CHM/OS User Interface (CHMOSG)

Previously known as IPD/OS (Instantaneous Problem Discovery) or as CRF (Cluster Robustness Framework).

NOTE: This CHM/OS Graphical User Interface (CHMOSG) is not a part of any Oracle Grid Infrastructure release. It ships separately on OTN. Please review the included README for required O/S level and Grid Infrastructure versions. Beginning with the 11.2.0.2 release of the Oracle Grid Infrastructure, CHMOSG must be installed and run on a local node of the cluster to be monitored. The standalone version of CHM/OS provides a proxy process, oproxyd, that supports connecting to ologgerd from a remote node outside the cluster. Running CHMOSG remotely has the advantage of reducing the CPU usage that a Java-based GUI can impose on a monitored node; however, the process names and associated metrics are not encrypted and thus would be transmitted on the public network.

CHM/OS Graphical user interface (CHMOSG) is invoked using the chmosg script available in the bin directory of the CRF Home. CHMOSG is designed to be small and simple; minimizing buttons and menus. It communicates with a metrics archiving daemon, ologgerd, that in turn receives metrics from each cluster node's osysmond daemon. A series of intuitive displays allow you to browse through metrics, sort them, analyze short term history, and observe trends and alerts.

### 1. CHMOSG Command line options and reference

List of command line options can be obtained using the "-?" or "-h" option. An example:

```
node:/home> chmosg -?
Cluster Health Monitor V2.05

Usage: chmosg [flags] [params]
    -h    print this help
    -?    print this help

    -m <node>    Name of the node to connect to
    -d <when>    Replay earlier data
        <when> can be:
                    <mm>:        minutes ago
                    <HH:mm:ss>:  seconds ago specified by hours,min,sec

    -r <sec>     Refresh rate
    -W <sec>     Maximum poll time for connection

    -l          With ologgerd data V1.x show histogram tables with cluster view

    -i [<mode>]  interactive with cmd prompt
        <mode> comma separated list with these options:
                    nogui:     Query mode without windows
                    incopy:   Verbatim copy of incoming data, incl. trailing blanks
```

The following are detailed descriptions of each option and its use.
Use option `-m` to specify to the node to connect to:

```
$ chmosg -m <nodename>
```

where `<nodename>` is any node where CHM/OS v1.x is installed when running remote from the cluster. If running from any server within the cluster, CHMOSG will automatically find the node running ologgerd and connect reporting the specific node. The `-m` option can be used inside of the cluster as well, for example when starting CHMOSG on a node on which the osysmondd does not run anymore, or does not respond.

Use the `-d` option to replay data collected at an earlier point of time specified by a time difference from cluster’s current time.

```
$ chmosg -d <when>
```

where `<when>` is a duration specified in minutes from current time, `<mm>`: or as a fully specified duration with seconds resolution, `<HH:mm:ss>`. This mode of operation is also called historical mode. Data is then delivered as if it were occurring in real time. To switch to a different time period or timestamp start CHMOSG with a different `-d` option.

Use the `-r` refresh rate option to specify how often often CHMOSG requests metrics from ologgerd.

```
$ chmosg -r <sec>
```

The default refresh rate is 1 sec. Please see Section 1.1 Choosing Refresh Rate before using this option. When using the CHMOSG with assistive technology, such as a screen reader, a refresh rate greater than 15 seconds is recommended.

Use the `-W` poll period option to adjust the maximum amount of time CHMOSG waits for a response.

```
$ chmosg -W <sec>
```

The default maximum time for CHMOSG to wait for a response from ologgerd is 600 seconds. This default may not be appropriate when running locally or with a lightly loaded cluster.

Use `-l` option to show with ologgerd V1.x data to show histogram tables.

```
$ chmosg -l
```

When connected to 1.x versions of CHM/OS (previous text table oriented cluster view) this option will launch a graphical style cluster view and associated histogram tables.

Use the `-i` option to activate the interactive mode.

```
$ chmosg -i
```

Activates the "chm> " command line prompt in the OS terminal, such as "cmd.exe" in Windows or a shell Unix terminal.

The command line option is a facsimile of the same prompt provided in most of the CHM/OS view windows, such as cluster, node or device view. The same commands may be entered in each of these windows. The text response will be directed to the same window in which the command was entered. This mode is useful as it provides a dedicated window region for commands and responses distinct from messages and alerts. For example, when the CHMOSG receives an alert, it will appear in the scrollable area of the corresponding view and may
interfere with the reply to a command entered manually.

In such situations usage of the otherwise quiet terminal window is clearly preferred. Another advantage is that the user may benefit from using the terminal’s own line buffer and use OS facilities to log terminal activity. This option can also be used to start the UI without any windows; for example, in batch test scenarios.

1. Cluster View: Connecting and Monitoring

To start CHMOSG the command chmosg (chmosg.bat on Windows) is executed from the CRFHOME/bin directory. No additional parameters are required if the CRFHOME is on a node of the cluster being monitored. If the CRFHOME is on a remote node, then the -m <nodename> parameter is required where nodename is a node in the cluster.

Once started and connected to a cluster node, the following Cluster View is displayed:

CHMOSG shows the data in a collection of views. Each view has a set of vertically arranged sections. The common components are text headers and tables with metrics. The content of each view is determined by the cluster ologgerd, not by CHMOSG itself. Thus the layout of the views and their content may vary based upon the Oracle cluster version. CHMOSG is designed to be backward compatible and is able to adapt to many differing sets of data. However, this compatibility is not guaranteed as CHMOSG is not an officially supported Oracle product.

The initial view is called the primary view which is typically the Cluster View. Only upon closing the primary view will the entire CHMOSG will terminate.

The Cluster View contains the following sections and examples of their content:

**Title Bar**: Program name and version; Communication protocol version; cluster name (if applicable); view number; and refresh rate.

**Overview**: Cluster timestamp; Client timestamp; and number of monitored nodes.

**Top Resource Consumers**: Table of top process resource consumers by node. Resources evaluated include CPU utilization, private memory, shared memory open files and threads.

**Cluster Stats**: Table of overall physical resource configuration and use per node. Resources
reported include #CPUs, CPU Utilization, CPU Queue length, Physical memory free, Cache Memory used, Swap Memory free, I/O KBps Read, I/O KBps Writte, I/O Requests, Net I/O Read KBps, Net I/O Write KBps, # Processes, # Real Time Processes, # Open Files, # Disks, and # of NICs.

**Unresponsive Nodes:** Last known cluster stats for one or more nodes that are marked as unresponsive.

**Scrollable Text Area:** Various text messages associated with alerts will be displayed. These alerts will include the specific metric and its value, the associated process or device, specific issue being raised.

**CHMOSG Command line prompt:** The text field where interactive commands may be entered.

Values exceeding safe or recommended use of system resources may be accompanied by alerts. They appear marked in color **red**, **orange** or **yellow**, dependent on the severity of the alert. In the following example on node **strdu10** the process **sqlplus** uses 99.9% of the CPU. In addition to the colored values in the table cells, the text of each alert appears in the scrollable text area. In most cases the alert text contains the following information:

- The alerted metric and its value,
- The associated entity - process or device,
- The nature of the issue - too high, too low, too many, etc.
- The timestamp and level of the alert.

When a view becomes inactive, by either an unresponsive node or a loss of connection, CHMOSG keeps the window open to preserve the last received set of data. Indication of the inactive status is a gray overlay of the view window:
1.1 Choosing the Refresh Rate

CHOMSG has default refresh rate of 1 second. This rate represents the frequency of request per second that it makes to ologgerd. This rate is independent of the cluster sample rate. The "-r <sec>" option is available to change the default refresh rate. When the CHMOSG refresh rate is slower than the cluster sample rate the view will show the current data sample not an average of samples that occurred since the last refresh. When the CHMOSG rate is faster than the cluster sample rate, the view will consist of nodes that have had there updates completed and nodes that will have the same data from the previous request. The cluster sample rate is the following:

- Oracle Grid Infrastructure 11.2.0.2: 1 second
- Oracle Grid Infrastructure 11.2.0.3+: 5 seconds
- CHM/OS OTN Release: 1 second

NOTE: The cluster sample rate is only the minimum sample frequency. When a cluster or node is under stress, the actual sample period may be significantly longer.

1.2 Unresponsive Nodes

NOTE: Using refresh rate of 5 sec. Warning: A refresh rate other than 1 sec will display 1 sec averages every 5 sec.
If one of the cluster nodes becomes unresponsive, which may be a result of an eviction or a heavy load, ologgerd will remove rows for that node from the both tables and display a row in a table of Unresponsive Nodes:

The following shows node staiu13 is unresponsive, 2 other nodes are still sending metrics. Rows in the Unresponsive Nodes table show their last collected metric values.

CHMOSG's metrics tables are limiting in that they only show the most recent value and do not report trends. Please refer to Section 3 to learn how to use the same Cluster View but represented as scrolling histograms, that support dynamically observing short-term trends.

1.3 Interactive Operations on Text Tables

Metric and attribute data are represented in tables. To simplify interaction between CHMOSG and the cluster, interactive mouse operations are available from columns where its values identify a node or device. Clicking on cells in these columns retrieves specialized views. Columns can also be sorted in various orders. The following is a summary of supported mouse operations:

1. Double Click on a cell with a node name: Retrieves a node view.
2. Double Click on a cell with a device name: Retrieves a device view.
3. Click on column header: Sorts rows by the values in this column in descending order. Any other sort criterion will be reset.
4. Shift-Click on column header: Sort rows by the values in this column in ascending order. Any other sort criterion will be reset.
5. Control-Click on a column header supports multiple functions:
   a) Column has sort activated: Will toggle the sort order descending vs. ascending.
   b) Another column has one or more sort(s) activated: Will add an additional sub-sort order to a column thus preserving the existing sorts.

A sort order in a column is indicated by a small triangle pointing up or down in the direction of the sort. The following example is a portion of a Node View with 3 columns using a custom set sort criterion:
CHMOSG sorts in the order of the mouse clicks and not the position of the columns. In the above example, the sort is ordered first by the CPU column, then the PRIV-MEM column and finally the SH-MEM column.

Note the default sort order is in descending order for numeric values and alphabetical order for names.

1.4 Refreshing Windows and Exiting from CHMOSG

To exit CHMOSG, use the "chm> quit" command, or close the window of the primary view or Cluster View. When entering quit or closing a window of any other view, only that window will be closed. When closing an active Histogram View, entering the command refresh will reopen it. However, if the view itself was quit, then it must be retrieved again by double-clicking the appropriate cell in the Cluster View as the refresh command will not restore it.

1.5 Adjusting Text Size

In order to support vision impaired users or to allow monitoring from a larger distance CHMOSG provides a chm> zoom in/out option that increases/decreases the size of the fonts. The following screen shots display the two font sizes available.

Standard font size followed by zoomed in size:
2. Command Line Prompt and Text Terminal

The command line prompt visible in the Cluster View window supports a set of text commands and database queries. It can be accessed also from the local terminal when the option '-i' is being used as follows.

```
chmosg -i
```

To see the terminal prompt you will need return focus to the terminal and then press the Enter key.

When started with this option, both the Cluster View and the terminal have access to the "chm>" prompt and the same set of commands. Enter a question mark (?) to see the list of available commands:

```
chm> ?
```

```
connect           Attempts to connect after expiration of poll period
device  <name>    Display Device View of the device  <name>
display <item>    Display one of the following:
                    [no]grid      paints horizontal lines in tables
                    [no]strip     paints alternating color background in tables
list    <item>    Lists one of the following:
                    alerts        collected alerts
                    inputs        IO connections and their status
                    nodes         nodes known to UI
```
Commands can be abbreviated, for example instead of using the command "status" use the first letters "st":

```
chm> st
Status of CHM Cluster Monitor V2.05, reading data V1.02:

  Started at:    2010-11-12 16:04:39.167
  Server time:   2008-12-06 00:28:58.000
  Running for:   8:11.568 min

Connections and average amount of data:
  i_0:localnode:./staiu13_1205.trc" reading, 75.198 KBytes (0.153 KB/sec)

Nodes:
  3:  "staiu13"  Up up count=2, down count=1
  2:  "staiu14"  Up
  1:  "staiu16"  Up
  3 nodes

The same principle applies to command parameters. For example, instead of "list alerts" use "la":

```
chm> la
List of alerts collected since start of the UI

View 0: Cluster "MyCluster", NODENAME "staiu13", Alerts: 41
  Red Alert 32  Time: 00:29:30.173--00:29:31.173 (1.000sec)
    CPU usage on node staiu13 (99.0%) is Very High (> 90%). 19 processes are waiting for only 2 CPUs.
  Red Alert 33  Time: 00:29:31.173--00:29:32.173 (1.000sec)
    CPU usage on node staiu13 (99.0%) is Very High (> 90%). 19 processes are waiting for only 2 CPUs.
...etc
```

3. Cluster View Histograms

Upon starting CHMOSG with the option "-1", it will present an alternative representation of the Cluster View. Two additional windows appear and display the **Top Resource Consumer** table and the **Cluster Stats** table in the form of scrolling histograms:
Note: The windows may appear initially overlapped. Once they have been placed in their desired position their location will be remembered upon restart.

The information displayed here is identical to that in the text-based Cluster View window. What differs, is its representation. It provides visual feedback to easily identify trends and explore interactively the short term history of events. For example, by placing the cursor in the PRIV-MEM cell, the above image shows that on node stbbg11 a process "ons" with the PID 11030 consumes a steadily increasing amount of private memory. Moving the cursor up or down that column will show whether ons is behaving the same way on the other nodes.

To examine the histogram tables and their interaction in closer detail, rows in these tables are associated with cluster nodes. Columns describe metrics, plus optionally the type of entity associated with the metric, such as a device or a process. The first of the windows, Table 1, displays the top consumers (processes) for several metrics. Every cell in these histograms is attached to its associated entity. Note the orange vertical bar following the cursor position and the information in light blue on the bottom. It shows the details about the referenced entity and its metric. These are referred to as Entity Histograms:

With every position of the cursor, the associated entity might change. In the above illustration the cursor position shows that process "oraclerac2" was using 32% of the CPU time on node strdu10 at 18:11:26. Note the data is changing every second. This is due to the cluster sample rate being 1 second in concert with the default 1 second CHMOSG refresh rate. Should the CHMOSG sample rate be greater, then that will be evident in the resulting stepped display.
A simpler form of histograms shows values only. They are referred to as **Value Histograms**. The second window displays the Cluster Stats using Value Histograms. Note that the text tables may display metrics as well as static attributes, e.g. number of CPU’s in the system. Configuration or capacity data such as this is not displayed in histogram views as it is not dynamic.

The **distance of the cursor to the right side of the histogram** is proportional to the passage of time. As the cursor is moved in a histogram cell, it displays data for the selected time where the right-most sample is for the current time. In the above example, the mouse position selected an event 33sec in the past. Several time periods are marked red as an indication of alerts reported by CHM/OS. CHMOSG shows the text of the alerts on a translucent strip placed atop of the histograms:

![Example of CHM Cluster Monitor](image)

Here CHM/OS alerted that process "ps" was using 99.9% CPU time at 18:11:50, 1 min 47 sec ago on this display. However, as soon the next set of values arrive, the alert moves away from the cursor position. In this mode it may be difficult to read the metric values and alert texts, and to analyze them. In order to support browsing and analysis of short term history, CHMOSG provides a **pin mode**.

### 3.1 Using Pin Mode to Explore Short Term History

Pressing the letter "p" while clicking on one of the histograms “pins” the vertical bar to its current position. It will no longer follow the cursor. Its color changes from orange to blue to indicate this mode of operation. Use the left/right cursor keys to shift the pinned point of time a pixel at a time to examine each data point.
In this example the pinned time is 19:02:06 when an alert was generated that reported ocssd.bin was using 81.6% CPU time.

Note that in pinned mode the cursor may move to any place on the histogram; however, CHMOSG still shows the pinned value. Furthermore, when the cursor moves to any other cell in the table, CHMOSG also shows the corresponding value at the pinned time. This allows for manually examining values around a specific point of time as long as it remains visible. That period of visible time depends upon the length of the histogram, which is function of the pixel width of the screen.

Press "f" to switch CHMOSG back to the follow mode, in which the orange colored marker always follows the cursor. CHMOSG will change back to follow mode automatically when the pinned marker bar reaches the left or right end of the histogram.

3.2 Showing Unresponsive Nodes

Periods of time in which some nodes were unresponsive are depicted by a purple background. If a node is currently unresponsive, the name of the node appears in gray with the entire row using purple in its histograms.

![Image of histogram showing unresponsive nodes]

In the above example 4 of the 5 nodes were recently unresponsive, and nodes stact38 and stadg10 are currently unresponsive.

3.3 Finding a Global Maximum of a Metric

The histogram views will automatically scale to the largest encountered value. This maximum amplitude scale can be node-maximum (default) or cluster-maximum. Depending on the metric, node values may vary significantly as displayed in the following example:
On node strdu09 the histogram of CPU Queue shows a maximum value of 289, thus the entire histogram scales to that value. However on node strdu12 such large process queues were not recorded as seen when the cursor moves down:

CPU Queue maximum for strdu12 was 14, thus the histogram scales to a much smaller range of values.

In order to see all nodes of a cluster in the same scale, the cluster-maximum mode is activated by pressing "=" or clicking the right side of the column title. As seen in the following Cluster View, a red "=" equals sign is displayed to indicate that this column now uses a common scale based upon the cluster's maximum value.
Note how the CPU Queue length varies across all of the nodes and the extent of the load on node \textit{strdu09} relative to the others. Click on the red "\text{=}" sign or the "\text{"}=\text{"}" key to switch off the cluster-maximum mode. Both modes have their uses and advantages. Node-maximum mode displays metric values in better resolution on each node, whereas cluster-maximum mode better displays node-to-node metric consistency or imbalance.

3.4 Statistics of top users.

Histograms are limited to short term history. In order to preserve a long term trace of these values, CHMOSG maintains a compact set of statistics of the values since launched. CHMOSG displays these statistics in a bar graph with an average value marker. To access this Statistics View, mouse over a cell with a node name of a histogram view (left most cell on any row) and press "\text{s.}" A window such as the one below will display:
The green bar depicts graphically the range of recorded values minimum to maximum, with an orange marker showing the average mean value. If the range of the min, average and max values is small, the green bar is not displayed.

Select a tab to change metric type. Click on the header of the column to change the sort criterion. The default sort criterion is "Sum of Averages," which a product of the number of occurrences (Count column) and the average value (Ave column.). Click on the "Nr" (number) to use the default criterion. Click on any other column header to sort the values in descending order based upon that column. For example, the following statistics view shows the same table sorted by descending maximum (Max) values:

<table>
<thead>
<tr>
<th>Nr</th>
<th>Count</th>
<th>Incar</th>
<th>Min</th>
<th>Ave</th>
<th>Max</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>165262</td>
<td>2</td>
<td>0.5</td>
<td>1.4</td>
<td>4.0</td>
<td>osysmond</td>
</tr>
<tr>
<td>2</td>
<td>60263</td>
<td>1</td>
<td>0.5</td>
<td>2.0</td>
<td>10.7</td>
<td>loggerd</td>
</tr>
<tr>
<td>3</td>
<td>21671</td>
<td>1</td>
<td>0.5</td>
<td>1.7</td>
<td>12.0</td>
<td>mdnsd</td>
</tr>
<tr>
<td>4</td>
<td>547</td>
<td>427</td>
<td>0.7</td>
<td>26.3</td>
<td>99.9</td>
<td>perl</td>
</tr>
<tr>
<td>5</td>
<td>297</td>
<td>231</td>
<td>6.3</td>
<td>48.0</td>
<td>94.6</td>
<td>mrtg</td>
</tr>
<tr>
<td>6</td>
<td>8882</td>
<td>1</td>
<td>0.5</td>
<td>0.9</td>
<td>2.0</td>
<td>ncssd</td>
</tr>
<tr>
<td>7</td>
<td>3267</td>
<td>1</td>
<td>0.5</td>
<td>2.2</td>
<td>5.9</td>
<td>rpciod</td>
</tr>
<tr>
<td>8</td>
<td>5777</td>
<td>1</td>
<td>0.4</td>
<td>1.2</td>
<td>4.9</td>
<td>ora_mmon_cr32v8</td>
</tr>
<tr>
<td>9</td>
<td>3573</td>
<td>1</td>
<td>0.5</td>
<td>1.6</td>
<td>5.8</td>
<td>emagent</td>
</tr>
<tr>
<td>10</td>
<td>3793</td>
<td>1</td>
<td>0.5</td>
<td>1.5</td>
<td>3.0</td>
<td>oraagent</td>
</tr>
<tr>
<td>11</td>
<td>472</td>
<td>367</td>
<td>0.6</td>
<td>11.6</td>
<td>80.6</td>
<td>automount</td>
</tr>
<tr>
<td>12</td>
<td>3471</td>
<td>1</td>
<td>0.5</td>
<td>0.9</td>
<td>2.0</td>
<td>cssdagent</td>
</tr>
<tr>
<td>13</td>
<td>2302</td>
<td>1</td>
<td>0.5</td>
<td>1.0</td>
<td>2.0</td>
<td>orarootagent</td>
</tr>
<tr>
<td>14</td>
<td>1091</td>
<td>891</td>
<td>0.9</td>
<td>1.7</td>
<td>16.4</td>
<td>netstat</td>
</tr>
<tr>
<td>15</td>
<td>1142</td>
<td>1</td>
<td>0.5</td>
<td>1.4</td>
<td>6.9</td>
<td>ora_cj00_cr32v8</td>
</tr>
<tr>
<td>16</td>
<td>1710</td>
<td>1</td>
<td>0.5</td>
<td>0.9</td>
<td>2.0</td>
<td>ora_dia0_cr32v8</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>4</td>
<td>10.4</td>
<td>60.0</td>
<td>99.9</td>
<td>java</td>
</tr>
<tr>
<td>18</td>
<td>1484</td>
<td>1</td>
<td>0.5</td>
<td>1.0</td>
<td>2.0</td>
<td>csssdmonitor</td>
</tr>
<tr>
<td>19</td>
<td>58</td>
<td>42</td>
<td>1.9</td>
<td>20.2</td>
<td>95.5</td>
<td>ora_j000_cr32v8</td>
</tr>
<tr>
<td>20</td>
<td>37</td>
<td>12</td>
<td>3.7</td>
<td>25.8</td>
<td>94.4</td>
<td>rsnq</td>
</tr>
<tr>
<td>21</td>
<td>854</td>
<td>1</td>
<td>0.5</td>
<td>1.0</td>
<td>2.0</td>
<td>rwhod</td>
</tr>
<tr>
<td>22</td>
<td>805</td>
<td>1</td>
<td>0.5</td>
<td>1.0</td>
<td>2.0</td>
<td>ohasd</td>
</tr>
<tr>
<td>23</td>
<td>101</td>
<td>80</td>
<td>0.8</td>
<td>7.4</td>
<td>38.3</td>
<td>ora_w000_cr32v8</td>
</tr>
<tr>
<td>24</td>
<td>718</td>
<td>1</td>
<td>0.5</td>
<td>0.9</td>
<td>2.0</td>
<td>init.cssd</td>
</tr>
</tbody>
</table>
The following is the same table sorted by the average value (Ave or orange marker):
Note, that these statistics have a somewhat limited scope as they contain only the top consumers in each metric per sample. However, over a significant time period these statistics can provide valuable information to identify which processes are consistently the top resource consumers.

4. Node and Device Views

In addition to the various Cluster Views, CHMOSG can retrieve from ologgerd more detailed information about nodes and devices in the form of a **Node View** or a **Device View**. A Node View is retrieved by entering at the chm> prompt:

```
chm> node <nodename>
```

As in the following example:

```
chm> node adc2110712
...Request Node View 'adc2110712' from loggerd.
chm>
```

Alternatively, double-clicking a cell containing a node name will retrieve a Node View. Generally, if the mouse cursor changes to a finger-pointing hand when hovering over a cell a more detailed view is available as displayed below:

```
In this table, the cursor turned into a hand over a cell containing a node name. Upon double-clicking, a node view will be retrieved. This view presents the detailed statistics on processes, disks and NICs of interest based upon predefined heuristics. The cursor may
```
also indicate available views for certain devices such as disks and network interconnects (NICs). The content of all views is determined by ologgerd and may vary by CHM/OS release as well as O/S, platform or version.

The following is an example of a node view on Linux:

As in Cluster Views, Device and Node Views display text alerts in the lower scrollable text area. These alerts are generated when the sampled value of a resource metric either goes above or falls below a threshold that could lead to potential problems on the node and hence on the cluster.

NOTE: Alerts are generated by ologgerd and may vary in scope or existence based upon CHM/OS version.

Double-clicking the disk name will retrieve further details about its partitions. The information is presented in the Device View as displayed below. This Device View provides a detailed list of partitions and corresponding stats for each one. It also clearly marks which partitions which belong to certain categories such as Voting, OCR, SWAP, and ASM disks:
Network device views provide useful communication statistics such as Reads (KB/s), Writes (KB/s), Errors, Packets In/Out and Latency:

5. Replay session (Historical Mode)

Since all of the metric data that CHMOSG displays is retrieved from a database in the cluster, a range of historical data is available based upon the configured retention time. Invoking the CHMOSG with `-d` option starts it in Historical Mode.

```
$ chmosg -d  "<hh>:<mm>:<ss>" -m <nodename>
$ chmosg -d  "<mm>" -m <nodename>
```

where `-d` is used to specify hours `<hh>`, minutes `<mm>` and seconds `<ss>` in the past from the current time. For example, `chmosg -d "05:10:00"` starts CHMOSG and displays information from the database which is 5 hours and 10 minutes in the past from the current time.

Once started, CHMOSG retrieves the data as if it was in real time. There is no facility to skip to an earlier time or fast forward without restarting from those points.

**Appendix A. Customizing Java run time system.**

Java Virtual Machine (JVM) is the runtime process which interprets user Java classes. All contemporary JVM's incorporate the Hot Spot Technology, which translates Java byte code into a native code on the fly, and so except for beginning invocations of these classes in most cases Java methods run in native code. Consequently they perform at speeds comparable to programs written in native languages, like C. CHMOSG is written in Java, that gives it platform independence and different modes of operation. The script `"chmosg"` invokes a JVM. Knowledgeable users may consider customizing this script, or use environment variable `_JAVA_OPTIONS` to influence the way the JVM executes code.
CHMOSG defaults to an initial memory size of 15MB (-ms15m) and a maximum memory heap size of 250MB (-mx250m).

On many platforms Java supports native 2D and 3D graphics, thereby significantly improving display performance. It is highly recommended that should OpenGL support be available to Java, it be configured for default use.

**NOTE:** Performance of CHMOSG interactive operations are the most greatly impacted by the lack of Java OpenGL support.

Information about the OpenGL library is available at: [http://www.opengl.org/](http://www.opengl.org/). Most manufacturers of rendering hardware, i.e. graphics cards, provide a version of this library for their video cards. It is important to obtain a current version of this library aside of the current drivers for the graphics card. See the following site for detailed information about Java rendering using OpenGL:

[http://docs.oracle.com/javase/1.5.0/docs/guide/2d/new_features.html](http://docs.oracle.com/javase/1.5.0/docs/guide/2d/new_features.html).

**JVM activation flags for OpenGL are:**

- `-Dsun.java2d.opengl=true` : use the OpenGL pipeline
- `-Dsun.java2d.d3d=true` : for Windows, use the Direct3D accelerator

CHMOSG does not render in 3D, but benefits greatly from the accelerated region repaint available through Direct3D.

**JVM options** may be used with the "java" command, or declared as an environment variable, `_JAVA_OPTIONS`.

**Linux Example:**

```
_JAVA_OPTIONS="-Dsun.java2d.opengl=True -Dsun.java2d.d3d=true"
export _JAVA_OPTIONS
```

Note the capitol "T" in the "true": If written with a capitol letter, Java will print to the standard output whether the OpenGL pipeline is available or not. The following is an example of a warning that OpenGL is not available:

```
java oracle.evc.toprac.run -i -mMaster
Picked up _JAVA_OPTIONS: -Dsun.java2d.opengl=True -Dsun.java2d.d3d=true
Cluster Health Monitor V2.05
Could not enable OpenGL pipeline for default config on screen 0
```

In this case, the system does not have OpenGL-supporting hardware (e.g. desktop with on-board graphics chip). The following is an example output of a system with a graphics card supporting OpenGL:

```
java -Xmx256m oracle.evc.toprac.run -i -1
OpenGL pipeline enabled for default config on screen 0
Cluster Health Monitor V2.05
```

In the case of Nvidia GeForce graphics cards, OpenGL requires the option:

```
sun.java2d.opengl.fboobject=false.
```
See section "3.1.5.5 Diagnosing Rendering and Performance Issues" at:

http://www.oracle.com/technetwork/java/javase/index-142560.html

This link is the current and comprehensive description of potential issues with OpenGL and Java 2D drawing package in conjunction with specific hardware/driver versions.

**Appendix B: Choosing cursor scheme on Windows.**

Java uses several cursor shapes, which will be mapped to the native cursors on each platform. Unfortunately on Windows, the default cross-hair cursor is represented by single stroke, and may be poorly visible on some backgrounds as in the following example:

![Cross-hair cursor example](image1)

The visibility can be improved by selecting the "Windows Black system scheme" scheme from the Control Panel:

![Mouse Properties](image2)

With this scheme, the cross-hair cursor is visibly enhanced with 3 lines and a white outline as displayed below:

![Cross-hair cursor with enhanced visibility](image3)

Copyright ©2008, 2012 Oracle and/or its affiliates. All rights reserved. Legal Notices