VS036

Monday, May 9, 1994

1:30 p.m. – 2:30 p.m.

Intermediate/Technical

OpenVMS System Performance Management I: Memory Management

Harry E. Flowers
The University of Memphis
112 Administration Bldg.
Memphis, TN 38152
flowers@narnia.memphis.edu

Disclaimer: The author and The University of Memphis accept no responsibility for the use or reliabilty of information presented in this session.

Memory Management

- Installing images for better performance
- Working set and page list concepts, quotas, and SYSGEN parameters
- Page and swap file tuning
- AUTOGEN is your friend (with a little help)

Installing Images

- Advantages
  - Faster image activation

    Without qualifiers Even if you don't use any of the qualifiers on an install, the image becomes a "known file", and you skip the disk accesses needed for the directory lookup of the file.

    Open File is kept open by the system, and channels may easily be assigned to it without additional disk access

    Header Resident Images have internal headers that contain information about the how virtual memory is mapped in program sections. If this information is kept in memory, it further speeds activation.
  - Faster image execution and conservation of physical memory

    Shared If someone else has faulted the executable page in, you perform a global fault rather than a hard or possibly soft fault. Also, the same page of physical memory is mapped to all working sets that reference it, conserving physical memory.
- Identifying targets for installation
  - Use `SHOW DEVICE/FILES/NOSYS` to find images
  - Some will have to be created, others replaced

Working set and page list concepts

Quantum End A process is given a certain amount, or quantum, of CPU time before it is removed from execution and evaluated for certain conditions. A CPU-bound process will go through quantum end often, while one doing terminal I/O will take longer to exhaust its quantum. One of the quantum end activities is working set list size adjustment.

Working Set (WS) A working set is a set of pages of physical memory mapped to your process. In the case of global pages, many processes have the same pages in each of their working sets.

Working Set List (WSL) Each process has a working set list which contains the list of pages currently in their working set, as well as any empty slots available for new pages. The list size is fixed while a process is executing. Page faults will fill the list until there are no more slots, when a page must be removed from the list to bring in a new one.

Working set and page list concepts, cont'd

Page faults In order for a memory location to be referenced, the page represented by it must reside in the working set. If it's not there, it causes a page fault. The page fault in turn causes the page to be brought into the working set. The page may be an existing global page (a global fault), or come from the modified page list, or the free page list. The information for the page may come from one of these (modified and free are soft faults), or disk (a "hard fault").

Page fault rate The page fault rate is how fast pages are being faulted for a process. It is used in determining when to adjust the working set list size at quantum end.

Free page list The main pool for free memory pages. Pages which have not been modified will be paged out of working sets and cached here. Modified pages will also be cached here after their contents have been preserved.

Working set and page list concepts, cont'd

Modified page list The main cache for memory pages which have been modified by a process. When this list is flushed, the information in the pages is first written out to their backing store, usually a system page file, before the page is returned to the free page list.

WSDEFAULT Default working set list size (SYSUAF)

WSQUOTA Maximum working set list guaranteed expansion (SYSUAF)

WSEXTENT Maximum working set list borrowed expansion (SYSUAF)

WSMAX Maximum working set list size for any process (SYSGEN)

PGFLQUOTA Maximum number of pages which can have system page files as their backing store (SYSUAF)

A few seconds in the life of a process

- The process activates an image. A working set list size of WSDEFAULT is set to hold the working set references. As the process page faults successive image pages, the working set list fills. Once the list is full, pages must be removed from the list before others can be added. If the page isn't a global one, it gets placed on the appropriate list.

- After the process reaches quantum end, the page fault rate is evaluated. If it is greater than the SYSGEN parameter "PFRATH" (Page Fault RATe High), then the working set list is enlarged by WSINC pages, with "no questions asked" up to WSQUOTA. If growth beyond WSQUOTA is indicated, then the process' working set list size may grow to WSEXTENT provided there are at least BORROWLIM pages on the free page list.

A few seconds in the life of a process, cont'd

- As the image is again executing, pages may be faulted into the working set up to the current working set list size. If the process is above WSQUOTA, an additional check is made to ensure there are a minimum of GROWLIM pages on the free page list.
- When the image exits, non-shared pages are returned to the free page list, and any modified pages in the modified page list or page file are freed. When another image is activated, the cycle starts again.

From this we can see that there are two factors in working set growth:

1. While the process is executing, pages are faulted into the working set up to the current list size.
2. When the process is evaluated at quantum end, the working set list size may change.

Free page list SYSGEN parameters

FREELIM Minimum number of pages desired on free page list

FREEGOAL Target number of pages desired on free page list; the newer memory reclamation procedures key off this value

BORROWLIM Minimum number of pages on the free list to allow WSL expansion

GROWLIM Minimum number of pages on the free list to allow WS growth

MMG_CTLFLAGS Controls proactive memory reclamation procedures

- **bit 0** controls trimming of periodically executing processes, and is triggered when the free list drops below two times FREEGOAL
- **bit 1** controls outswapping of idle processes, and is triggered when the free list drops below FREEGOAL

Modified page writer SYSGEN parameters

MPW_LOLIMIT Number of pages on the modified page list (MPL) where MPW stops flushing

MPW_THRESH Threshold where MPW is not used as a memory recovery mechanism

MPW_HILIMIT Number of pages on the MPL where MPW is called to begin writing pages to their backing store and move them to the free list

MPW_WRTCLUSTER How many pages the MPW writes at a time

MPW_WAITLIMIT MPL size where processes wishing to place pages on the list are placed in RWMPB (Resource Wait; Modified Page writer Busy) state; it should be an integral number of MPW_WRTCLUSTER pages over MPW_HILIMIT

MPW_LOWAITLIMIT MPL size where processes are removed from RWMPB state; one cluster below MPW_HILIMIT is typical.

Page and swap files: what, where, and why

Page files are the primary backing store for modified pages. Since they fragment internally very rapidly, the rule of thumb for use is 50% at peak system load. This may seem like a lot of overhead, but every user on your system will wonder what's going on as it creeps towards becoming full. You'll even get warning messages on the system console as the problem progresses.

Swap files are used to move the memory contents for entire processes to disk. Since this is done more infrequently and in much bigger chunks, the swap file doesn't fragment as much internally. The rule of thumb for use is 75% at peak system load. Since VMS V5.0, many systems don't even use swap-only files. The term "swap-only" is used because page files can also be used for swapping, but swap files can't be used for paging.

Page and swap files: what, where, and why, cont'd

Page and swap files (if any) should be moved off the system disk if possible. The demand on the system disk is already high in almost any environment, and you don't want to contend for I/O there. A disk just for paging, or having little other I/O is preferable. Remember, it's memory transfers that are waiting on this, and low average usage is okay to provide the I/O bandwidth when the modified page writer needs it. Swap files tend even more so to have low average I/O, but sporadic high demand for I/O bandwidth. If you regularly have processes outswapped (their process state will end in "O"), it's a good idea to have swap files. If not, you can better use the space for paging, and let rare occurrences of swapping use the page file (since you've now got all that extra space you're not hogging with a swap file).

Some useful information on page and swap files may be obtained with:

```
$ SHOW MEMORY/FILES/FULL
         System Memory Resources on 16-MAR-1994 12:21:15.03

DISK$CASPIAN:[SYSFILES]SWAPFILE1.SYS
    Free Blocks                 15000   Reservable Blocks         15000
    Total Size (blocks)         15000   Paging File Number            1
    Swap Usage (processes)          0   Paging Usage (processes)      0
  This file is used exclusively for swapping.

DISK$CASPIAN:[SYSFILES]PAGEFILE1.SYS
    Free Blocks                 82060   Reservable Blocks         20421
    Total Size (blocks)        100000   Paging File Number            3
    Swap Usage (processes)          0   Paging Usage (processes)     29
  This file can be used for either paging or swapping.
```

HELP! I've got negative "Reservable Blocks"!

Many of you are used to seeing negative values for "Reservable Blocks". As long as you've followed the rules-of-thumb outlined before, you're probably okay. The "Reservable Blocks" value is just used for balancing usage between multiple page files. The special SYSGEN parameter "RSRVPAGCNT" (ReSeRVe PAGe CouNT) is used as the value to deduct from the reservable blocks each time a process is assigned to that file. Ignore it. The number of free blocks is what you need to be concerned with.

AUTOGEN is your friend (with some help)

AUTOGEN is a mixed blessing. Used carefully, it's a great tuning tool. Used carelessly, or even casually, it can render your system unbootable.

AUTOGEN takes from one to three parameters:

```
@SYS$UPDATE:AUTOGEN startphase endphase options
```

If only one parameter is specified, it's treated as both the starting phase and ending phase. Options may be NOFEEDBACK when starting from SAVPARAMS to disable feedback (needed if the system has been up less than 24 hours), and the phases are as follows:

SAVPARAMS Saves feedback information from the running system. Since some of the information it gleans is not summary or peak data, you should use this phase during regular peak load conditions.

AUTOGEN is your friend (with some help)

GETDATA Collects data used in calculations. This includes the feedback information as well as what you've placed in MODPARAMS.DAT, along with hardware information and Digital defaults.

GENPARAMS Does the calculations from the data and saves the results in SETPARAMS.DAT. A good way to check what's changed between AUTOGEN runs is to compare the last two versions of SETPARAMS.DAT. The report file, AGEN$PARAMS.REPORT should also be checked. It contains usage information, but may not show all parameter changes. If you've run AUTOGEN many times since you set the parameters or rebooted, comparing against the active parameters is a good idea.

AUTOGEN is your friend (with some help), cont'd

TESTFILES An ending phase, allows you to check what changes AUTOGEN would make to the system page, swap, and dump files if you executed the next phase.

GENFILES Generates new page, swap, and dump files from its calculations if it determines the size is incorrect.

SETPARAMS Uses the information from GENPARAMS to actually set the new parameters. Since they are written as current, and not active parameters, a reboot is required for the changes to take effect.

SHUTDOWN Specifies that the system should be shut down after the AUTOGEN.

REBOOT Specifies that the system should be rebooted after the AUTOGEN.

AUTOGEN is your friend (with some help), cont'd

If you keep these things in mind, you'll find AUTOGEN your best performance tuning tool.

- It can calculate values for SYSGEN parameters that are not sufficient to boot or run the system. You usually find out the next time you attempt to reboot, and must become reacquainted with conversational booting to fix the problem.
- It can calculate values for SYSGEN parameters that are outside the limits for that parameter. You usually find out during the SETPARAMS phase, but not what the offending parameter was! You have to feed the SETPARAMS.DAT file through SYSMAN yourself to find out which one is bad. If you're using an earlier version of VMS, AUTOGEN may use SYSGEN instead of SYSMAN to perform the changes.

AUTOGEN is your friend (with some help), cont'd

- Since the change to using SYSMAN, a SYSMAN initialization file that does `SET ENVIRONMENT/CLUSTER` is disastrous in a cluster environment. Your cluster won't run well using one SCSNODE name and SCSSYSTEMID for every node.

- Its FEEDBACK mechanism is good for tuning many parameters. However, often you know better what the demands will be for certain resources. Use of "MIN_parameter" is recommended in MODPARAMS.DAT to keep these values at proper settings. Also, disabling its page, swap, and dump file sizing by setting the parameters equal to zero in MODPARAMS can save you a few surprises.

AUTOGEN is your friend (with some help), cont'd

AUTOGEN is best run in three steps:

1. `@SYS$UPDATE:AUTOGEN SAVPARAMS` run during peak load. This is quick, and won't impact your system performance.

2. `@SYS$UPDATE:AUTOGEN GETDATA GENPARAMS` run during off hours. This is the time-consuming part of AUTOGEN, and you may as well put it off peak load times. You should check the results before the next step.

3. `@SYS$UPDATE:AUTOGEN SETPARAMS` run after you've verified that you wish to make the changes, and can reboot sometime later to use the changes.

A batch job that performs steps 1 and 2 every working day is not difficult to set up, and doesn't hurt anything. Step 3 should not be automated, as you really need to check the results from step 2 first.

Questions?

Questions?

Questions?