

Best of the HP Customer Support Center

Mark 'Jilly' Jilson
Service/Support Consultant
Customer Solutions & Support Center
OpenVMS Internals, Drivers & Performance

Overview

One of the most frequent problem statements that the CSC receives is one asking help in finding out why some system behavior changed. These might be questions like "why is a job taking longer to run?" or "why is a system running slower?" or "why is that backup job using more tape?" etc. One of the basic steps in responding to and analyzing such problems is to compare before and after profiles of the involved areas, but the CSC very often finds that there are no profiles to compare. This article describes some of these profiles and methods for collecting them. These profiles should be stored in at least two places outside of the system/cluster they cover and I advocate at least one paper copy so that when you really need them they can be accessed. With before and after profiles, it becomes a lot easier to pinpoint what area has changed and often directly answers the "What Changed?" question.

What Changed?

What devices are there on the system? What are their hardware revision levels and what are their firmware versions? How do the system devices look from the utilities that can see them?

Configuration Profile

Start with the device listing that the console can provide. With the OpenVMS system shut down (in a cluster, preferably have all nodes shut down), capture the output from the available console SHOW commands. Available items will vary from system to system but the following commands should work for most systems. If any of these commands don't work, then check the system's hardware User's Guide or consult with your hardware services representative or the CSC. To capture this information without transcribing it requires the use of a hardcopy console or a console management setup.

For Alpha systems use:

```
>>> SHOW CONFIG  
>>> SHOW DEVICE  
>>> SHOW MEMORY  
>>> WWIDMGR -SHOW WWID -FULL
```

For VAX systems use:

```
>>> SHOW CONFIG  
>>> SHOW DEVICE  
>>> SHOW MEMORY  
>>> SHOW SCSI  
>>> SHOW DSSI  
>>> SHOW VERSION
```

With OpenVMS booted on all cluster nodes, use the following command to capture configuration

information:

```
$ SHOW DEVICE/FULL/OUTPUT={file.ext}
```

Editing this output to remove template devices and transient devices, like LTA or TNA, is recommended. But leave in any devices that are specifically created at startup like printer ports.

On OpenVMS Alpha, use the following commands to capture configuration information:

```
$ ANALYZE/SYSTEM  
SET OUTPUT {file.ext}  
CLUE CONFIG  
CLUE SCSI/SUMMARY  
EXIT
```

With the help of your hardware services representative and the CSC, annotate these displays with what the devices are, where they live in the system enclosure, card cage, etc., what external port they connect to on the system enclosure, what hardware settings they have (SCSI ID, CI node number, etc.), what hardware revision they are at (if any) and what firmware they are running (if any). If you make use of the Golden Eggs diagrams, available at <http://h18000.www1.hp.com/info/golden-eggs/>, then you can annotate these with references to this collected output. If any of the devices have logical names or are used by print queues or execution queues, or you have specific names for a device, note this information also.

For storage subsystems, collect the same type of information. Consult the User's Guide for the storage subsystem to determine what information is available and how to acquire it.

Hardware Setup Profile

This information is stored in the system console that selects boot options, device operating options, reboot options, etc.

For Alpha systems, use the following command at the console to collect this information:

```
>>> SHOW *
```

For VAX systems, use the following commands at the console to collect this information:

```
>>> SHOW BFLG  
>>> SHOW BOOT  
>>> SHOW ETHERNET  
>>> SHOW HALT  
>>> SHOW SCSI_ID
```

Annotate these displays with descriptions for what disk device and root the system is booting from, operating options for network cards, operating options for SCSI cards, etc.

System Setup Configuration

This is the information that is used at boot time to configure the devices on the system and to set operating parameters.

For OpenVMS Alpha, collect output from the following commands:

```
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:ALPHAVMSSYS.PAR
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:ALPHAVMSSYS.OLD
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:AUTOGEN.PAR
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:SETPARAMS.DAT
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:PARAMS.DAT
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:MODPARAMS.DAT
```

For OpenVMS VAX, collect output from the following commands:

```
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:VAXVMSSYS.PAR
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:VAXVMSSYS.OLD
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:AUTOGEN.PAR
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:SETPARAMS.DAT
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:PARAMS.DAT
$ DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$SYSTEM:MODPARAMS.DAT
```

The dates and ids of these files will show when AUTOGEN was last executed, when parameters might have been changed with SYSBOOT or SYSGEN without an AUTOGEN, or if a parameter change was contemplated but not actually implemented.

Execute the following commands to capture system parameter information in a file:

```
$ MCR SYSGEN
USE CURRENT
SET/OUTPUT {file.ext}
SHOW/ALL
SHOW/SPECIAL
EXIT
```

The output file will now contain the settings of all the system parameters in the on-disk system parameter file.

Save a copy of the system files SYS\$SYSTEM:MODPARAMS.DAT, SYS\$SYSTEM:SYCONFIG.COM, and SYS\$SYSTEM:SYS\$DEVICES.DAT and add any other device creation commands that are executed at system startup time like LTA or TNA devices used for printing. With this information it will be very easy to determine if anything in the system setup might have been altered.

System Boot Profile

Save the console output from a normal system startup. If startup logs via STARTUP_P2 are implemented, then also save a copy of a log from a normal system boot. If possible, boot an OpenVMS Alpha system with full verbose boot options at the console by setting the boot flags to 30000 and save the console output from this. If possible boot the system with STARTUP_P2 set to TRUE and save the console output. Annotate these outputs with what portions of startup are executing.

Save the output from the following command:

```
DIRECTORY/DATE=(CREATED,MODIFIED)/FILE_ID SYS$STARTUP:*.COM,
SYS$SYSTEM:*.COM
```

Additionally gather this file information from any other system startup procedure that is executed.

After the system has booted normally, log in and save the output from the following commands:

```
$ ANALYZE/SYSTEM
SET OUTPUT {file.ext}
SHOW SUMMARY/IMAGE
EXIT
```

Annotate this output with descriptions of any application processes.

Performance Profile

A system performance profile, at minimum, should represent significant time periods in the system operation cycle and should cover a maximum 30 minutes of elapsed time. Save both the raw performance data and the performance reports for these time periods. OpenVMS provides the MONITOR utility and the ECP Data Collector and Performance Analyzer is freely available at <http://h71000.www7.hp.com/openvms/products/ecp/index.html>; either or both of these can be used.

OpenVMS provides examples of using MONITOR to gather performance data and generate a performance report in the following files: SYS\$EXAMPLES, SUBMON.COM, MONITOR.COM & MONSUM.COM . These files need very little editing to be usable. The following are the minimum MONITOR commands needed to gather a performance profile that covers 30 minutes of elapsed time; these commands can be submitted via batch.

```
$ MONITOR
ALL_CLASSES/RECORD={recfil.ext}/NODISPLAY/INTERVAL=120/END="+0:30:0.0"
$ MONITOR ALL_CLASSES,DISK/ITEM=ALL,SCS/ITEM=ALL/NODISPLAY -
  /INPUT={recfil.ext}/SUMMARY={file.ext}
```

Both the recording file and the summary file are saved.

If MONITOR data is already being saved, then here is an example of a command for creating a report if the period from 9am to 9:30am is a peak operational time.

```
$ MONITOR
ALL_CLASSES,DISK/ITEM=ALL,SCS/ITEM=ALL/NODISPLAY/SUMMARY={file.ext} -
  /INPUT={recording file that covers the date}/BEGIN=12-MAY-2003:09:00:00.00 -
  /END=12-MAY-2003:09:30:00.00
```

Both the recording file and the summary file are saved.

If ECP is collecting data, then the following command is an example of creating a report if the peak time period of interest was 13:30 to 14:00.

```
$ PLAN ANALYZE/CPC_VMS_FILE=ECP$PERF_DATA:NODE_2003MAY12_1.CPC -
  /BEGIN=12-MAY-2003:13:30:00.00/END=12-MAY-2003:14:00:00.00 -
  /ANALYZE_REPORT_FILE={file.ext}
```

Both the report and the .CPC file are saved.

In addition to overall system performance reports that cover significant operation cycles, collect accounting information from significant jobs that execute on the system. These are jobs that you

expect to complete in a certain elapsed time or are usually finished before a certain time. Also the accounting logs from backup jobs should be recorded. If any of these jobs have multiple parts to them, then adding SHOW PROCESS/ACCOUNTING commands to the job's command procedure(s) are beneficial. Then, when needed, different portions of a job can be compared before and after. If you have any application benchmark jobs or can create some simple benchmark jobs, save the accounting information from these. Finally, if you have application-provided performance data (for example, number of queries per minute or number of orders per hour, etc.) save this also.

Managing and Using the Profiles

Profiles should be refreshed periodically and after any changes are made to the system or the system's application load. Integrating these profiles into your system management structure will ensure that when the time comes to answer the question "What Changed?", you will be well prepared to quickly zero in on the area responsible for the change.