Late-Breaking HP Software Operations (OVO) News

Greg Baker – gregb@ifost.org.au

August 23, 2007

Contents

1 Getting to 75% in 15 minutes

There are two major complaints most customers have about HP OpenView Operations (HP Software Operations):

- The vast quantity of clutter that the operators have to wade through to find any important events they are supposed to respond to. This leads to disillusionment and disinterest.
- The very long deploy times, and the even longer return-on-investment times. My experience is that it takes up to three years to get a return on investment on an Operations deployment.

It doesn't need to be this way. In fact, the infrastructure is in place to bring almost all the useful alert messages to operators very, very simply.

Firstly, most of the clutter comes from logfile readers. If you have already installed OVO agents on a few nodes, run `opctemplate -l` on a few of them, and identify the log file templates (or log file policies if you are using OVO/W). Disable them with `opctemplate -d`, or better still, de-assign them.

Now create a new logfile template called "Unix logs". For the file name, use logfile discovery. I use a script like this:

```
<‘find / -local -type f \( -path
’*/log/*’ \) -o \( -name ’*.log’ \) -print
‘>
```

The `find` command takes 20-30 seconds of CPU time on a modern HP-UX installation. Searching the log files also takes some time, so schedule this only to run every 5-10 minutes.

Set a default severity of "major".

Create one condition, which looks for a space, followed by any one of the following words, followed by another space:

Negative words couldn't – wasn't – don't – can't – won't – didn't – not

Termination words denied – failed – abort – unsuccessfully – permission – expired

Capacity words enough – full – empty – missing – all

Adverbs such – due

Create a hidden node group for all Unix boxes, and assign this template to that node group. Distribute.

There will be an initial burst of lots of messages (because of the number of pre-existing messages in each server's log files). After 15 minutes, acknowledge everything.

Show your operators the new, clean message browser, and get them to tell you about any unnecessary messages they want suppressed. There won't be very many, and the operators will miss very few events of any importance.

2 Notification

There are several commercial solutions:

PageAlert `http://www.nobix.com/pagealert.phtml` – quite cheap, can do voice, email, pagers

OpenLinkXS `http://www.hiplinkwireless.com/products/openlinkxs` – can also look up events and let mobile users interact with OpenView somewhat

TeleAlert `http://www.telearlert.com/` – long-standing, well-known in OpenView community

PageManagerPro `http://www.atlantissoftware.com/pageman.shtml` – quite cheap, can use wireless modems directly, supports escalation, has a personnel schedule (roster); can do SMS, pagers , email.

3 More Administrators

Sometimes it is helpful to have more than one "proper" administrator. While it is possible to have many template administrators (and obviously many operators), there is no way using the GUI to add additional users with privileges equivalent to the opc_adm account:

- add other users
- push software out to other systems
- assign templates to systems
- any of the other tasks normally done by `opc_adm`

.

The following procedures have some side-effects, but they at least give the capability to other users.

We will be spending a lot of time manipulating the Oracle database manually here. To do so,

1. Log in as root
2. Run `su - oracle`
3. Run `sqlplus "connect /nolog as sysdba"`
4. Each time we run a command, we will type it in at the prompt, terminated by a semi-colon.

To get some background on what we need to do, run the following SQL command:

```
select name, user_role
  from opc_user_data;
```

This should produce a table something like the following.

```
NAME                 USER_ROLE
-------------------- ----------
opc_adm                       1
itop                          0
netop                         0
opc_op                        0
bgates                        0
smcneally                     0
cfiorini                      0
gbaker                        3
nightshiftprofile             4
dayshiftprofile               4
10 rows selected.
```

From this you can guess the appropriate numbers:

- Ordinary operators have a user_role of 0
- Template administrators have a user_role of 3
- A profile has a user_role of 4
- The *opc_adm* account has a role of 1 (and is the only account to do so)

So. . . first create a a profile called `other_adm_profile` (you will need it later). Next create a user who is a an ordinary operator (go into the user bank and select Add from the Actions menu). Call it (say) `other_adm`, and assign `other_adm_profile` to it. Check that this has been set up correctly, by logging in as this user.

Log out again, and change the user role:

One last time in `sqlplus`,

```
update opc_user_data
    set user_role = 1
  where name = 'other_adm';
commit;
```

Now it should be possible to log in as `other_adm`. You will have access to all the menu items and windows of `opc_adm`. The "Root" window will have two top levels – "OVO Node Bank" and "Managed Nodes", rather than the usual set for `opc_adm`.

The only annoying side effect of this (as of version 8.1) is that it is impossible to alter `other_adm` through the Node Bank. Which isn't really a problem – you created a profile called `other_adm_profile` and included that profile in `other_adm`. If you want to make changes, make them in the `other_adm_profile`.

In previous versions, you would receive a number of error popups complaining that the user is undefined (that it perhaps does not exist in the user bank). It appears to be safe to ignore these messages. Deliberately saving the geometry of all windows seems to remove one of the messages.

4 Annotations

When using the trouble-ticketing interface in VPO, normally there is no way of providing a cross reference into the help desk system. This means that an operator looking at a message can only see that the message exists in the system, but not see any more detail about it. A solution to this is to use `opcaddanno`.

For this to work, the trouble ticketing script needs to be able to get a reference number out of the help desk system. When it does, the script grabs this (into a variable $ID, for example) and runs

```
opcaddanno $1 Ḧelp desk id is $ID¨
```

The trouble ticketing script is always passed $1 as the message reference number in the VantagePoint database, which is a necessary parameter for `opcaddanno`. `opcaddanno` adds an annotation to the message in VantagePoint of the second argument – in this case a little message about the identifier in the help desk system.

5 Changing the colour scheme

Red-green colour-blindness is quite common among Caucasian males. Unfortunately, this prevents an operator from ever seeing any difference between a network going critical and one which is normal!

Fortunately, X-Windows is designed to cope with this quite well, as colour and font information is just a “resource”, configurable independantly for each application and each user. The model for defining this information changed with X11R2 (some time ago), but the following procedures should work for any release.

In your home directory there should be a file `.Xdefaults`. This sets resource values for X-Windows applications. Each application has its own name (e.g. `OVw`), and creates a heirarchy for each individual widget. Wildcards can be used.

Append the following lines to the `.Xdefaults` file, or create a new file if one does not already exist.

```
OVw*unmanagedStatusColor:   #fff5e1
OVw*testingStatusColor:     salmon
OVw*restrictedStatusColor:  #8d6057
OVw*disabledStatusColor:    #5A2322
OVw*unknownStatusColor:     #727cff
OVw*upStatusColor:          #00e626
OVw*warningStatusColor:     #41ffff
OVw*marginalStatusColor:    #ffff29
OVw*majorStatusColor:       #ff8500
OVw*downStatusColor:        red

OVw*unmanagedStatusLineColor:  black
OVw*testingStatusLineColor:    salmon
OVw*restrictedStatusLineColor: #8d6057
OVw*disabledStatusLineColor:   #5A2322
OVw*unknownStatusLineColor:    black
OVw*upStatusLineColor:         black
OVw*warningStatusLineColor:    #41ffff
OVw*marginalStatusLineColor:   #ffff29
OVw*majorStatusLineColor:      #ff8500
OVw*downStatusLineColor:       red

*.normalSeverityEventColor:   #00e626
*.warningSeverityEventColor:  #41ffff
*.minorSeverityEventColor:    #ffff29
*.majorSeverityEventColor:    #ff8500
*.criticalSeverityEventColor: red
```

This should be re-read the next time you start X-Windows (i.e. login and logout again). To use this immediately, run `xrdb -merge .Xdefaults`. Then start `opc`, and the colours should be subtly different. The OpenView development team tested the above colour sets and found that they were sufficiently distinct for people with no capability for distinguishing red and green.

Surprisingly, total colour-blindness is handled much more easily – simply buy a grey-scale monitor. At startup OpenView will attempt to start requesting colours (for the initial logo), fail immediately, and fallback to 8-bit greyscale.

6 Database Maintenance

The Administrator's Reference recommends that you periodically reorganize the database, by running `opcdbreorg (1M)`. This frees unused pages as well as restructuring the index tables. For more details, it says, see the man page. What this does is a complete export and import of the database. The official HP word is that it is important to do this on a regular basis, but in practice this is not always necessary.

Firstly, if (for performance reasons, or for more efficient use of disk space) there are any table spaces which use two or more data files, `opcdbreorg (1M)` will fail.

Here is a quick SQL command that can be used to check it.

```
select  count(file_name),
        tablespace_name
from
        dba_data_files
group by
        tablespace_name;
```

If any tablespace has a number more than 1 associated with it, you will not be able to run `opcdbreorg`.

Secondly, if you are running Oracle 8.0.5 or onwards, the level of degradation due to transactions is very low. There are many sites that run Oracle on production servers with very high transactional loads (e.g. financial and stock management roles) who do not have the downtime to afford to do a complete rebuild. They report that it takes several years to start seeing noticable performance problems.

Unless there is a major runaway (such as a logfile template picking up hundreds of thousands of entries), the level of transactional throughput from Operations is much, much lower than many other "real world" applications. So missing out on running `opcdbreorg` may not necessarily affect a Operations installation over the life of a given machine.

For details about the Oracle database, see the documentation supplied with the database and the online documentation in `/opt/OV/ReleaseNotes/opc_db.tuning`.

7 Database Size

These are just some rough figures. It will depend on the actual content of the messages (e.g. if you have very long messages, you may see more space used up).

The main Oracle tablespaces of importance are

- OPC_1 and OPC_2 – these contain information about the unacknowledged messages
- OPC_6, OPC_7, OPC_8 and OPC_9 – these contain information about the acknowleges messages.

My quick test was to generate 10000 messages on IT/0 5.10 running against Oracle 8.0.5. Oracle was configured to auto-extend tables – however, it will often extend to have more than enough space. So these figures may be overly pessimistic about the space usage of messages.

These are not particularly accurate, nor the best way of getting accurate information out of Oracle!

Tablespace	**Size (bytes)**
OPC_1	10493952
OPC_2	5242880

From this we can calculate that each message takes up roughly 1 kbytes in opc_act_messages. This table just contains the message identifier number, time of reception and such like.

The actual content of the message is spread across tables in the OPC_2 tablespace – mine were from quite short messages, so the 500 bytes per message is a little less than would be expected.

When messages get acknowledged, they move out of these tables, and take up a similar amount of space in OPC_6, OPC_7, OPC_8 and OPC_9. (These is a small amount of extra space used up for "time of acknowledgement" and "acknowledged by".

When messages are downloaded (configured through the database maintenance menu item on the Node Bank), they are plain-text files, one message to a line.

10000 messages turn into 3.5MB of data. `gzip` will compress this down to around 1.2 - 1.8 MB.

8 How to use a vendor-supplied MIB file with OpenView Operations

Manually creating SNMP conditions in a trap template is a job for people with very accurate typing skills, an eagle eye for typos and a great deal of patience.

The rest of us cheat and load the MIBs into Network Node Manager, and get it to export out in `opccfgupl`-compatible format.

Suppose you want to load `cisco.mib`.

Run `xnmloadmib -event -trapType -load cisco.mib`.

Run `ovtrap2opc /etc/opt/OV/share/conf/C/trapd.conf NNM-Traps NNM`.

Answer "Y" to the questions about uploading and replacing existing templates.

Assign the template to your OVO management server:

```
opcnode -assign_templ templ_name=NNM-Traps
templ_type=SNMP_TEMPLATE
node_name=‘hostname‘ net_type=NETWORK_IP
```

And distribute:

```
opcragt -distrib -templates
```

9 Using `opcmon` in C

There is both a program called `opcmon`, and also a function call called `opcmon`.

If you are writing a C-language program to do some kind of monitoring (which is the best thing to do if you are worried about the overhead of your management system), the function call incurs *very* low overhead.

Here is a program that gives you an idea of how long it takes for a small program on your system to start up.

```
#include <stdio.h>
#include <sys/times.h>
#include <opcapi.h>

int main (argc, argv)
    int argc;
    char * argv[];
{
  struct tms process_times;
  int rc;


  /* Find out how much time this
     process and all it's children
     have consumed now  */
  times(&process_times);

  /* Send this as an OpcMon message */
  rc = opcmon("startup_time",
     /* user time + system time */
      (double)
       (process_times.tms_utime+
        process_times.tms_stime)
      );

  /* Did that work? */
  switch (rc)   {
  case OPC_ERR_OK: /* okay */
    break;
  case OPC_ERR_OBJNAME_REQUIRED:
    fprintf(stderr,"Name␣is␣empty.\n");
    break;
  case OPC_ERR_NO_MEMORY:
    fprintf(stderr,"No␣memory\n");
    break;
  case OPC_ERR_NO_AGENT:
    fprintf(stderr, "Agent␣not␣running.\n");
    break;
  default:
    fprintf(stderr, "Error␣%d\n", rc);
    break;
  }
  return 0;
}
```

9.1 VNC

VNC is a free product (under a Gnu License) which allows cross-platform desktop takeover of MS-Windows, MacOS, OS/2 and a few other platforms. It also acts as an X-server when used to "control" a Unix system. The viewer component runs on many more platforms, ranging from all of the above, plus BeOS, Windows CE and simple Java devices.

Precompiled binaries for HP-UX 11.11 (PA-RISC) are available from `http://www.realvnc.com/cgi-bin/download.cgi`. It works happily in emulation mode on Itanium

HP-UX systems. This can make a very useful addition to a MS-Windows management system:

- Install the VNC service on your MS-Windows systems. It requires Service Pack 3 on WinNT (no requirements for Win-9X or 2000) and has no known conflicts.
- Write the following shell script, and store it in some sensible place.

  ```
  #!/usr/bin/sh
  for system in $*
  do
   xterm -e vncviewer $system:0 &
  done
  ```

  (`vncviewer` asks for a password from standard input which forces us to start an `xterm` window to get this).
- Add a new application to the `NT Tools` Application Group, with command being the above shell script, to be run as a **X-Windows Application** on the management server.
- Assign this new application (if it were not already available to your operators) to each operator's application bank.
- Now operators can do desktop takeover by simply dragging the system over the icon and providing the right password.

Better still, this can potentially be used as the "remote console" for a MS-Windows system. (Select the system in the Node Bank, right click and select modify, then choose "advanced" properties – fill in the remote console command.)

There is also a Java client, delivered from port 5800 of a VNC-served machine, so a slight alteration of the above could be used for a Network Presenter Registration File, giving the same capabilities to the NNM web interface.

On a related topic, but the other way around, the VNC server on HP-UX acts like an ordinary X-server, and can be accessed with any VNC viewer. This makes it possible to run the "native" `ovw` within Xvnc, which can then be accessed with the Java-based VNC viewer in any web browser.

10 OpenView Operations Commands

10.1 Backup

- **opc_backup** – Backup and restore the entire ITO configuration on the Management Server.
- **opc_recover** – Restore the entire ITO configuration on the Management Server (previously backed up using opc_backup)
- **opccfgdwn** – Download configuration data from the ITO database to flat files.
- **opccfgupl** – Upload configuration from flat files into the ITO database.

10.2 Messages

- **opcack** – Troubleshooting tool to externally acknowledge active messages assigned to an operator.
- **opcackmsg** – Externally acknowledge active messages using message IDs.
- **opcackmsgs** – Troubleshooting tool to externally acknowledge active messages according to specified message attributes.
- **opcmack** – Acknowledge an ITO message by specifying the message ID.
- **opchistdwn** – Download history messages to a file.
- **opchistupl** – Upload messages previously saved off to a file, and put them into the database.

10.3 Managed Nodes

- **opcactivate** – Activate an ITO pre-installed agent.
- **opcagt** – Maintain ITO agent services on current system.
- **opcragt** – Remotely maintain ITO agent services on Managed Node(s), from Management Server.
- **opcchgaddr** – Change the node address of nodes in the ITO database.

- **inst.sh** – Install ITO software on managed nodes.
- **opcnode** – Put nodes into node groups, and assign templates
- **opctemplate** – Enable and disable templates

10.4 Outages

- **opc_so_sh** – Not a command at all. It is just a man page giving an overview of configuration of scheduled outages and service hours.
- **opccfgout** – Configure condition status variables for scheduled outages in ITO.

10.5 Management Server

- **opcconfig** – Configure an ITO management server.
- **opcdbreorg** – Re-organize the tables in the ITO database.

10.6 Templates

- **opcdcode** – View ITO encrypted template files (version 7 only).
- **opcmon** – Submit the current value of a monitored object to the ITO Monitoring Agent.
- **opcmsg** – Submit an ITO message to the ITO Message Interceptor.
- **opcpat** – Test program for ITO pattern-matching.
- **opctmpldwn** – Download and encrypt ITO message source templates.
- **ovtrap2opc** – Convert trapd.conf file and the ITO template file.

10.7 Interfaces

- **opc** – Graphical user interface for HP OpenView IT/Operations
- **ito_op** – Java-based GUI for IT/O.