Technical white paper

Tuning and troubleshooting HP OpenVMS OPCOM

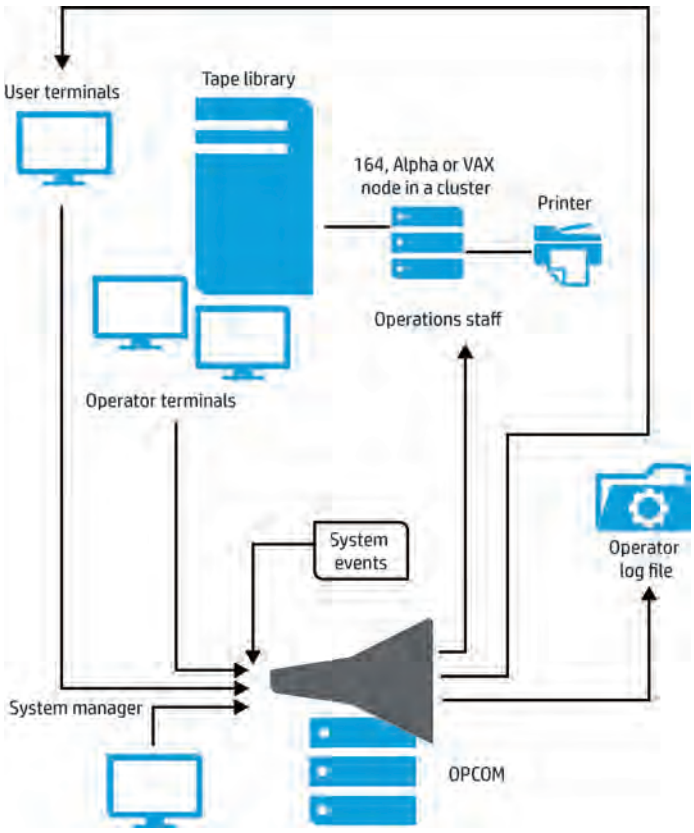
Table of contents

Overview

This white paper outlines the different tuning mechanism available for HP OpenVMS operator communication manager (OPCOM). It also articulates the common issues reported by different customers and how to overcome these issues.

Introduction to OPCOM

The OPCOM is a tool for communicating with users and operators on OpenVMS systems. OPCOM provides a mechanism to communicate events to users on the same system and to users on different members of an OpenVMS cluster. Events can be related to device configurations, user requests, operator replies, or system events. OPCOM is similar to the syslog facility on UNIX® and Linux systems.



Operator events are classified to different classes as shown below:

CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, CLUSTER, SECURITY, LICENSE, OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10, OPER11, OPER12

Operator classes OPER1-OPER12 can be redefined using logicals OPER1–OPER12.

Logical names should not collide with any devices, usernames, or software packages that are installed on the OpenVMS system:

$ DEFINE /SYSTEM TEST_OPER1 OPER1

Tune OPCOM during startup

OPCOM provides a set of logicals to control message broadcasts. These logicals can be defined in SYS$MANAGER:SYLOGICALS.COM.

Following are the list of logical using which message broadcast can be controlled.

1. OPC$OPA0_ENABLE – Enables or disables OPA0: device as operator terminal. If the logical is not defined, by default the OPA0: is not an operator terminal. Select the kind of broadcast messages to OPA0: device by using the commands.

   $ SET BROADCAST

   For example:

   $ SET BROADCAST=MAIL

   The above command would broadcast messages to terminals only related to mail notifications. Note that this command should be used only on terminals other than OPA0: device.

2. OPC$OPA0_CLASSES—Classes for which OPA0 device is enabled as an operator terminal to receive BROADCAST messages.

   Possible classes for this logical are cards, central, cluster, devices, disks, license, network, oper1 to oper12, printer, security, and tapes.

   By default enabled for all classes.

3. OPC$LOGFILE_NAME—Decides the location of OPERATOR.LOG. By default resides at SYS$MANAGER:

4. OPC$LOGFILE_ENABLE—Enable or disable the creation of OPERATOR.LOG.

5. OPC$ALLOW_INBOUND—Controls the inbound traffic from a node. By default set to TRUE. The node will not receive most OPCOM messages from other nodes in cluster when set to FALSE.

6. OPC$ALLOW_OUTBOUND—Controls the outbound traffic to a node. By default set to TRUE. The node will not send most OPCOM messages to other nodes in cluster when set to FALSE.

   Detailed information about these logicals is available in the file SYS$STARTUP:SYLOGICALS.TEMPLATE.

Apart from defining these logicals, message broadcast can be controlled interactively by REPLY/LOG, REPLY/ENABLE, and REPLY/DISABLE commands.

- REPLY/LOG: Enables terminal on which this command is issued as an operator. This command also closes the existing OPERATOR.LOG and creates a new OPERATOR.LOG file without rebooting the system.
- REPLY/ENABLE[=(keyword[,...])]: Enables terminals as an operator, to receive the messages belonging to classes specified as keyword. If no keyword is mentioned all the events are enabled. This cannot be entered from batch job.
- REPLY/DISABLE[=(keyword[,...])]: Disables terminals as an operator for messages belonging to classes specified in keyword. If no keyword is mentioned disables for all the events. This cannot be entered from batch job. It is recommended to never disable the OPA0: as an operator terminal.

The OPER privilege is needed to enable or disable the operator terminal. The SHARE privilege is needed if another process is logged into the designated operator's terminal. The SECURITY privilege is required to enable/disable security events.

Tune OPCOM flow by DECwindows

OPCOM messages can be redirected to a serial console or a graphics console. To redirect the OPCOM messages to DECwindows define global symbol DECW$CONSOLE_SELECTION to one of the values WINDOW, DISABLE, or ENABLE in the customized startup file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM. By default these symbols are defined as DISABLE

For example DECW$CONSOLE_SELECTION == ENABLE

If the value to the symbol defined is:

1. WINDOW: Displays console messages in the Console Window application. The console is a graphic console. The Console Window is displayed in the lower right corner of the login screen by default and continues to be displayed after the user logs in to the system. The Console Window looks similar to the Message Window without a menu bar. To control the initial position of the Console Window modify the symbol DECW$CONSOLE_GEOMETRY and the classes of OPCOM output that are enabled. DECW$CONSOLE_GEOMETRY symbol can be defined in the file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM.

   The default value is "-0-0", which specifies the location of the window in the lower right corner of the screen.

   To position the window at the lower left corner of the screen, add the following line to command file:

   DECW$CONSOLE_GEOMETRY == "+0-0"

2. DISABLE: Disables broadcasts to the OPA0: device. Console messages are not displayed.
3. ENABLE: Displays console messages in the console window. The console is a serial console. The console window is a six-line display area at the top of the workstation screen. It is recommended to not use this option, since displaying console messages by default in the console window can corrupt the contents of the workstation display.

Tune message flow to OPERATOR.LOG file

Default log file which captures operator messages is OPERATOR.LOG available at location SYS$MANAGER:. The OPC$LOGFILE_NAME name is used to specify an alternative location. This log file is created if OPC$LOGFILE_ENABLE logical is defined as “TRUE”, as explained above. The system creates a new version of OPERATOR.LOG each time the system is rebooted. In clustered environment one operator log file exists on each node. This file should not be shared across the nodes in a cluster.

Operator messages which gets logged into the log file can be controlled depending on the operator class to which the message belongs to using the below commands.

- Use REPLY/LOG/ENABLE= (keyword) and REPLY/LOG/DISABLE= (keyword) commands to specify which operator classes to include/exclude in the log file.
- Use the /LOG qualifier alone with REPLY to include all classes in the log file.

If a log file is already open, the list of classes enabled is preserved and enabled on the newly created log file. If a log file is not open, the value of the logical OPC$LOGFILE_CLASSES is used. If that logical does not exist, all classes are enabled on the new log file.

Messages from all operator classes logged to OPA0: do not flow to the OPERATOR.LOG. Messages that do not flow are:

- System SECURITY alarms and audits enabled using SET AUDIT command. These SECURITY messages are either logged to operator terminal and/or to SYS$COMMON:[SYSMGR]SECURITY.AUDIT$JOURNAL.
- Messages generated by the LANACP LAN Server process when a device status changes. These messages are displayed on the operator terminal and included in the log file written by LANACP, SYS$MANAGER:LAN$ACP.LOG

Note

- Format of system messages can be changed with the DCL command SET MESSAGE; these messages might not appear in the log file.
- If OPC$LOGFILE_CLASSES includes an invalid class then all classes are enabled.
- OPERATOR.LOG messages are buffered within the OPCOM process. Hence there could be a delay up to five minutes when the OPCOM messages are actually logged into the operator log file.

Tune security class events

Security-relevant activities can be monitored by recording the events as they occur on the system and then analyzing the audit log. Security-related events are divided into a number of categories called **event classes**.

To enable auditing for different event classes, use the following command:

SET AUDIT /ENABLE=event-class[,...] {/ALARM | /AUDIT}

To check the events for which Auditing is enabled, use the command:

SHOW AUDIT/ALL

Below is a sample output:

List of audit journals:

Journal name: SECURITY

Journal owner: (system audit journal)

Destination: SYSD$:[AUDIT]SECURITY.AUDIT$JOURNAL

Monitoring: enabled

Warning thresholds, Block count: 100 Duration: 2 00:00:00.0

Action thresholds, Block count: 25 Duration: 0 00:30:00.0

Security auditing server characteristics:

Database version: 4.4

Backlog (total): 100, 200, 300

Backlog (process): 5, 2

Server processing intervals:

Archive flush: 0 00:01:00.00

Journal flush: 0 00:05:00.00

Resource scan: 0 00:05:00.00

Final resource action: purge oldest audit events

Security archiving information:

Archiving events: none

Archive destination:

System security alarms currently enabled for:

Breakin: server

System security audits currently enabled for:

Breakin: dialup, local, remote, network, detached

Log failure: batch, dialup, local, remote, network, subprocess, detached, server

Events that are listed under system security alarm section in the above output would be displayed only on operator terminals and won't be logged to operator log.

Events that are listed under system security audit section in the above output are logged to SYS$COMMON:[SYSMGR]SECURITY.AUDIT$JOURNAL only.

Common problems and solutions

Problem 1

OPCOM process crashes with INSVIRMEM. OPCOM could not empty the pending messages (910548 messages) to the console device. When OPCOM tried to allocate space for another message an INSVIRMEM error was generated, as it had run out of P0 process space. OPCOM is designed to handle unexpected conditions like this by producing a process dump and restarting itself (dropping the pending terminal messages stored in its process address space, though they should be in the operator log file).

Possible scenarios

1. The incoming messages flooded OPCOM so quickly that it could not keep up with writing them to the slower OPA0 console device.
2. The console (OPA0: device) is disabled to receive the operator message and thus OPCOM P0 memory is exhausted because of pending messages to console. Below are the ways by which console can be disabled.
   - XOFF or Hold Screen button for DECW terminals
   - Control-s, or similar operation.

Solution

If a huge incoming message flow was root caused to audit settings, audit settings can be viewed by issuing the command SHOW AUDIT/ALL.

The audit settings may be adjusted to reduce the message flow to OPCOM process. Messages can be directed to the audit journal only.

This command provides information under different sections. Look the Privilege use and Privilege failure information provided in the below section.

System security alarms currently enabled for:

Time

SYSGEN

NCP

Audit: illformed

Breakin: dialup, local, remote, network, detached, server

Privilege use:

BYPASS GRPPRV READALL SYSPRV

Privilege failure:

BYPASS GRPPRV READALL SYSPRV

Security messages generated for the privilege use and privilege failure section will set off alarms to the console thus resulting in huge message flow to OPCOM. The message volume resulted in P0 exhaustion and finally crashing the OPCOM process.

BACKUP of disks was being taken and while accessing each file an audit alarm is generated. Since the number of files being accessed is high, it resulted in huge amount of audit alarms.

You can increase the outgoing rate of the messages to the console to assist avoiding pending messages exhausting memory.

$ SET TERMINAL/PERMANENT/SPEED=(INRATE,OUTRATE).

Problem 2

OPA0 is enabled as an operator only when the operator logs in and REPLY/ENABLE or REPLY/STAT is issued on the OPA0 terminal, it was disabled when operator returns.

Possible scenario

1. The console is a shared console and some other user has disabled the OPCOM messages by issuing REPLY/DISABLE accidently. The console of the system being accessed is an Itanium system.

Solution

To prevent other users from logging into the console, TELNET ACCESS to the console can be blocked via the following procedure:

1. Log in to the MP Console.
2. In the Main Menu type CM.
3. In the CM Menu type SA.
4. Type T for LAN TELNET Port.
5. Type D for disabling the TELNET access.
6. Then type Y to save the configuration.

This disables any user from accessing the CONSOLE over the network using TELNET.

However, the CONSOLE can still be accessed via the COM port.

The OS can still be accessed via the TELNET.

Problem 3

A customer was using CA Console Management for HP OpenVMS. After one node in a three node cluster rebooted some of the entry timestamps displayed by CA console manager were different than the timestamp logged to operator.log.

Possible scenario

The messages for which there was a timestamp mismatch were repeated many times in the CA console manager but logged once in the operator.log. These messages were sent from one node to the other node when a user sent an operator message which required the acknowledgment using DCL “REQUEST” command.

In case of a cluster, a request on a node waits for the response from all the operator’s in the cluster.

For example:

$request/reply "System sending a message and waiting for response"

Above request command broadcasts the message to all the operator’s in the cluster. This message is displayed on all the operator console’s as below:

%%%%%%%%%%% OPCOM   6-MAR-2012 00:51:48.98 %%%%%%%%%%%

Request 7, from user TEST on TESTNODE

_TEST$TNA87:, System sending a message and waiting for response

This message is repeated on all operators’ console every five minutes until the message is replied. Any operator can respond to the message using the DCL “REPLY” command as shown below. While replying to these kind of messages, an operator should use the id displayed with the message in the operator console. For example in the above message, id is 7. So while responding to the above message user can use the id as shown below.

$reply/id=7 "Message received"

In customer’s case, messages not responded to were displayed in the CA console manager many times. Later displays in CA console have an updated timestamp. But the corresponding messages are logged only once into the operator console with original timestamp. This additional message logging with later timestamps made it appear as if there was mismatch between identical messages logged to the CA console manager and operator.log.

Scenarios in which OPCOM messages would help:

1. In debugging issues with Queues, Job Controller, Queue Manager.

   $ SEARCH OPERATOR.LOG QMAN,JBC/WINDOW=(2,4)

   OR

   $ SEARCH OPERATOR.LOG "-E-","-F-"/WINDOW=(2,1)

2. OPCOM messages would report any changes in events like channel open/close, channel movement IN/OUT of equivalent channel set (ECS) for PE driver.

   $ SEARCH OPERATOR.LOG "%PEA0"/WINDOW=(2,2)

3. Triage the issues with Mount/Dismount of disks.

   For example:

   %%%%%%%%%%% OPCOM 8-OCT-2008 08:15:02.83 %%%%%%%%%%%

   Device 1DGA113: (ALFDSK PGA) contains the wrong volume.

   Mount verification is in progress.

   %%%%%%%%%%% OPCOM 8-OCT-2008 09:02:58.35 %%%%%%%%%%%

   Mount verification has aborted for device 1DGA113: (ALFDSK PGA)

4. Troubleshooting issues with UETP and possible corrections.

   For example:

   %OPCOM, 22-JUN-2004 14:10:52.96, request 1, from user SYSTEST

   Please mount volume UETP in device _MTA0:

   %MOUNT-I-OPRQST, Please mount volume UETP in device _MTA0:

Conclusion

System administrators can take advantage of tips discussed in the document to tune the system properly to receive OPCOM message. In addition, they can understand how OPCOM messages would help in troubleshooting issue with devices and the network.

Reference documentation

- HP OpenVMS System Manager's Manual, Volume 1: Essentials
- DECwindows Motif for OpenVMS
- HP OpenVMS Guide to System Security

Learn more

For more information on understanding HP OpenVMS OPCOM messages, visit http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA4-0531ENW.pdf.

Sign up for updates
hp.com/go/getupdated


Rate this document

4AA4-4833ENW, February 2013