch job execute at appropriate times to run SYSGEN to reset some dynamic parameters. Assuming there is excess memory during these off-prime times (the Free Lists stays well in excess of FREELIM) the tuning changes would involve increasing working set sizes by lowering PFRATH and PFRATL. Further, so long as response times for any off-prime interactive load are not adversly affected, users may wish to double or triple QUANTUM and AWSTIME. If this is done WSINC and WSDEC should also be raised.