hp OpenView operations for UNIX 7.0

MessageStorm Detection White Paper

HP OpenView Operations for HP-UX is a management solution that keeps business-critical application services up and running. It offers sophisticated management functions to improve uptime of all layers of today's distributed IT Service environment: the network, systems, databases, application, and the Internet.

HP OpenView Operations Solutions manage systems and networks, and the services they provide.

Warranty Information

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD PROVIDES THIS MATERIAL "AS IS" AND MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL WHETHER BASED ON WARRANTY, CONTRACT, OR OTHER LEGAL THEORY.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard. This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

Copyright Notices

Restricted Rights Legend

Microsoft® and Microsoft Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

All other registered and unregistered trademarks mentioned within this paper are the sole property of their respective companies.

Table of Contents

About this Paper

This White Paper describes how to configure the OVO for UNIX management server to detect and stop message storms from a managed node. It is assumed the reader is familiar with OVO.

Functionality

An Event Correlation Services (ECS) circuit is used to prevent message storms. Please note that ECS Designer is NOT required to use this circuit as it is.

All messages that arrive at the management server are directed through this ECS circuit. For each node, the rate at which these messages arrive is measured. This rate is measured with a moving interval. If the rate of messages received exceeds the allowed message rate, a configurable action is started. The default action calls a script that stops the node from causing a message storm. Directly after this action has been executed, a critical message is generated which informs that node X caused a message storm and that the configured action has been executed. In parallel, the messages received rate for this node will be reset to zero in order to allow the messages to pass through to the management server again. This reset is performed after a configurable delay that allows the management server processes to process the pending requests. As soon as the circuit sees the first message coming from this node after a storm detection and reset, you will get a message informing you that the message storm is over.

You can configure the circuit so that it does not send the messages that are received by the management server to the message browser until the message storm is stopped.

If you decide to suppress them, you are informed of the number of messages that have been suppressed together with the message informing you that the message storm is over.

Message flow by example

Circuit configuration*:*

- Default interval is 5 minutes
- Default message rate is 0.3 messages per second (If over 90 messages are received in the 5 minutes, a message storm is indicated).
- Default actions are used

Scenarios

Scenario 1:

- Agent recently installed on a web server
- First template distribution
- Long web server log files
- Logfile template configured to always read from start of file

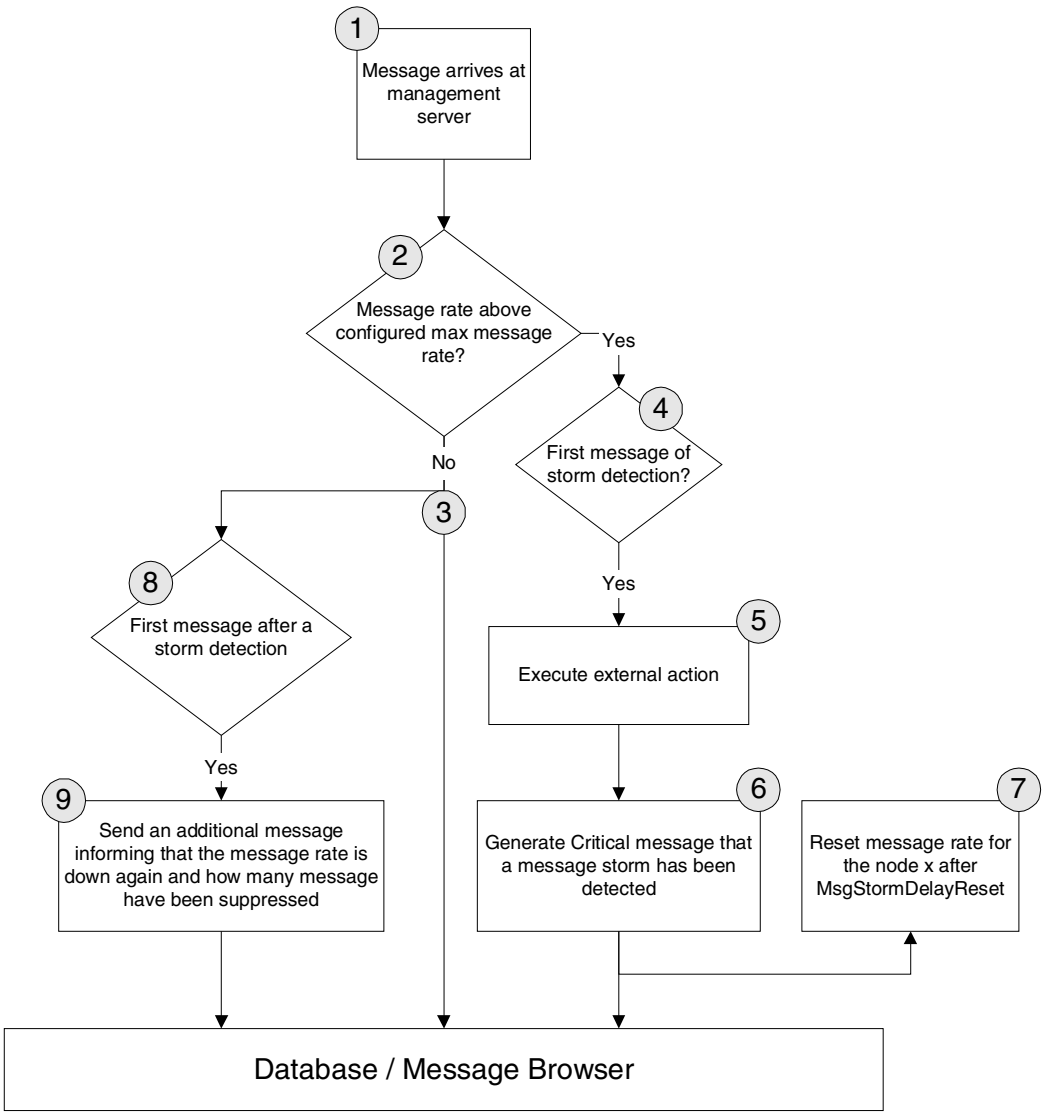
Scenario 2:

- Network segment 10.0.30.x was disconnect from the rest of the network due to a router problem during the weekend
- This caused a lot of problems with various applications in this segment
- Agents in this segment didn't have a connection to the management server during this time
- Agents found a lot of problems and generated messages
- Network problem gets fixed on Monday and agents start sending their messages
- Result is a message flood on the management server

Suppression enabled

Default used, which means message suppression is enabled.

Figure A. Message flow when suppression is enabled



Possible message flows:

- Normal flow 1 →2 →8
- Flow when detecting a message storm 1 →2 →4 →5 →6 →7
- Flow after a message storm 1 →2 →8 & 3 →8 →9

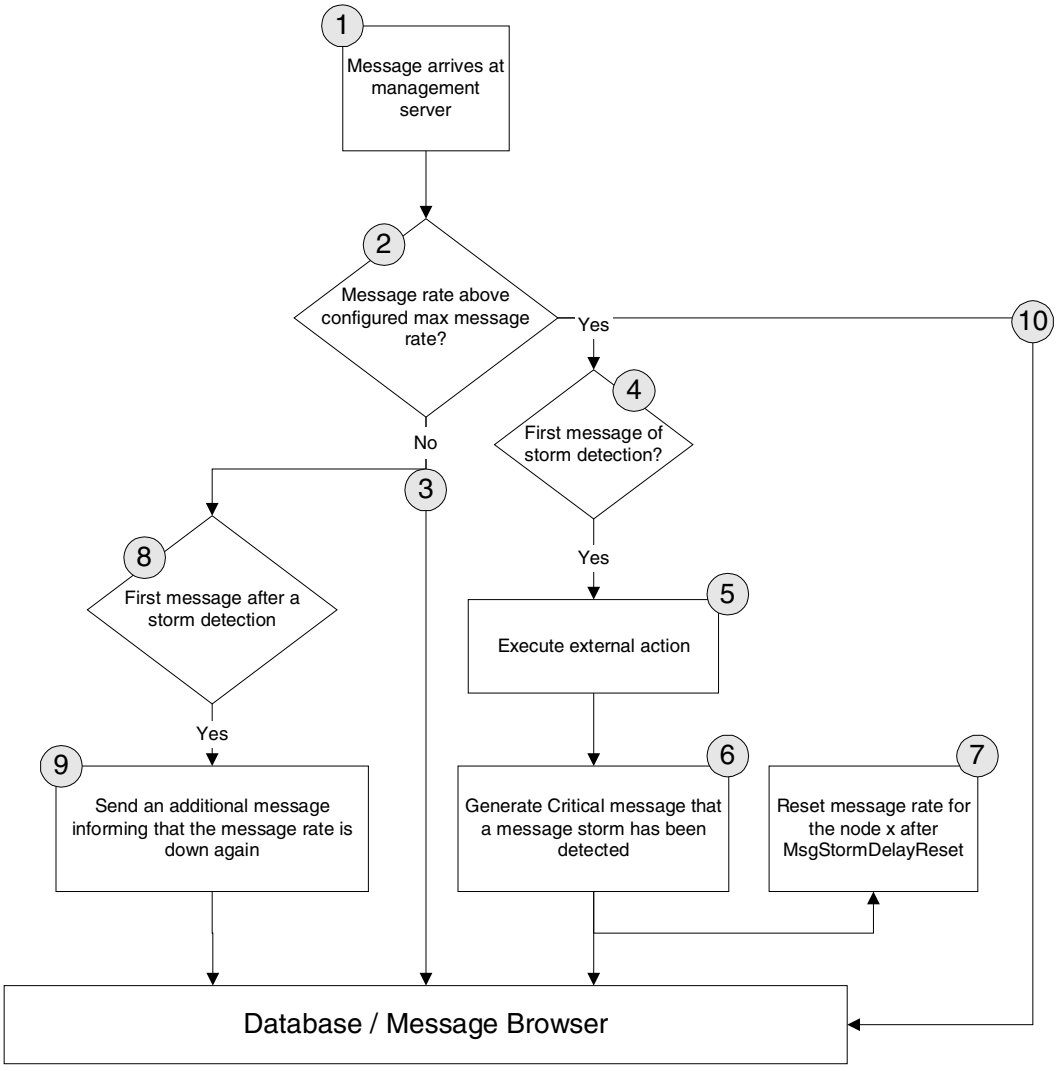
Steps:

1. Messages arrive at the management server and are intercepted by the circuit.
2. The message rate is calculated for each node.
   - The message rate is checked to ascertain whether it is below MsgStormRate, then there was no message storm for this node.
   - In the case that there was a storm for this node, the message rate must be below MsgStormRecoverRate.
3. If no message storm is detected, send the message to the message browser.
4. If message storm is detected, check whether this is the first time detection.
5. Execute the MsgStormAction (when using the default this will stop the agent).
6. Generate the critical message which informs of a detected message storm.
7. With a delay of MsgStormDelayReset, do a reset of the message rate that is stored for the particular node.
8. Messages that are coming from 3 are copied to 7 in order to check whether a recently discovered message storm has actually ended.
9. Generate a message reporting that the message rate of node x has returned to below MsgStormRecoverRate and report the number of messages that have been suppressed.

Suppression disabled

MsgStormSuppress is set to “false”.

Figure B. Message flow when suppression is disabled



Possible message flows:

- Normal flow 1 → 2 → 8
- Flow when detecting a message storm
  1 → 2 → 4 → 5 → 6 → 7 & 2 → 10
- Flow after a message storm 1 → 2 → 8 & 3 → 7 → 8

In addition to the steps described for "Suppression enabled" above step 10 is performed where messages are sent to the message browser even when a message storm has been detected.

Installation

Package Content

The directory /opt/OV/contrib/OpC/MsgStorm consist of the following parts.

- Templates (upload tree for the templates)
- stormstartagt.sh & stormstopagt.sh
  (sample scripts used to start and stop the agent)
- dstore.ds (sample datastore which allows the configuration of the circuit)

Hints

For users, who already make use of the "ECS Management Server" default group, please note that the required ECS template (MsgStorm_Dectect) is placed into this template group after config upload.

In addition, the optional message template (MsgStormMessages) is placed into the "Management Server" default group.

Required Section

1. Upload the templates

   ```
   > opccfgupld -add -subentity /opt/OV/contrib/OpC/MsgStorm/Templates
   ```

2. Configure the management server

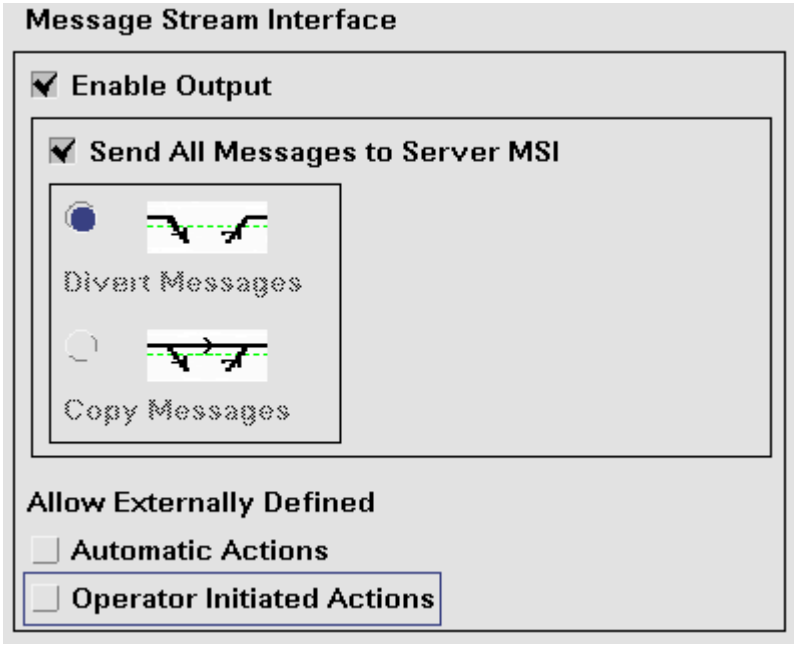
   Enable the MSI on the Management Server. From OVO's Nodebank, select:

   Actions → Server → Configure ...

   and check,

   - MSI, Enable Output
   - Send All Messages to Server MSI
   - Divert Messages

Figure C. Configure Server MSI



3. Assign and distribute the server template

   From OVO's Nodebank, select:

   Actions → Server → Assign Template ...

   and from the message source templates, select either:

   - **/Default/ECS Management Server/MsgStorm_Detect** ECS template
   - **ECS Management Server** default group

4. Next select:

   Actions → Server → Install / Update Server Templates ...

   to run the circuit on the Management Server.

   OVO's Manager Process (opcecm) must now be running. Verify this using the `opcsv(1m)` command.

Optional Section

After finishing the required section, you may proceed with the following optional steps.

Please note this only applies when using the default scripts stormstopagt.sh and stormstartagt.sh. These scripts generate messages that are intercepted by the management server node. The Message Template is mainly used to create message key patterns in order to acknowledge the message storm detection messages.

1. In OVO's Nodebank, select the management server node:

   Actions → Server → Assign Template ...

2. Assign either the "Default/Management Server" template group or only the message template "MsgStormMessages" to the management server node.
3. Next select

   Actions → Server → Install / Update Server Templates ...

   and distribute the templates to the agent.

Configuration

The circuit can be configured according to your needs by using an ECS datastore. The various values listed below can be changed in the datastore.

In order to make use of a datastore, you must place the datastore in the following directory:

`/var/opt/OV/conf/OpC/mgmt_sv/`

For VPO A.06.12, the datastore must have the name dstore.ds. Only a global datastore is available.

With OVO for UNIX 7.0 you can use either:

- The global datastore (dstore.ds)

or

- A datastore that is bound to the circuit itself by using the name ECmsg_storm.ds.

A sample of a datastore file is illustrated below:

```
#/var/opt/OV/conf/OpC/mgmt_sv/ECmsg_storm.ds#03/05/2002#1#0#
ADD DATA ("MsgStormInterval"       , 5m )
ADD DATA ("MsgStormRate"           , 0.3 )
ADD DATA ("MsgStormRecoverRate"    , 0.15 )
ADD DATA ("MsgStormDelayReset"     , 1m30s )
ADD DATA ("MsgStormSuppress"       , true)
ADD DATA ("MsgStormAction"         ,
             "/opt/OV/contrib/OpC/MsgStorm/stormstopagt.sh")
ADD DATA ("MsgStormActionTimeOut"  , 1m30s )
ADD DATA ("MsgStormOperatorAction" ,
             "/opt/OV/contrib/OpC/MsgStorm/stormstartagt.sh")
ADD DATA ("MsgStormMsgKeyPrefix"   , "EC_OVOMsgStormDetected" )
ADD DATA ("MsgStormMaxTransitDelay", 2m )
ADD DATA ("MsgStormServiceName"    , "")
```

Apply configuration changes

- *For VPO A.06.x users:*
  - You can only make the changes in `/var/opt/OV/conf/OpC/mgmt_sv/dstore.ds`
  - In order to apply the changes, you must restart the server processes.
- *For OVO for UNIX 7.0 users:*
  - You can make the configuration either in:
    `/var/opt/OV/conf/OpC/mgmt_sv/dstore.ds` or
    `/var/opt/OV/conf/OpC/mgmt_sv/Ecmsg_storm.ds`
  - To activate your changes the first time call:
    `ecsmsgr -instance 11 -data_load Ecmsg_storm \`
    `/var/opt/OV/conf/OpC/mgmt_sv/Ecmsg_storm.ds`
  - To update your changes call:
    `ecsmsgr -instance 11 -data_update Ecmsg_storm \`
    `/var/opt/OV/conf/OpC/mgmt_sv/Ecmsg_storm.ds`

Configuration variables

MsgStormInterval

Type	Duration
Default	5m
Format	<hour>h<minute>m<second>s

Description

The time period over which the message flow is analyzed.

MsgStormRate

Type	Real
Default	0.3

Description

In conjunction with the MsgStormInterval, the MsgStormRate specifies how many messages may arrive within the MsgStormInterval time period before a message storm is detected.

The rate (in messages per second) is calculated as follows:

$$\text{MsgStormRate} = \frac{\text{Number of messages received in MsgStormInterval}}{\text{MsgStormInterval (in seconds)}}$$

If the MsgStormRate measured over the MsgStormInterval is higher than the maximum allowable rate configured, it is considered to be a message storm.

For example, if you have selected a MsgStormRate of 0.3 and a MsgStormInterval of 5 minutes, the circuit will report a message storm as soon as there are more than 90 message within the 5 minute period or less time.

MsgStormRecoverRate

Type	Real
Default	0.15

Description

If you have chosen to suppress messages during a message storm, this value can be used to define the message rate under which a node has to fall for the circuit to assumes that the message storm is over.
Using the default values, the message rate must fall below 45 messages within 5 minutes.
For the calculation of the MsgStormRecoverRate see MsgStormRate.

MsgStormDelayReset

Type	Duration
Default	1m
Format	<hour>h<minute>m<second>s

Description

Delay used between detecting a message storm and resetting the message rate to 0 to allow the next messages to pass through.
This delay suppresses further messages from the same node that caused the message storm as they may already be in the server queues but not processed by the message manager.

MsgStormSuppress

Type	Boolean
Default	true
Format	true | false

Description

true – Prevents flooding the database and the message browser with messages.
false – Generates a message informing of the message storm but all messages are still sent to the database and to the message browser.

MsgStormAction

Type	String
Default	"/opt/OV/contrib/OpC/MsgStorm/stormstopagt.sh"

Description

Command to call within the circuit as soon as a message storm has been detected. The command will be called with the following parameters:

<Action> <nodename> <MSGID>

MsgStormActionTimeOut

Type	Duration
Default	1m30s
Format	<hour>h<minute>m<second>s

Description

Time period that the circuit waits for the <action> to execute. After this, the circuit will proceed and will no longer wait for the application end. Please note that the application won't be stopped but the return code and output will be ignored.

MsgStormOperatorAction

Type	String
Default	"/opt/OV/contrib/OpC/MsgStorm/stormstartagt.sh"

Description

Operator initiated action which is assigned to the message storm warning to run on <$OPC_MGMTSV>. The default action allows the operator to restart the agent that caused the storm.

Following call is used:

<Action> <nodename> <MSGID>

MsgStormMsgKeyPrefix

Type String

Default "EC_OVOMsgStormDetected "

Description

Prefix to add to the message key used in the message generated by the circuit. There are two different formats of keys generated:

1. <MsgKeyPrefix>:start:<nodename> is generated when the message storm is detected.
2. <MsgKeyPrefix>:end:<nodename> is generated when the circuit detects that the message storm is over.

MsgStormMaxTransitDelay

Type Duration

Default 2m

Format <hour>h<minute>m<second>s

Description

The ECS engine checks whether incoming messages are older than this MaxTransitDelay. Older messages are not processed with the circuit.
The delay is the difference between the message arrival time on the server and the current time.

MsgStormServiceName

Type String

Default ""

Description

Service name that should be used within the messages generated by the ECS circuit. As soon as this string is defined, the circuit will generate a service name with the format <MsgStormServiceName><node>.

Limitations

There are some limitations when you apply this circuit.

Using several ECS circuits on the management server

The ECS engine always processes all circuits in parallel. This means, when you have more than the message storm ECS circuit, the other circuits may also process the messages and pass them to the message browser even when a message storm has been detected. All actions described above will be taken and you will also get a message about a detected message storm but you may still receive messages from that node even when you selected to suppress all messages after a message storm has been detected.

Message Storm caused by the agent on the management server

In case the agent on your management server generates a message storm and you use the default action or your own script to stop the agent all operator-initiated actions which are added to the message storm detection warnings will fail since they would execute on that node.

The consequence is that you have to restart the agent on the management server manually.

To avoid this you can modify the stormstopagt.sh script to behave differently when stopping the node on the management server.

Proxy nodes

In the case of a node acting as a proxy for another device that doesn't have an agent installed, the proxy node is stopped when using the default scripts. This may happen when using the OS390/SPI that sends all message by using the node on the management server.

This can be handled by modifying the stormstopagt.sh script.

Default Values

The default values that are used to define whether a message storm occurs or not might not be the best for your environment and can be changed according to your needs.

Appendix Messages generated by the ECS circuit

1. Message storm detection

1.1 Action successful executed

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message Storm detected on <node> Rate is: <configured message rate> AutoAction successful executed, Output: <action output> Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node, it might be better to remove the file to prevent a reoccurrence of the message storm after agent start
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

1.2 Action returned an exit code different from 0

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message Storm detected on <node> action <MsgStormAction> failed, ExitCode: <Action exit code> Output: <action output> Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node, it might be better to remove the file to prevent a reoccurrence of the message storm after agent start
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

1.3 Action timed out

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message Storm detected on <node> automatic action <MsgStormAction> timed out (time out is set to <MsgStormActionTimeOut>) Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node, it might be better to remove the file to prevent a reoccurrence of the message storm after agent start
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

2. Message rate below configured threshold level

2.1 With suppression

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message rate from node <node> went down under the configured rate of <MsgStormRecoverRate> after a Messagestorm has been detected. <suppress count> messages have been suppressed during the storm. The message storm that has been detected for this node appears to be over as the message rate of this node is again below the configured recover rate. It is possible that the storm did not end but the rate has been reset since the storm duration is longer than the configured reset delay after storm detection.
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

2.2 Without suppression

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message rate from node <node> has dropped below the configured rate of <MsgStormRecoverRate> after a Messagestorm has been detected. The message storm that has been detected for this node appears to be over as the message rate of this node is again below the configured recover rate. It is possible that the storm did not end but the rate has been reset since the storm duration is longer than the configured reset delay after storm detection.
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

Please note that the service name is only created when MsgStormService is set.

www.openview.com

Publication Date: 04/2002