CHA Graphical User Interface (CHAG)

(Revision 0.8   2017-DEC-14)

> **NOTE: CHA Graphical User Interface (short: CHAG) is not a part of any Oracle Grid Infrastructure release. It ships separately on MOS. Please review the included README for required O/S level and Grid Infrastructure versions. Beginning with the 12.2.0.1 release of the Oracle Grid Infrastructure, CHAG must either be installed and run on a local node of the cluster to be monitored or used with an exported .mdb file on any other Java 8 supported environment. It MUST NOT be installed in an Oracle Home. CHAG is not supported to run remotely as the process names and associated metrics are not encrypted and thus would be transmitted on the public network.**

CHA is an acronym for *Cluster Health Advisor*.

CHA Graphical User Interface (CHAG) is a standalone, interactive, real-time capable front-end to CHA. CHAG requires a very small foot-print on the monitored system.

CHAG is invoked using the 'chag' script available in the bin directory of the CHA Home. CHAG is designed primarily for Cluster or Database experts. It is capable of handling large amounts of data efficiently. Its layout and mode of operation is designed for functional efficiency. Most of the operations can be executed using a positional pointer and hot-keys, or a floating menu available at the cursor position.

CHAG communicates directly with the Grid Infrastructure Management Repository (GIMR) using a JDBC connection. Data is provided in a form of CLOB's ("Character Large Object".) Since a CLOB contains readable text, the same data may be stored in a regular text file, which can be used to operate CHAG locally without any connection to the GIMR. A series of intuitive displays with status timelines and graphs allow browsing through time across all cluster components, see values of their metrics, analyze short term history, observe trends and view alerts.

0. CHAG Installation

The installation of CHAG is performed by unzipping into a selected directory not within an Oracle Home on one of the local cluster nodes monitored by CHA. CHAG may be installed on a non-monitored node or laptop with Java 1.6+ support. In that case it can only run by using exported .mdb files from a monitor cluster. The directory structure is the following:

```
./chag
    doc          (documentation directory)
    log          (log directory)
    jlib         (Java libraries directory)
    chag         (Bash shell executable)
    chag.bat     (Windows executable)
    LICENSE.txt  (Oracle License)
    README.txt   (Version-specific information)
```

0.1 Monitored Cluster Node Installation

Select a directory not in an Oracle Home to unzip the CHAG archive.

Open a bash shell.

Set the `ORACLE_HOME` environment variable to the Grid Home path.

Run `chag -?` from its local directory to view the list of command options.

0.2 Non-Monitored Node or Laptop Installation

Select a directory not in an Oracle Home to unzip the CHAG archive.

Open a bash shell.

Run `chag -f <mdb file name>` from its local directory to view the list of command options.

1. CHAG Command line options

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

```
     CHA UI version:     V1.0.000
     Java version:       1.8.160 on Linux

  Usage: chag [flags] [parameters]
   -h         print this help
   -?         print this help

   -f <name>       read from file, ".mdb" assumed if no suffix given
   -i              interactive with cmd prompt (recommended)
   -p <sec>        obsolete, use -q p<sec>
```

```
-q <value>[,<value>]  sets SQL-query parameters, not active with -f file input
     <minutes>: query time period in minutes. Default: 60min,  Minimum 2min
     clob:      use sql.Clob
     p<sec>:    postpone query in seconds. Default: sampling period in seconds,  Minimum 1sec

-D <item>[,<item>]  sets internal debug mode
     feed:      display timings of all data queries
     db:        activate 'alter session set event 10046'
     input:     copy input data (CLOB) to a log file
     probes:    use internal probe names
     rootcause: inform about start and stop of any problem
     topology:  show changes in set of entities ('topology')
     unit:      show warnings about implicit settings for units of measure

     (use a comma separated list for option parameters, no spaces)
```

By default, CHAG connects to the GIMR using Oracle Wallet credentials and retrieves the last 60 minutes of data. Option available to configure the connection to the GIMR is "`-q`".

Option "`-q <minutes>`" sets the amount of data based upon time for the initial query. As the CHA sampling rate is 5 seconds, in the default display case of 60 minutes, there will be 720 data points in the CLOB. This option may be used to create smaller or larger data displays. Note that when longer times are selected via `-q` option, a substantial time might be added to the startup process, especially when the monitored configuration has many nodes and databases. The current maximum number of minutes is determined by the width of the screen (number of pixels) divided by the number of samples per minute, which is 12 for the current 5 second sample interval.

Option "`-q clob`" directs CHAG to use alternative path of retrieving CLOB from the database. In some versions of Oracle Database a direct retrieving of CLOB from sql-query leads to fragmentation in the database. When user enters this option "`-qc`", CHAG uses a 2-step process to obtain CLOB's, with an explicit disposal command. Elapsed time for every query will increase.

Option "`-q p<seconds>`" adjusts the delay between the "*time in the query*" versus "*time of the query*". Default delay is one sampling period, 5 sec. For example '`-qp10`' would cause that a query for 5sec of data in the period of 10:20--10:25 will be invoked at 12:35 or later. Use this option when you observe that CHA too slow, cannot commit transactions on time for CHAG and random gaps in data are indicated. On a fast system even '`-qp1`' can be used without any adverse effects.

Use the `-i` option to activate the interactive mode. This option permits entering additional commands that are not available from the GUI.

```
$ chag -i
```

When started with this option, a "`cha> `" command line prompt appears in the OS terminal. This can be used to enter terminal commands not available on the graphical panels. These commands are enumerated by entering “`help`” at the prompt and are explained in section 5.

Option `-Dinput` can be used to obtain a complete copy of the data received by CHAG, stored in a file, called "mdb-file," after Management Database. This file can be used as an argument with the `'-f <file>'` option.

On Windows all comma-separated arguments must be enclosed in quotes. For example: `"-Dprobes,input"`, or shorter "`-Dp,i`".

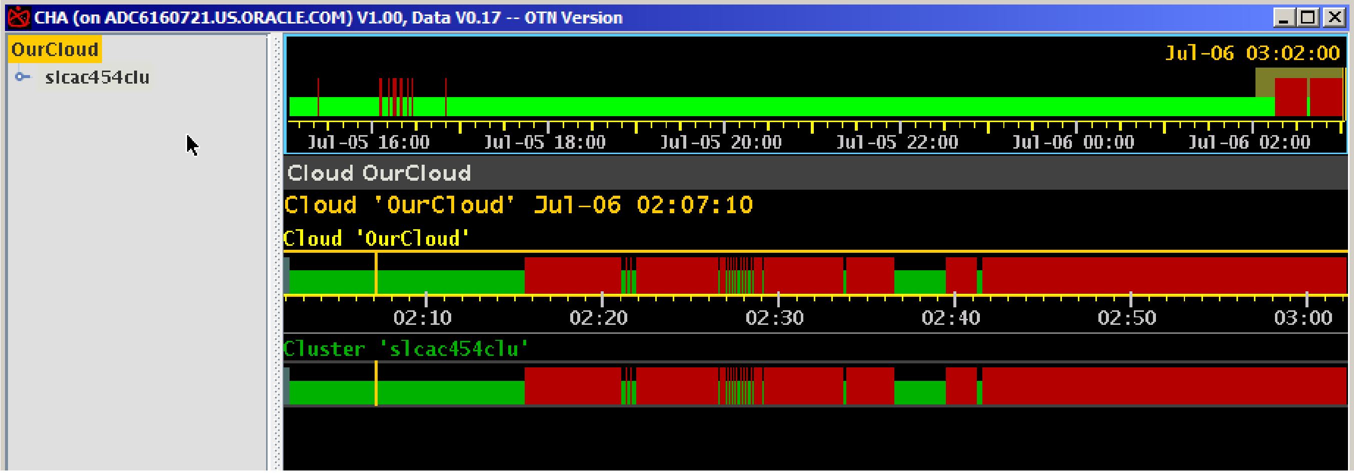
2. Cluster View: Connecting and Basics of Monitoring

Start CHAG using the script '`chag`' on Linux/Unix or '`chag.bat`' on Windows. This script is located in the chag directory. No additional parameters are required if CHAG is installed one of the cluster nodes being monitored. Connection parameters to the Grid Infrastructure Management Database (GIMR) will be obtained using Oracle Wallet services, and a connection to the GIMR will be established.

Once started and connected to a cluster node, a top-level Cloud View is displayed, shown below in section 2.1. Upon closing of the main window, CHAG will terminate.

2.1 Basics of Navigation through Entity Panels

This is the default window which will appear after the initial start of CHAG:



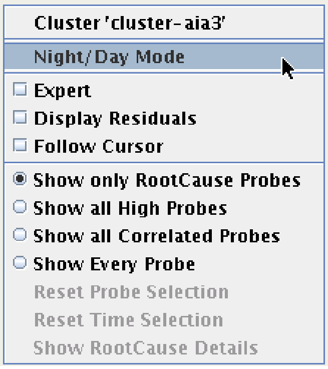
The window contains a *Navigation Tree* on the left side, and an *Analysis Panel* on the right side. The Navigation Tree always starts in a symbolic top-level **system 'OurCloud'** and shows all components within its connected Cloud. CHAG currently connects to only one GIMR, and thus contains data for all monitored nodes and RAC databases within a single cluster.

The *Analysis Panel* contains 3 components:

1. A *System Timeline* atop, showing a condensed long term system history,
2. A *Status Line*, and below
3. An *Entity Panel*, associated with the Entity selected in the Navigation Tree.

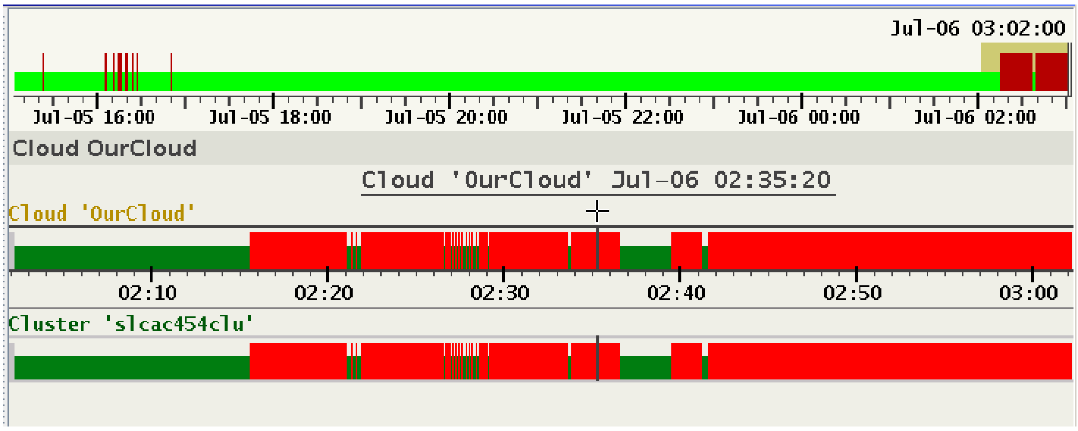
The currently selected Navigation Tree Entity is marked with a yellow background.

The right side Analysis Panel initially appears with a dark background called a *Night Theme*. A *Day Theme* displaying dark text on a light background is also available. The theme can be toggled by moving cursor onto the Entity Panel and pressing lowercase `'w'`, or use a right mouse click to access a floating menu and select "Night/Day Mode":



This is the first example of how to use either a *Hot-Key*, or a *Floating Menu* to perform the same function. A complete list of Hot-Keys can be found in Appendix A.

The main panel in *Day Theme* looks like this:

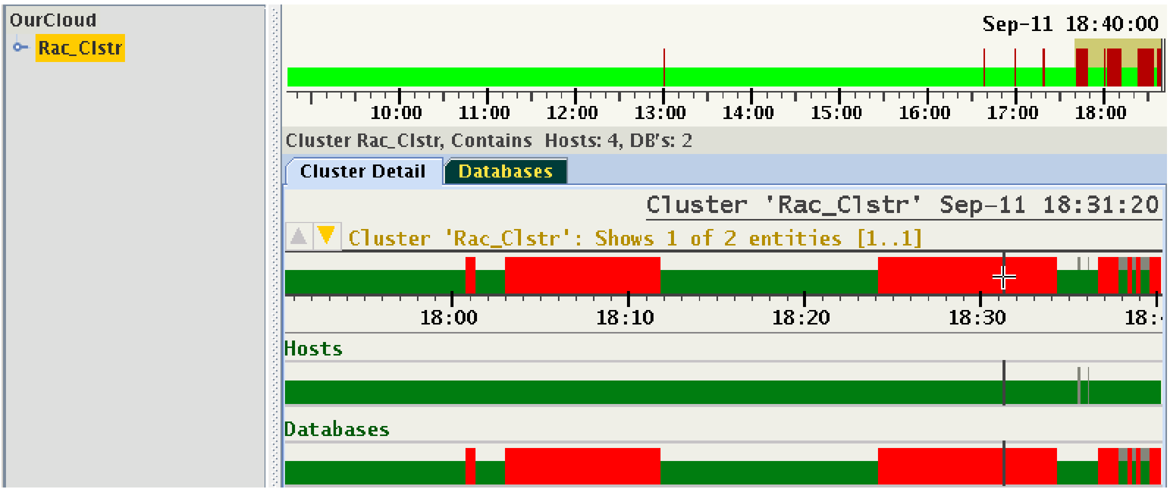


This change is *persistent*. With every consecutive start the most recent selected theme is used. The additional settings that CHAG persists are listed in Appendix B. One such stetting is the size and position of the window. Note: The persistent storage for these settings is associated with the *specific user* on a *specific host*. These settings need to be repeated when using CHAG on multiple hosts.

The status timelines in the Entity Panel display for each time-stamp the state of its entity, marked either green, red or gray.

1. Green means that at that point of time none of its sub-entities indicate any problem flagged as "abnormal".
2. Red means that at least one of its sub-entities is in abnormal state. Mathematically this means that CHAG calculates a union of sets of all sub-entities and propagates its value upward to the top most entity, which is here OurCloud.
3. Gray marks periods of time in which no data was received for any of the sub-entities.
4. Gray atop over green marks periods of time in which some sub-entities received no data.

Click on the cluster entity under OurCloud. CHAG now marks the cluster name by a yellow background, and the right side the Entity Panel changes to display the panel of the cluster. Note also that when a cluster entity is selected, the status line under the Timeline shows the number of hosts and databases in this cluster.



Cluster Rac_Clstr, Contains Hosts: 4, DB's: 2

Cluster 'Rac_Clstr' Sep-11 18:31:20

Cluster 'Rac_Clstr': Shows 1 of 2 entities [1..1]

In this hierarchy the cluster entity contains two sub-entities "Hosts" and "Databases". The above display indicates that all red-marked abnormal states come from one of the databases only.

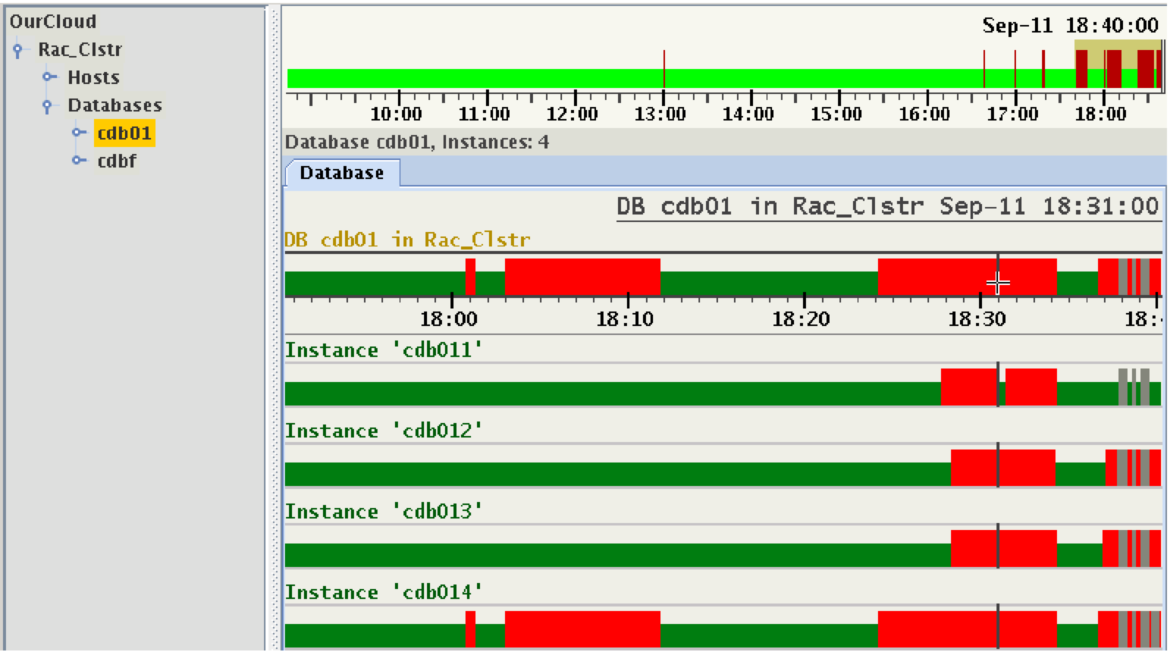
Any entity in the tree may be expanded to show its sub-entities. Note that *expanding* of the tree does not change the Entity Panel on the right side. This Panel changes only when a different Entity is *selected* in the tree.

Note the position of the cursor on the status timeline of the cluster. The entity name will change color depending on the 'Night' or 'Day' theme being used to indicate that it has focus. Moving the cursor over the 'Databases' status timeline switches the highlighting and focus accordingly. Focusing on a specific entity allows to enter its panel without moving mouse pointer to the navigation tree. In order to step down to the panel of any sub-entity, like 'Databases' in the above image, user can chose between these methods:

1. Select the Entity in the navigation tree on the left side (what requires a movement of the mouse pointer to the tree);
2. Place the cursor on the Entity name and double click on it;
3. Use a hot-key without moving the mouse pointer: Press `<Enter>` and CHAG will select the focused Entity and enter its panel. The selected Entity will be automatically expanded and highlighted in the navigation tree;

This is the 2nd example of how to use hot-keys: Instead of moving the pointer across distances to expand an entity in the navigation tree, focus the entity using one hand, and press `<enter>` using another hand. This is the by far fastest way to traverse through the entities.

In this example Entity 'Databases' contain two databases, of which one indicates abnormal state. When we select this DB and step down to its panel, we will see timelines of all its instances:



Database cdb01, Instances: 4

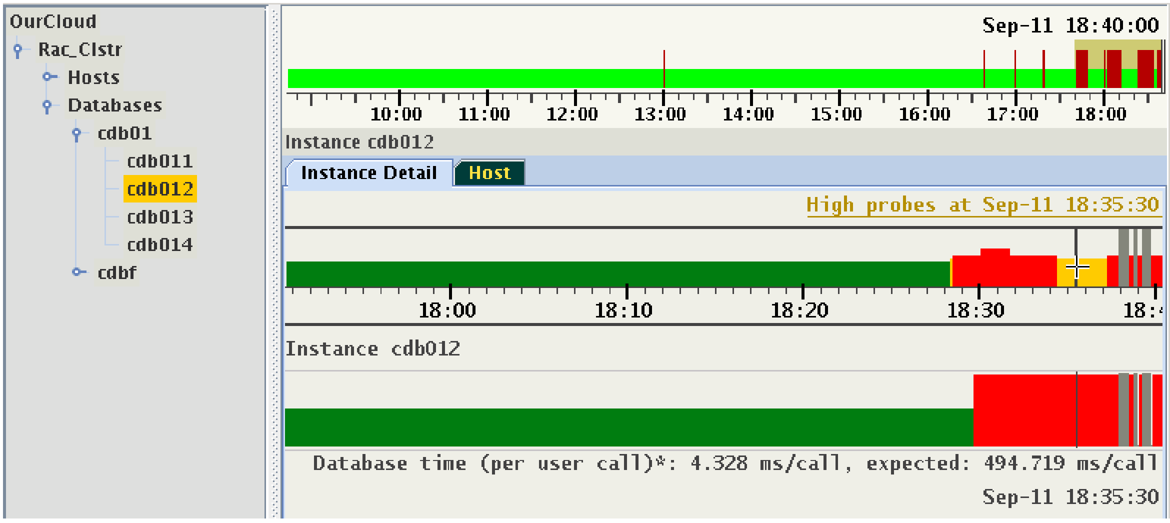
DB cdb01 in Rac_Clstr Sep-11 18:31:00

From this display we can clearly see that the state of the DB is in fact a *union of states* of its instances. On this display user can also notice the *gray marked areas* in status bars of instances. They indicate lack of monitored values for some periods of time. We do not know the exact reason why the values were not reported in these times: The entity might have been stopped, terminated, evicted or maybe merely non-responsive. The only known fact is that not a single value was recorded for this entity at this point of time.

With the 'Instances' we reached one of the final Sub-Entities. These are the ***Target Entities*** of CHA. Every kind of a Target Entity has its own specialized panel in CHAG.

2.2 Target Entities.

CHA currently supports two kinds of target entities: *Host* and *Instance*. Using the above example, lets enter the panel of the second from the top instance `cdb012`:



High probes at Sep-11 18:35:30

Database time (per user call)*: 4.328 ms/call, expected: 494.719 ms/call

Sep-11 18:35:30

On the panel of an Instance (or a Host) the top timeline represents the state of the target entity. Any status timeline below represents a time series of individual metrics collected for the target entity, which are refered to as *signals* or *probes*. The state of a target entity is represented by 4 colors:

1. **Green**: The state is normal, none of the probes are in abnormal condition.
2. **Yellow**: Entity state is still normal, but some of the probes are in abnormal condition.
3. **Red**: Entity is in state abnormal. At least one Problem (also called "Decision") indicates the reason for the abnormal state. The height of the red bar changes with the number of Problems at that point of time.
4. **Gray**: Data was not received.

Every single point in the timeline of a target entity holds a set of values:

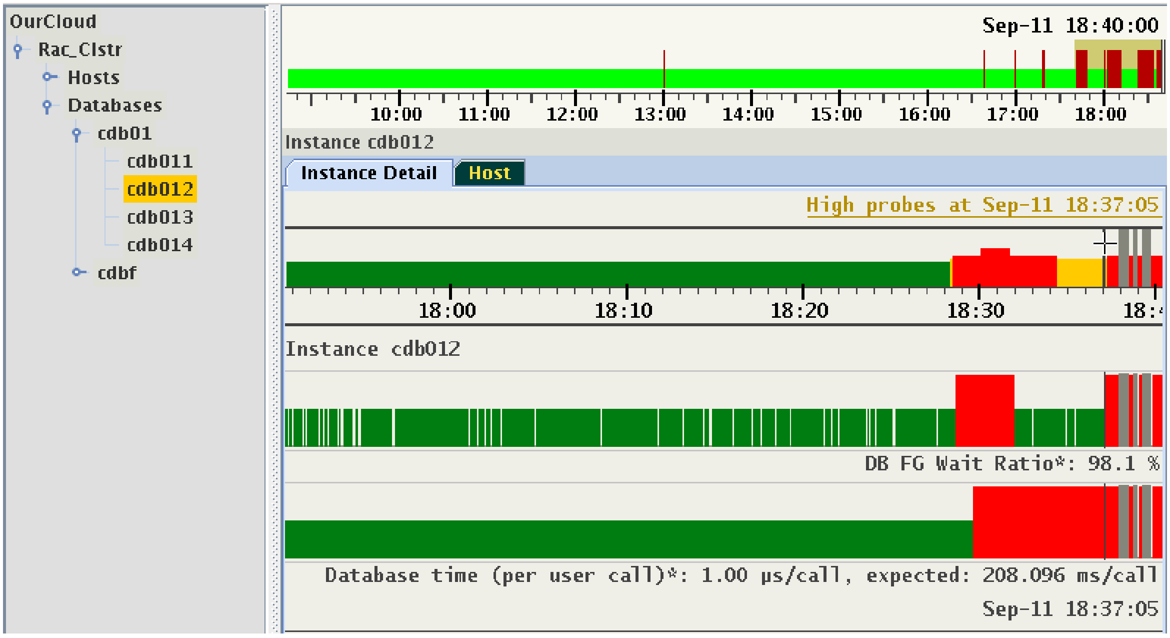
1. State of the Entity, which may be Green, Yellow, Red or Gray (missing), and
2. Set of Problems (if indicating Red state,) or
3. Set of Probes in abnormal state (if indicating Yellow state).

The set of indicated probes may change with every time-point. By default only probes which are in an abnormal state are shown. When there is no problem indicated, the target entity will be yellow on the timeline. There are a collection of options controlling selection of displayed probes that will be described later.

In the above illustration at the time 18:35:30 the state of the instance is indicated by a yellow bar, and below we see that the probe "`Database time`" is in state abnormal. Its value at this point of time is 4.328 ms/call, albeit the predicted (or expected) value in the active calibrated model was 494.719 ms/call. The the set of the probes in abnormal state and their value may change in every individual point of time. When a probe value is reported in a sample, the data set contains:

1. State of the probe (Normal/High/Low),
2. Observed Value, which may be missing at some points of time,
3. Predicted Value, which may also be missing at *some* points of time, or when no prediction for this probe is available, this value is missing at *all* points of time.

Not every probe is reported in every sample. Some probes might be reported only at certain points of time. A typical example of such probes are DB wait events. When a probe value is not reported, an empty space is displayed in the timeline denoted as gray. Below is another example for the same Target, showing a sample at a different point of time. At this point of time 2 probes are indicated as abnormal:



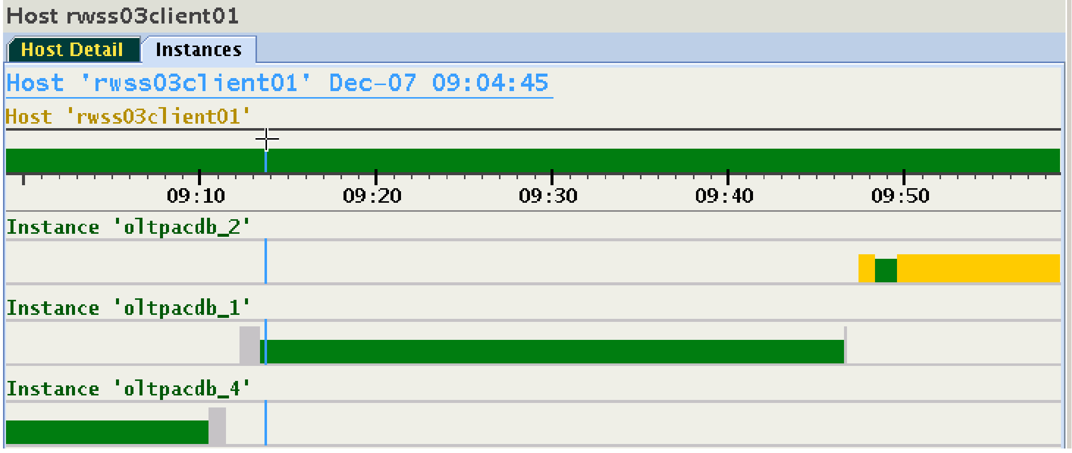
Notice the numerous gaps in the indication of "`DB FG Wait Ratio`": This is an example of a probe without expected values, and whose observed value is not being reported with every sample. This differs from the indication in time-stamps in which no any values were reported for an Entity. These periods of time are marked by the gray bars.

Every kind of a Target Entity has its own custom made panel with specialized tabs. For example, the panel of an instance has a tab "*Host*", which shows the host where the instance is running. Similarly, a panel of a host has a tab called "*Instances*", which shows a set of instances running there.

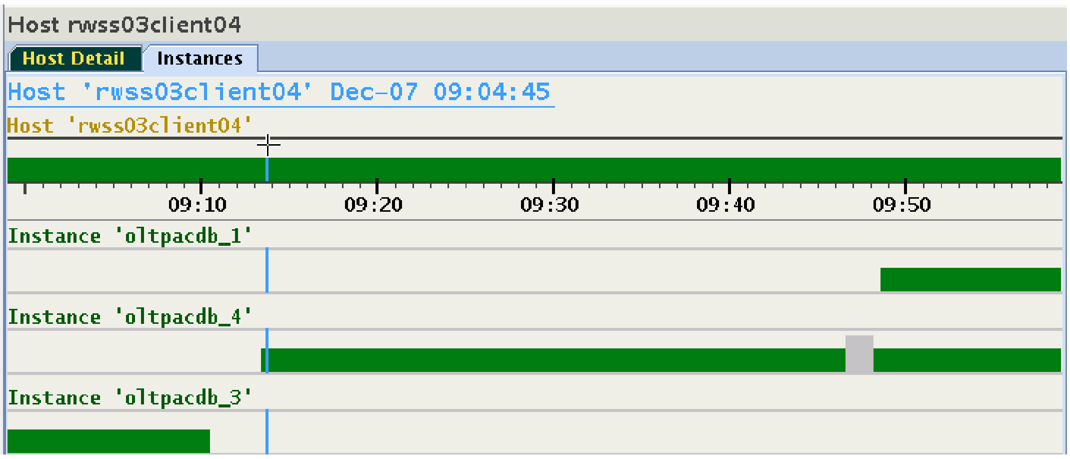
Generally, this panel is used to visualize *dynamic dependencies* between target entities. The navigation tree depicts only the *hierarchy of the logical structure* of the installation. The logical structure contains a mix of hardware and software target components in their hierarchical dependency. For example, an entity "Cluster" points to "Hosts", which contains a list of individual hosts. These relationships does not change with time. The dynamic dependencies describe how the target entities relate to each other. These relations may change with time. They are defined by pairs of target entities and ranges of time in which they are valid. Some of these relations may be exclusive. For example, at a specific point of time an Instance is said to run on one Host; however a Host can have a number of Instances running on it at the same point of time.

2.2.1 Panel of a Host

A panel of a host has two tabs: "Host Detail" (see section 2.2) and "Instances." An example:



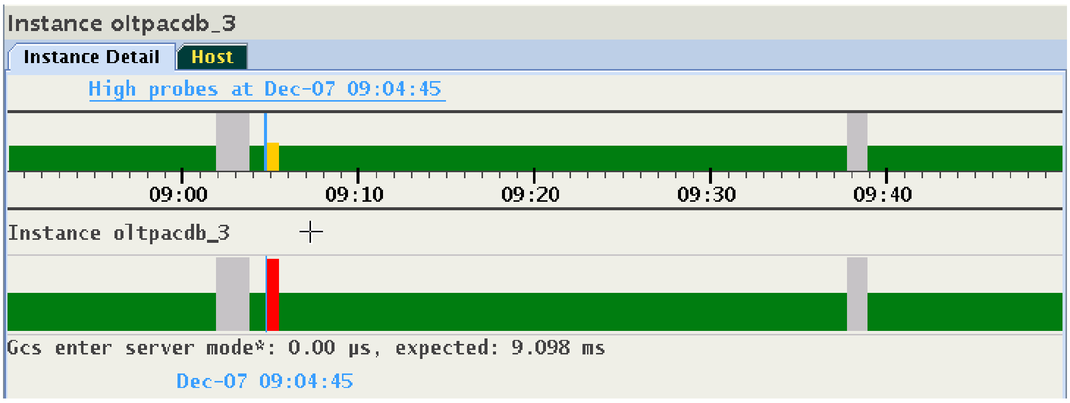
This host is running 3 different instances between 9 and 10 o'clock. Something happened around 9:11 and 9:46, which caused the instance to shut down, or change its host. As more than one instance from different databases may run on a host, there is a 1:N relationship. After approx. 9:48 two instances are running on this host:



In order to switch from a Host panel to the desired Instance panel, move focus to any of the instances using `<Down/Up Arrow>` and press `<Enter>`, or double-click on any of the instance names. The display will switch to the panel of the chosen instance, select its node in the navigation tree, and expand the navigation tree to make this node visible.

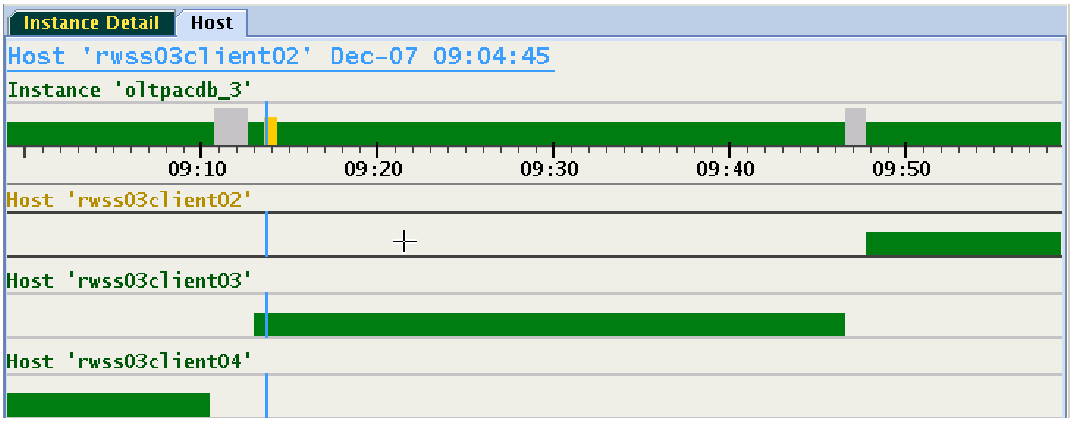
2.2.2 Panel of an Instance

From the above example lets switch to the instance `otlpacdb_3`. It is the customary status panel, discussed in section 2.2:

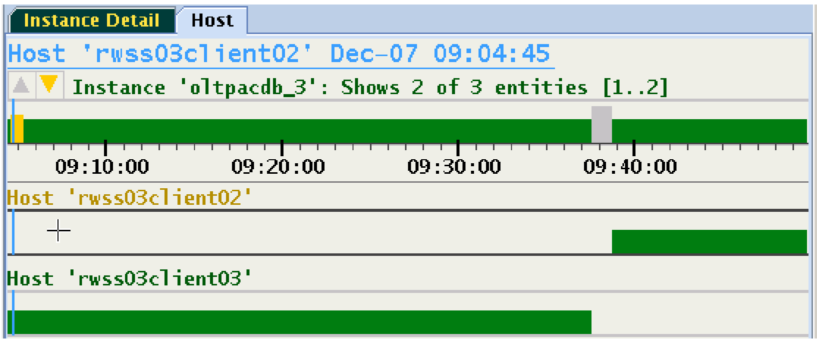


Gcs enter server mode*: 0.00 µs, expected: 9.098 ms
Dec-07 09:04:45

The panel shows the two interruptions in service, mentioned in the previous section. This display does not indicate that the instance was in fact running on more than one host at different points of time. Click on the "Host" tab to visualize where the instance was running. The relationship is 1:1, an instance runs always on only one host at a particular point of time:



From this display switching to any of the hosts is performed by either double-clicking on its name, or by selecting the timeline and pressing `<Enter>`. Using this method user can quickly switch between host and instances. The diagram on this panel dispalys a form of a cascaded time ranges from most recent on top to the oldest on the bottom. When resizing the window, the display may change dynamically causing timelines to be added or removed. For example:



The run time of the instance on host `rwss03client04` is no longer indicated, because it ended prior to 9:04 and this time is not showing in the narrower window.

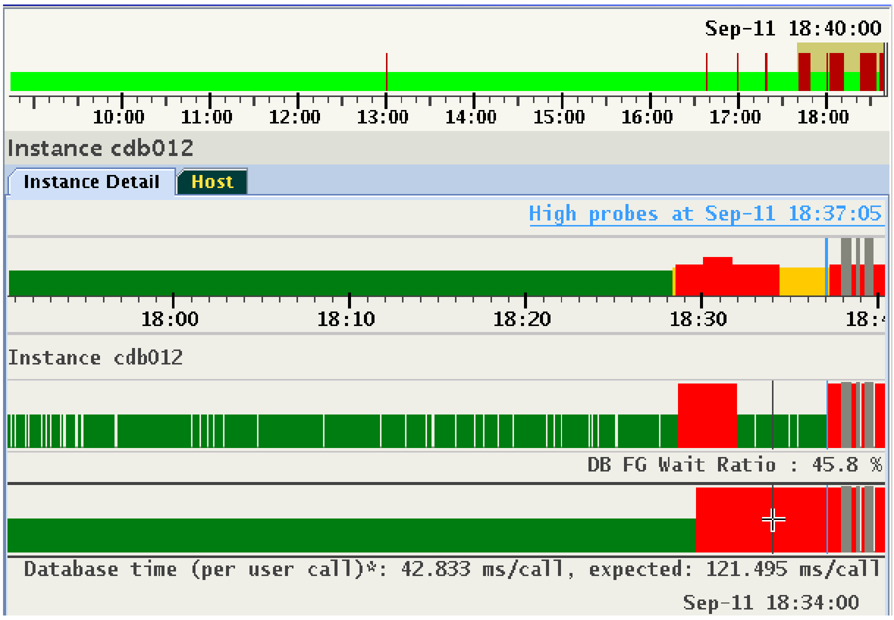
2.3 Browsing Through Time and Pin Operation.

In a live-session the display is dynamic. receiving new samples and advancing all timelines. With every new sample or with a change of a cursor position the corresponding time point is automatically displayed with a set of abnormal probes together with their values. This time point is indicated by a vertical marker. Once a time point of interest is identified, it can be "pinned" using

1. Double click on any time at the target status timeline,
2. Press hotkey `"p"` while hovering over target status timeline,
3. Press `<Left/Right Arrow>` while hovering over target status timeline.

The pinned time point may be fine-tuned by pressing the left or right arrow keys and observing the displayed timestamp. Once the time point is pinned, the position marker will no longer follow the cursor. Its color changes from orange to blue to indicate this mode of operation. Moving the cursor to any other time and pressing `"p"` changes the pin point to the location. Pressing `"f"` will unpin the time point causing it to again follow the cursor.

Note that pinning to a specific time point selects its corresponding *set of probes* in abnormal state. When browsing through the visible values of this set, the pin point can be preserved. To achieve this, either hover cursor over the timelines of the probes, or use `<Down Arrow>` to move focus to these timelines. Press and hold `<Control>` and move time stamp marker sideward using mouse or Left/Right Arrow keys. Aside of the light-blue pin marker, a second marker is displayed following the cursor position. The bottom information line shows the time point associated with this marker. The floating label at the bottom of each bar graph shows the values at the position of the cursor.



DB FG Wait Ratio : 45.8 %

Database time (per user call)*: 42.833 ms/call, expected: 121.495 ms/call

Sep-11 18:34:00

Image shows a set of 2 probes in abnormal state at pinned time 18:37:05. Depress `<Control>` and move cursor along the timeline. Each position corresponds to a different point of time. In this example, the values sampled at 18:34:00 of a set of probes selected at 18:37:05 are shown. With `<Control>` remaining depressed, use `<Left/Right Arrow>` to step through time points one sample at a time.

To activate this feature permanently select checkbox "Follow Cursor" in the floating menu.

Pin point is automatically preserved in every Entity sharing the same cluster. This feature is used to move between different panels and explore values of probes at the same time point. When viewing live data the pin will be released when its timestamp reaches the left or right end of the viewport.

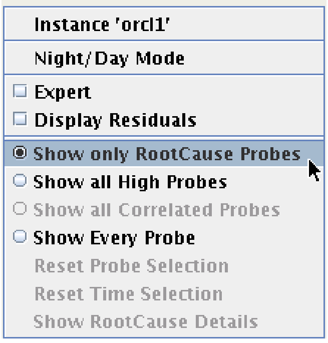
2.4 Changing set of Visible Probes

By default, only probes in abnormal state are displayed in yellow for every time point. When the selected time point is marked red, a set of Problems is active (see section 2.6), each with its own set of abnormal probes. At this point only the description of the Problem is displayed. Once a Problem is selected, its associated set of abnormal probes are identified and displayed. This will be discussed further in section 2.6: Problems.

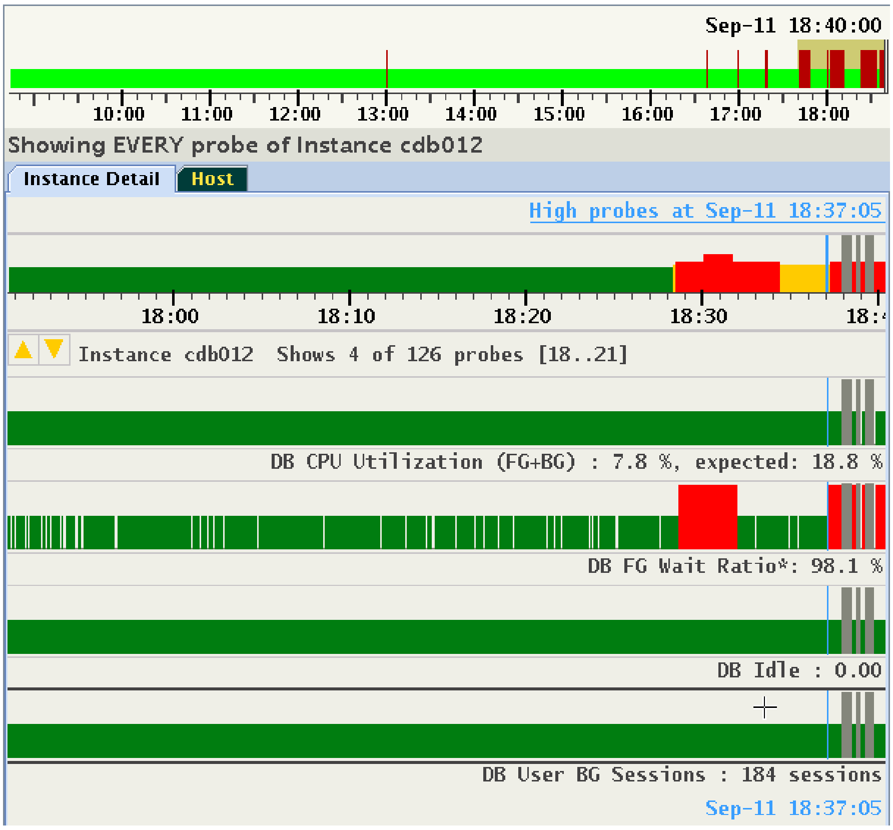
Some users may prefer to display all high probes at any time point, or even would like to explore visually time-series *of every probe,* regardless their state. The following care the custom settings for selection of probes:

1. Display only abnormal probes (the "yellow" marked zone) or probes of a selected Problem. This is the default.
2. Display always all abnormal probes (on "yellow" or "red").
3. Display every existing probe regardless its state.
4. Display all probes belonging to the same category (called "*correlated probes*").
5. Display every abnormal probe in a specific time range (see section 2.5).
6. Display only subset of all existing probes (3) by their category (see section 3.3).

The following examples show how to select options (2), (3) and (4) from the above list. Use either floating menu or a hot-key to change the mode in which probes are being displayed. For example:



Either select "*Show all High Probes*" or press `"a"` to display every high probe at any point of time. Select "*Show Every Probe*" or press `"A"` to view every probe regardless its state:



Showing EVERY probe of Instance cdb012

Instance cdb012 Shows 4 of 126 probes [18..21]

DB CPU Utilization (FG+BG) : 7.8 %, expected: 18.8 %

DB FG Wait Ratio*: 98.1 %

DB Idle : 0.00

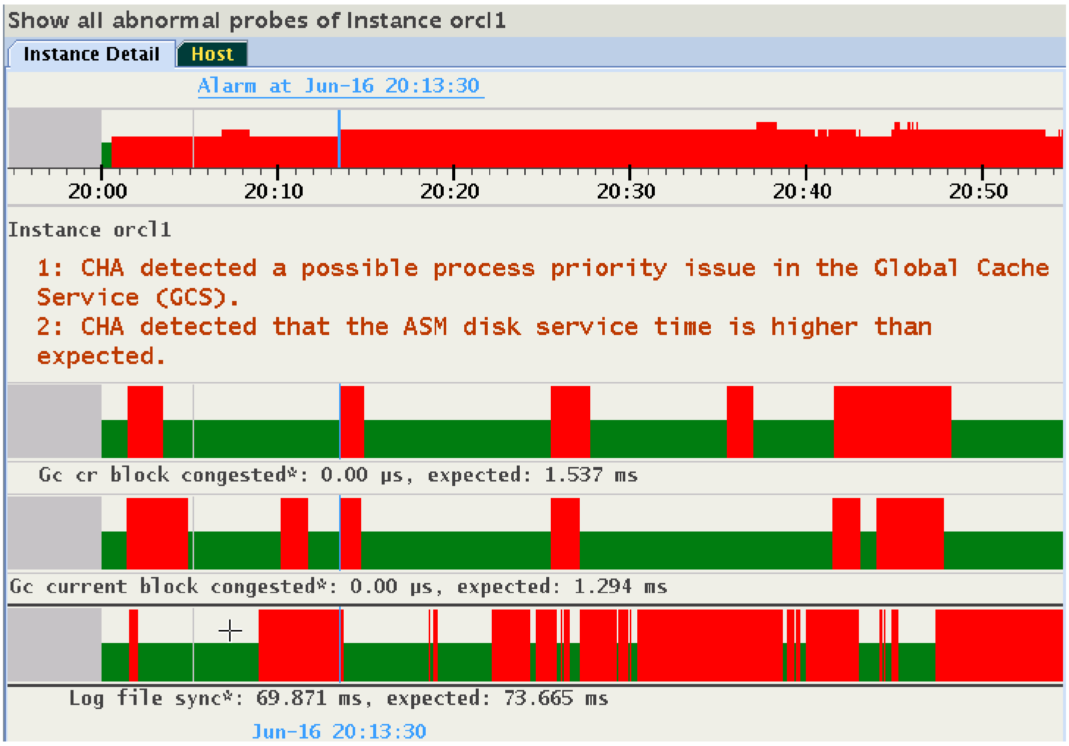
DB User BG Sessions : 184 sessions

Sep-11 18:37:05

The number of probes may exceed the size of the display window. In this example only 4 probes out of a total of 126 can be displayed. The Status label under the time line indicates which subset of probes is being shown and provides Up/Down arrows to navigate. In this example, probes 18, 19, 20 and 21 are displayed. Both Up/Down icons are highlighted indicating that scrolling both directions is available using *Cursor Up/Down* or *Page Up/Down* keys or using mouse wheel.

It is also possible to display all probes belonging to the same category. Categories of probes will be explained in the chapter devoted to Expert Mode, section 3.3. Every probe may belong to one or more pre-defined categories. For example "Buffer Cache," or "Global Cache Exceptions" are categories of probes associated with database instances.

Note that when cursor hovers over any probe's timeline, its boundaries are highlighted as it gets focus. Alternatively, use `<Up/Down Arrow>` keys to move focus to a specific probe, without moving the mouse pointer. In the example below focus is on "`Log file sync`":



Instance orcl1

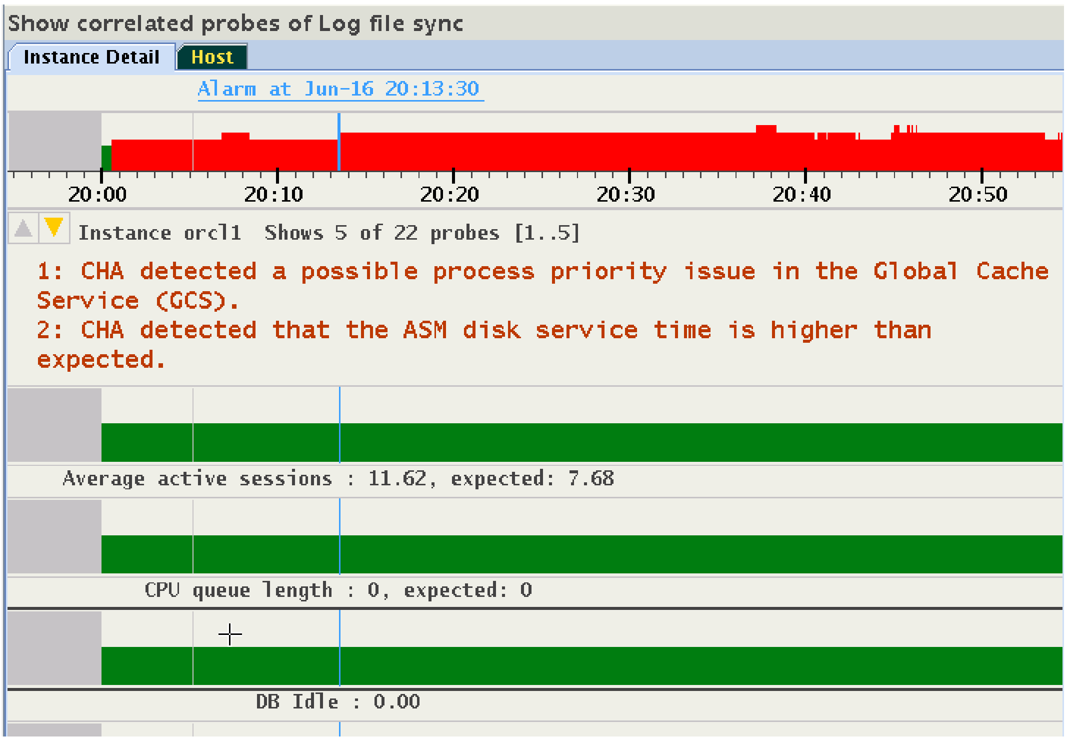
1: CHA detected a possible process priority issue in the Global Cache Service (GCS).
2: CHA detected that the ASM disk service time is higher than expected.

Gc cr block congested*: 0.00 µs, expected: 1.537 ms

Gc current block congested*: 0.00 µs, expected: 1.294 ms

Log file sync*: 69.871 ms, expected: 73.665 ms

Use the right-click popup menu, or press "c" to display all correlated probes belonging to all "Log file sync" member categories:



Instance orcl1 Shows 5 of 22 probes [1..5]

1: CHA detected a possible process priority issue in the Global Cache Service (GCS).
2: CHA detected that the ASM disk service time is higher than expected.

Average active sessions : 11.62, expected: 7.68

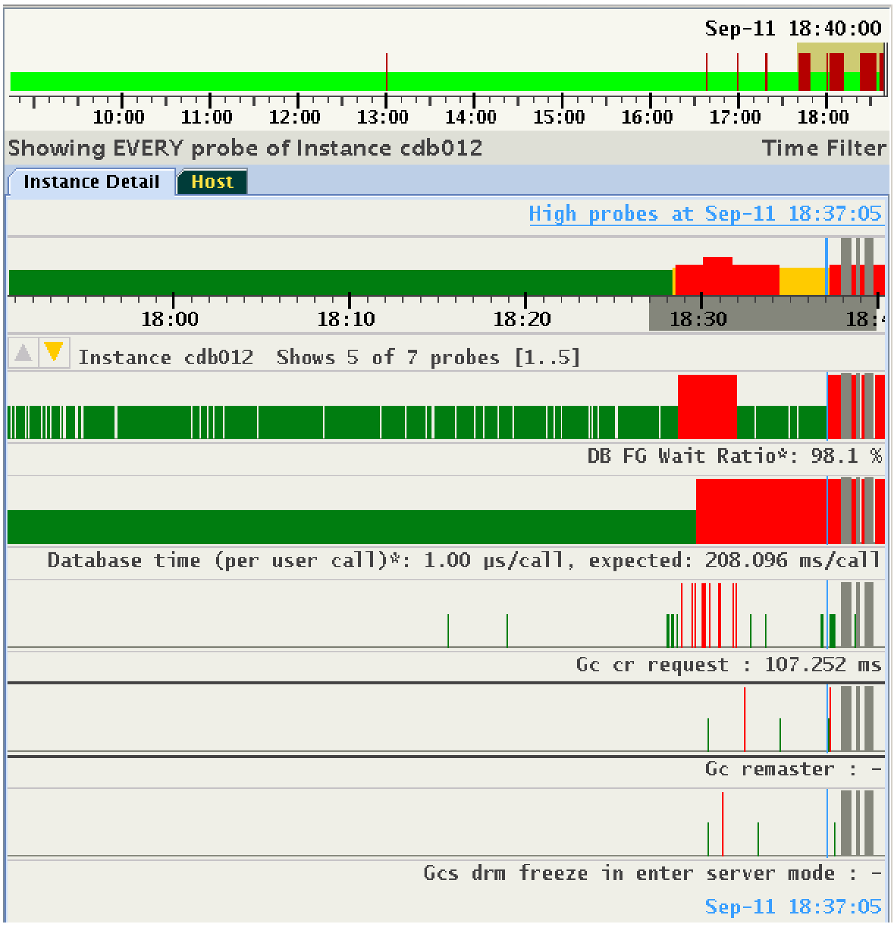
CPU queue length : 0, expected: 0

DB Idle : 0.00

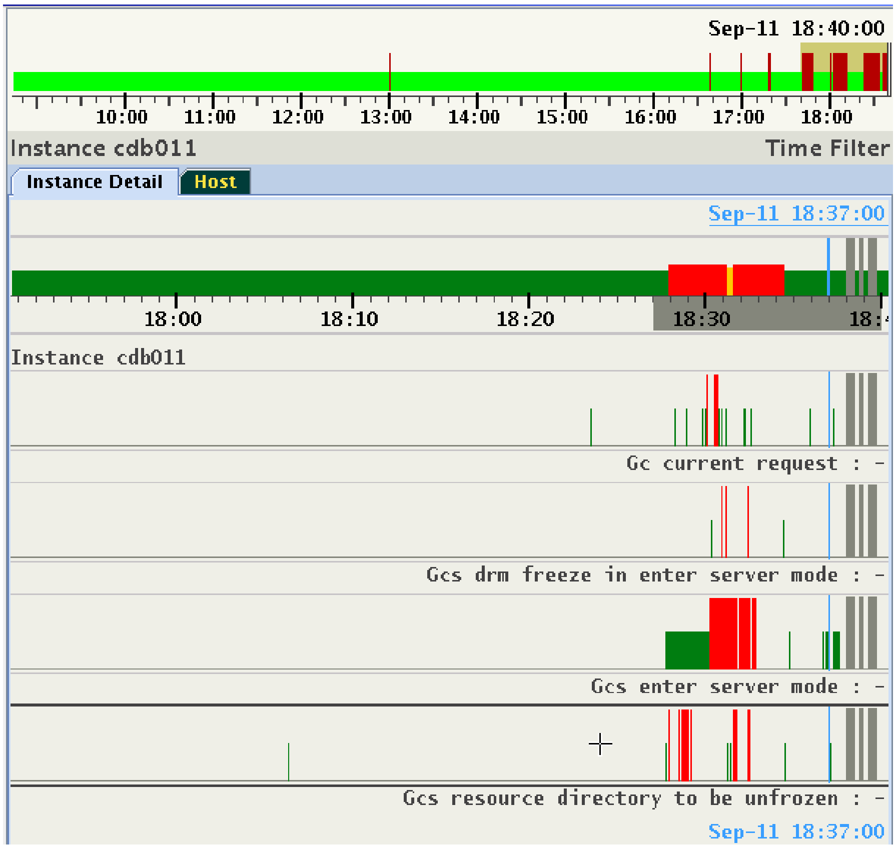
Should the number of probes not fit into the display, the Up/Down cursor icons appear and the status line shows details about visible subset of probes. See section 3 'Expert Mode' for the steps to select and create subsets of probes.

2.5 Selecting Abnormal Probes in any Time Range

Tracking probes in abnormal state between times points is supported. Move the cursor the first time point, press <Shift> key, press left mouse key and move the cursor to the second time point, release the mouse button, and than release <Shift>. The selected time range is indicated by a gray bar under the time line. The Information Strip indicates on the right side that a "Time Filter" is active. In the example, a total of 7 probes are indicated high between 18:27 and 18:40, with probes 1 to 5 visible in the display:


Sep-11 18:40:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
Showing EVERY probe of Instance cdb012
Time Filter
Instance Detail
Host
High probes at Sep-11 18:37:05
18:00
18:10
18:20
18:30
18:
Instance cdb012 Shows 5 of 7 probes [1..5]
DB FG Wait Ratio*: 98.1 %
Database time (per user call)*: 1.00 µs/call, expected: 208.096 ms/call
Gc cr request : 107.252 ms
Gc remaster : -
Gcs drm freeze in enter server mode : -
Sep-11 18:37:05

The selection of Time Range is preserved cluster wide, similar to a selection of Pin Point. User can change Target Entities and see the sets of abnormal probes in the selected time range. For example, in the same period of time only 4 high probes for another instance of the database are detected:



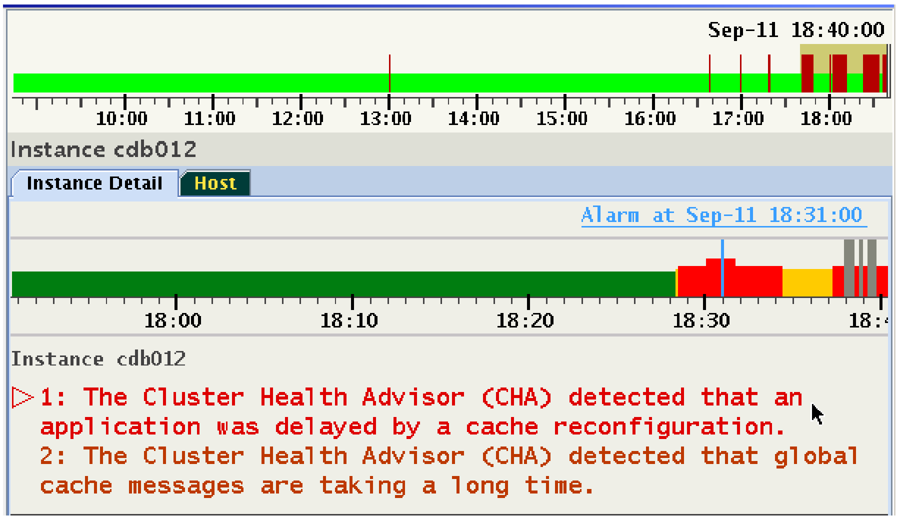
To deactivate the filter either select "Reset Time Selection" on the right click pop-up menu, or press `<Escape>`.

2.6 Problems

Whenever the timeline of the entity displays *red*, one or more Problems exist at that timestamp. Every Problem contains the following information:

1. Name. This is an internal identifier.
2. Description. An explanation of the nature of the diagnosed problem (displayed by default on the panel).
3. Confidence Probability in percentage (calculated per time point).
4. Root Cause and diagnosis of the problem.
5. Suggested corrective actions.
6. Set of Probes associated with the Problem (calculated per time point).
7. Set of Inference Chains in Bayesian Network, one chain per Probe associated with the Problem.
8. Set of Tables with detailed information (optional, calculated per time point).

Portions of this information will be displayed only in the Expert Mode, described in Section 3 "Expert Mode." Below is an example of 2 Problems indicated at 18:31:00:



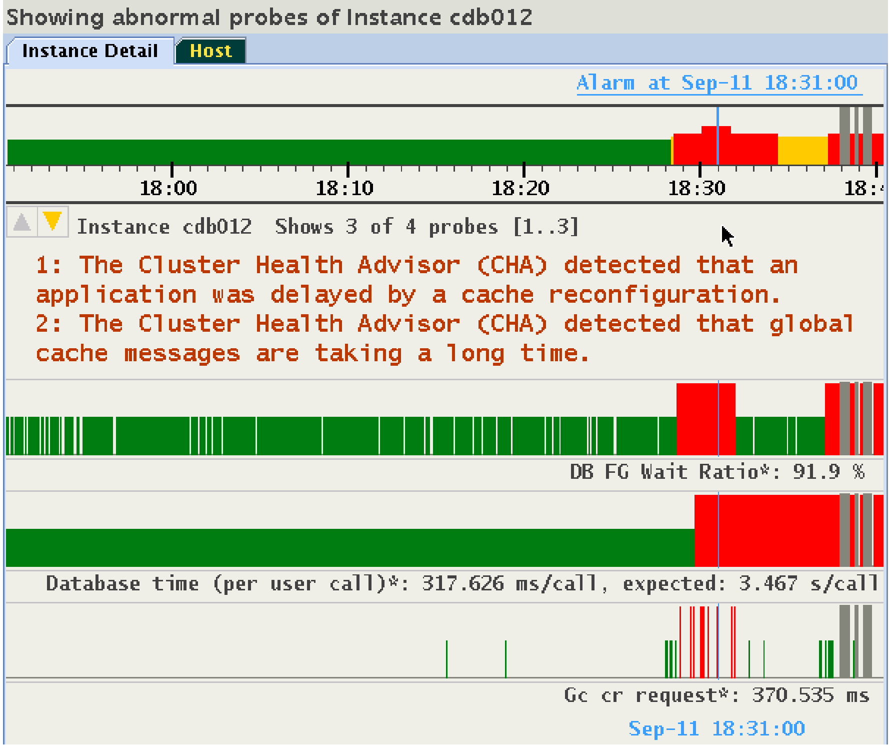
Instance cdb012

▷1: The Cluster Health Advisor (CHA) detected that an application was delayed by a cache reconfiguration.

2: The Cluster Health Advisor (CHA) detected that global cache messages are taking a long time.

Problems are listed in the order of their confidence probability. Problems that have probes that are *Key Performance Indicator (KPI)* will be displayed at the top of the list regardless of their confidence probability. The order and set of Problems may differ at each time point.
The Problem with a focus is indicated by a color change and a right arrow to the left of its number. Focus may also be changed using the `<Up/Down Arrow>` keys.

Each Problem has a set of Probes in abnormal state. As long a specific Problem is not *selected*, CHAG will not display any probes. This default preference for display of probes may be changed by using the right click pop-up menu, or by using a hot-key. For example, when selecting radio button *"Show all High Probes"* or pressing `"a"`, all abnormal probes across all problems at 18:31 are displayed:



Instance cdb012 Shows 3 of 4 probes [1..3]

1: The Cluster Health Advisor (CHA) detected that an application was delayed by a cache reconfiguration.

2: The Cluster Health Advisor (CHA) detected that global cache messages are taking a long time.

DB FG Wait Ratio*: 91.9 %

Database time (per user call)*: 317.626 ms/call, expected: 3.467 s/call

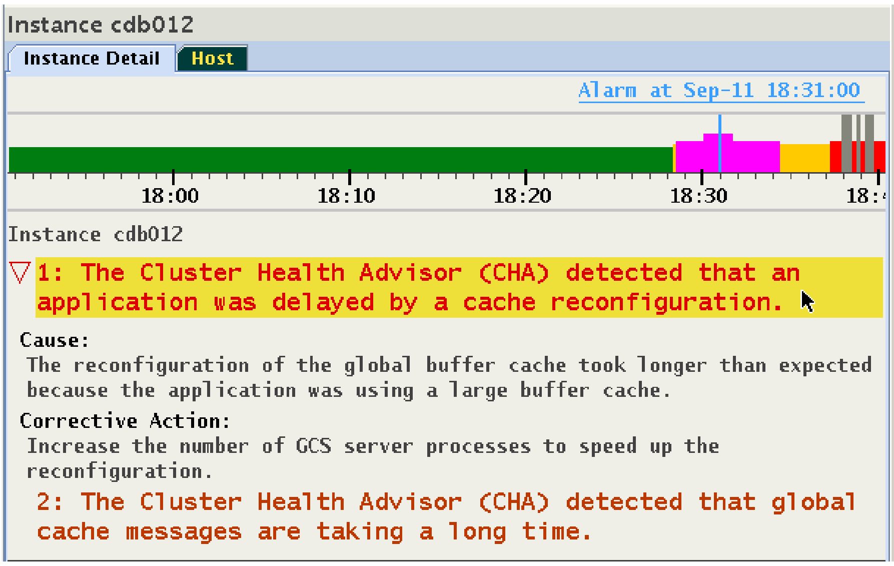
Gc cr request*: 370.535 ms

Sep-11 18:31:00

The highlighted Up/Down icons in the upper left side of the panel indicates that not all probes can be displayed. In this example the Information Bar displays `"Shows 3 of 4 probes [1..3]"`. Use `<Down/Up Arrow>` or `<Page-Up/Down>` keys to scroll through the probes while keeping same time position.

Probes specific to a Problem are still available by returning to the default mode, and by selecting the Problem of interest. Press "a" again to remove the display of all abnormal probes and return to the default mode.

To step through the analysis of a Problem in detail, highlight the Problem and either click on it, or press <Enter>. This action *selects* the highlighted Problem and displays the root cause diagnosis and a recommended corrective action. In addition, on the status timeline every occurrence of the same Problem is displayed in magenta. In the example below the top Problem was indicated between 18:28 and 18:34:



Instance cdb012

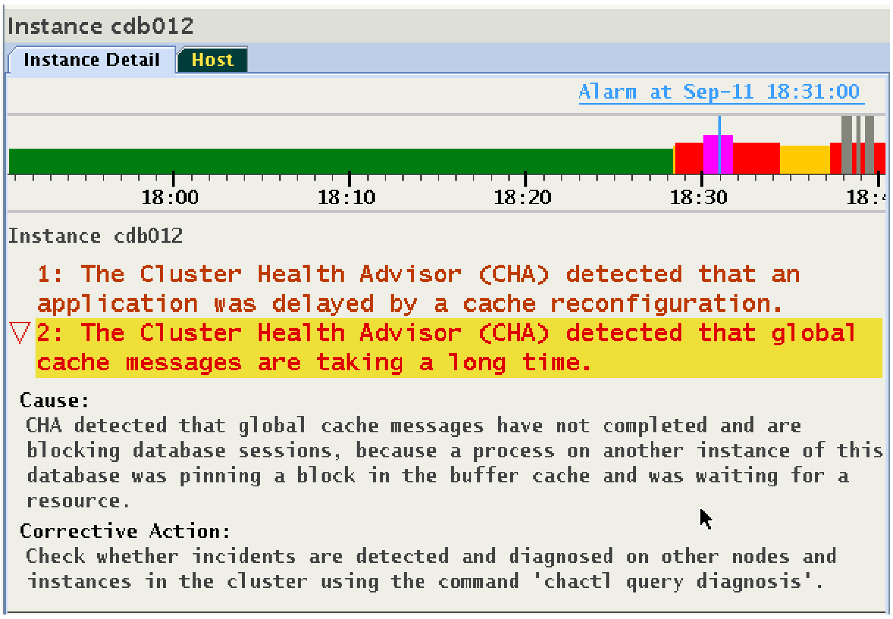
▽ 1: The Cluster Health Advisor (CHA) detected that an application was delayed by a cache reconfiguration.

Cause:
The reconfiguration of the global buffer cache took longer than expected because the application was using a large buffer cache.

Corrective Action:
Increase the number of GCS server processes to speed up the reconfiguration.

2: The Cluster Health Advisor (CHA) detected that global cache messages are taking a long time.

To switch to another Problem, either click on it, or use <Up/Down Arrow> to highlight it, and press <Enter>. Note how the magenta marked time period has changed on target's status timeline:



Instance cdb012

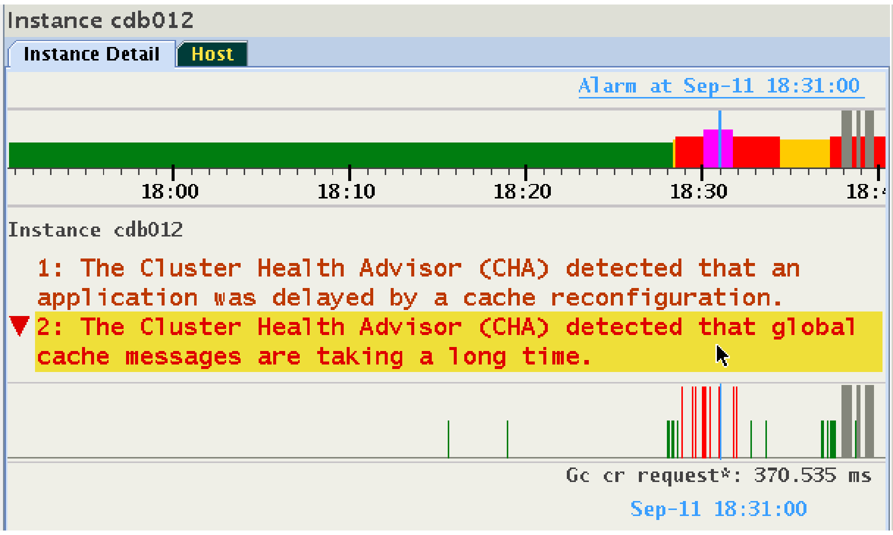
1: The Cluster Health Advisor (CHA) detected that an application was delayed by a cache reconfiguration.

▽ 2: The Cluster Health Advisor (CHA) detected that global cache messages are taking a long time.

Cause:
CHA detected that global cache messages have not completed and are blocking database sessions, because a process on another instance of this database was pinning a block in the buffer cache and was waiting for a resource.

Corrective Action:
Check whether incidents are detected and diagnosed on other nodes and instances in the cluster using the command 'chactl query diagnosis'.

Click once more on the selected Problem, or press <Enter> to display the set of the probes providing evidence supporting the Problem determination. In this example the problem has only one probe "Gc cr request":



Alarm at Sep-11 18:31:00

Instance cdb012

1: The Cluster Health Advisor (CHA) detected that an application was delayed by a cache reconfiguration.

▼ 2: The Cluster Health Advisor (CHA) detected that global cache messages are taking a long time.

Gc cr request*: 370.535 ms

Sep-11 18:31:00

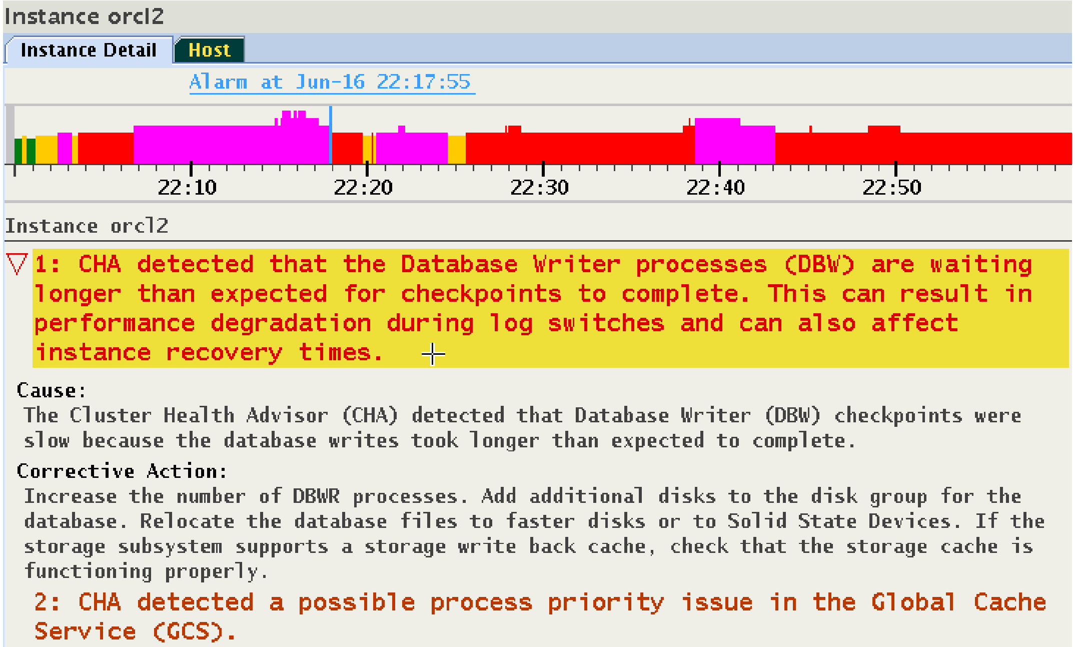
Press `<Enter>` again, and note that the display returns to its list of Problems to facilitate analysis of the additional ones. There are the following three Problem displays which cycle upon clicking or pressing `<Enter>`:

1. Problem is highlighted (empty right arrow points to the Problem name).
2. Problem is selected, shows textual descriptions of Cause, Corrective Action (empty down arrow points to the descriptions).
3. Problem is selected, shows status timelines of probes associated with the Problem (filled down arrow points to the probe timelines).

Note that when selecting a different Problem by a mouse click or using `<Enter>`, the display mode stays the same allowing stepping through each Problem in the same mode.

2.7 Browsing through active time of a Problem

Once a Problem is selected, the time periods in which it was active is marked by a magenta color on the status timeline. Press and hold `<Shift>` and use `<Left/Right Arrow>` to fast forward to a previous or next active time point in which this Problem was indicated. For example, at 22:17:55 a "*DB Writer checkpoint*" problem is displayed. The magenta color shows several time ranges in which this condition was also diagnosed:



Alarm at Jun-16 22:17:55

Instance orcl2

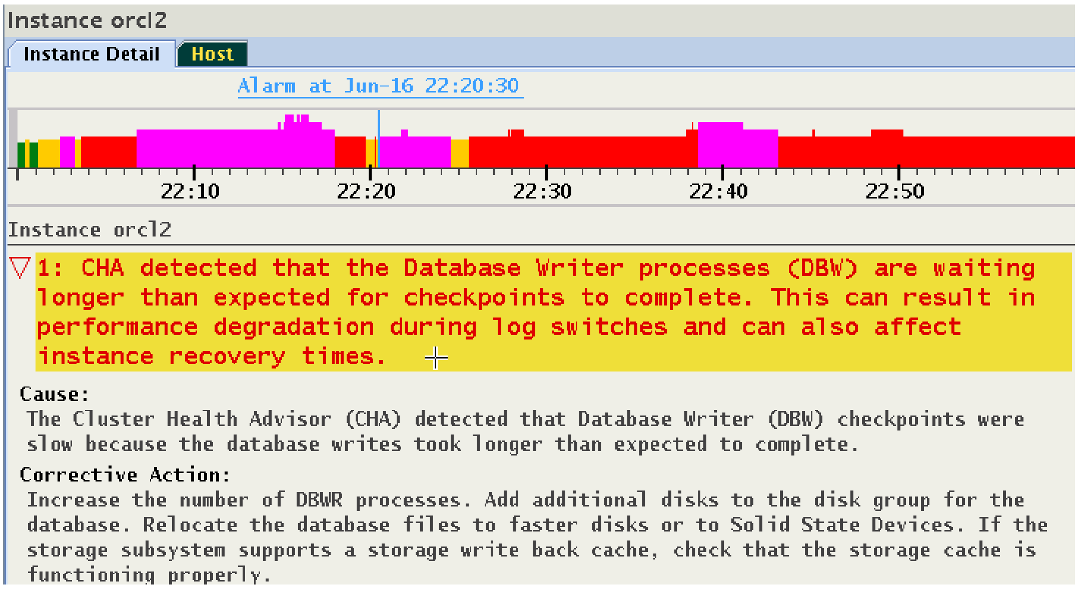
▽ 1: CHA detected that the Database Writer processes (DBW) are waiting longer than expected for checkpoints to complete. This can result in performance degradation during log switches and can also affect instance recovery times.

Cause:
The Cluster Health Advisor (CHA) detected that Database Writer (DBW) checkpoints were slow because the database writes took longer than expected to complete.

Corrective Action:
Increase the number of DBWR processes. Add additional disks to the disk group for the database. Relocate the database files to faster disks or to Solid State Devices. If the storage subsystem supports a storage write back cache, check that the storage cache is functioning properly.

2: CHA detected a possible process priority issue in the Global Cache Service (GCS).

The selected timestamp is at the end of the magenta colored time range. Press `<Shift+Right Arrow>`, and the cursor will move to the next later timestamp in which the same Problem was reported. In this case, it was 22:20:30, approx 2 minutes later:



1: CHA detected that the Database Writer processes (DBW) are waiting longer than expected for checkpoints to complete. This can result in performance degradation during log switches and can also affect instance recovery times.

Cause:
The Cluster Health Advisor (CHA) detected that Database Writer (DBW) checkpoints were slow because the database writes took longer than expected to complete.

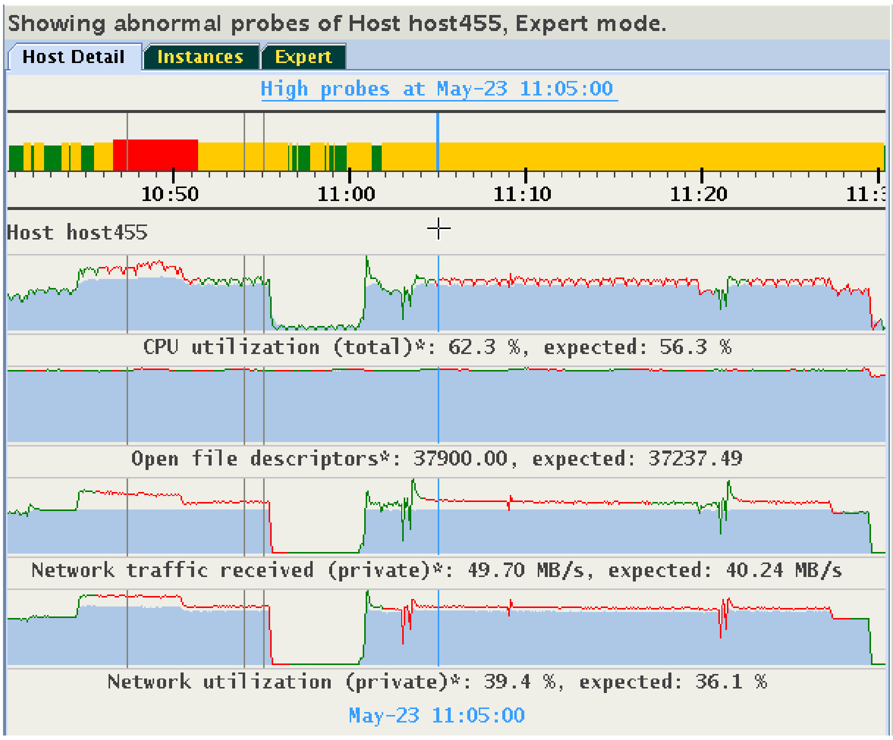
Corrective Action:
Increase the number of DBWR processes. Add additional disks to the disk group for the database. Relocate the database files to faster disks or to Solid State Devices. If the storage subsystem supports a storage write back cache, check that the storage cache is functioning properly.

3. Expert Mode

Section 2 cover the *Standard* mode of operation. However, since this diagnosis is based upon an applied machine learning model of predicted metric values, there is always a probability that an abnormal condition will not be diagnosed correctly - either raising a warning too late or providing a false one. To facilitate an advanced analysis of the metrics and their observed values against the predicted ones, an *Expert Mode* is provided.

3.1 Activating Expert Mode

Press "e" to toggle the Expert Mode, or use the right-click pop-up menu and select the "Expert" box . The probe status timelines change their appearance, and an additional "Expert" tab appears. Only the Target timeline stays unchanged:


Showing abnormal probes of Host host455, Expert mode.
Host Detail
Instances
Expert
High probes at May-23 11:05:00
10:50
11:00
11:10
11:20
11:3
Host host455

CPU utilization (total)*: 62.3 %, expected: 56.3 %

Open file descriptors*: 37900.00, expected: 37237.49

Network traffic received (private)*: 49.70 MB/s, expected: 40.24 MB/s

Network utilization (private)*: 39.4 %, expected: 36.1 %

May-23 11:05:00

Each probe histogram contains now an overlapping display of 3 values:

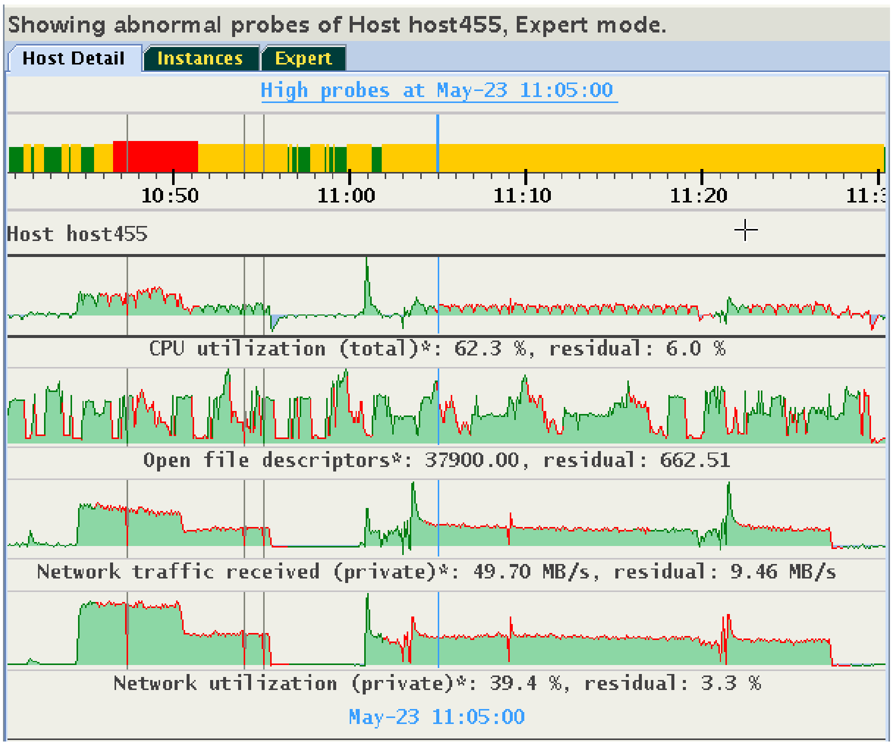
1. Time series for the Expected (or: Predicted) value displayed as light blue lines from 0 to its value.
2. Time series for the Observed value. Plotted atop of the predicted values in Green or in Red.
3. State of the probe, marked by the Green/Red color of the observed value plot.

Some probes do not have expected values. Others, like the Database wait events, will not have observed values at every time point.

This display supports evaluating how well the exiting model calibration aligns with the actual observed values. If the observed values are consistently and significantly lower than predicted, it is likely the model is not well-calibrated to this workload. Performing a CHA calibration based upon an actual normal workload period should be considered.

The histograms of the probes are *dynamically self-adaptive* to the range of values of each metric's predicted or observed value. This causes that parts of the visible time series are "flat", like the "`Database time (per user call)`" in the above image. In other cases the predicted and observed values are so similar to each other, that the differences between them are barely visible. In such cases user might prefer to see a plot of a difference between Observed and Predicted value, called a "*Residual*." Select box "`Display Residuals`" on the floating menu, or press "`r`" to toggle between the display of Residuals, or of Predicted/Observed values:

The display scale for each probe is dynamically set to the maximum value of the displayed predicted and observed metrics (auto-scaled). This may result in cases where the predicted and observed values are so similar that the differences between them are not easily visible. In such cases an alternative *Residual* plot of a difference between Observed and Predicted value is available from the right click pop-up menu's "`Display Residuals`", or by pressing "`r`" to toggle between the display of Residuals, or of Predicted/Observed values:


Showing abnormal probes of Host host455, Expert mode.

Host host455

CPU utilization (total)*: 62.3 %, residual: 6.0 %

Open file descriptors*: 37900.00, residual: 662.51

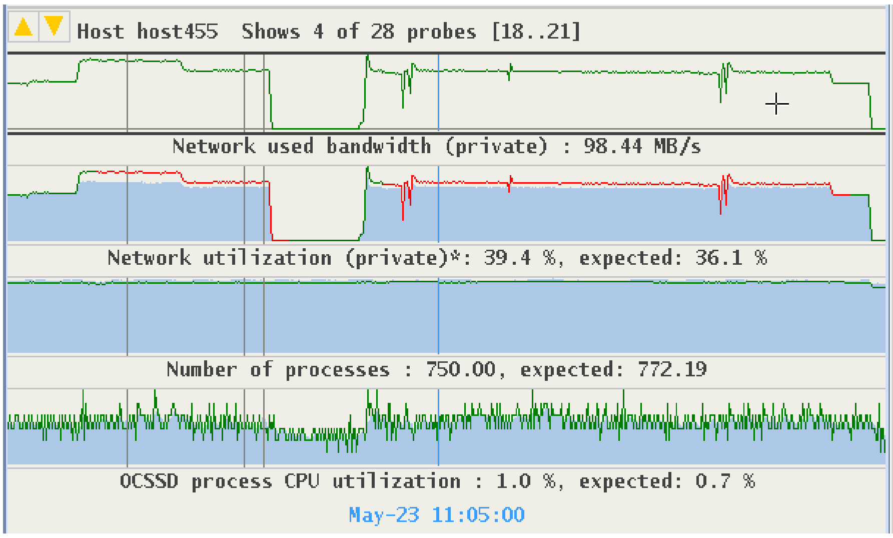
Network traffic received (private)*: 49.70 MB/s, residual: 9.46 MB/s

Network utilization (private)*: 39.4 %, residual: 3.3 %

May-23 11:05:00

Note how the differences in values of "`Open file descriptors`" became now well visible. Positive residuals are display by a light green lines (observed value greater than the predicted), and negative residuals are by light blue lines (observed value less than predicted).

Note that this display will be meaningful only for probes for which CHA provides both expected and observed values. This is not always the case. In this mode timelines for probes without expected values will appear empty. An example: The metric "`Network used bandwidth`" does not have expected values:


Host host455 Shows 4 of 28 probes [18..21]

Network used bandwidth (private) : 98.44 MB/s

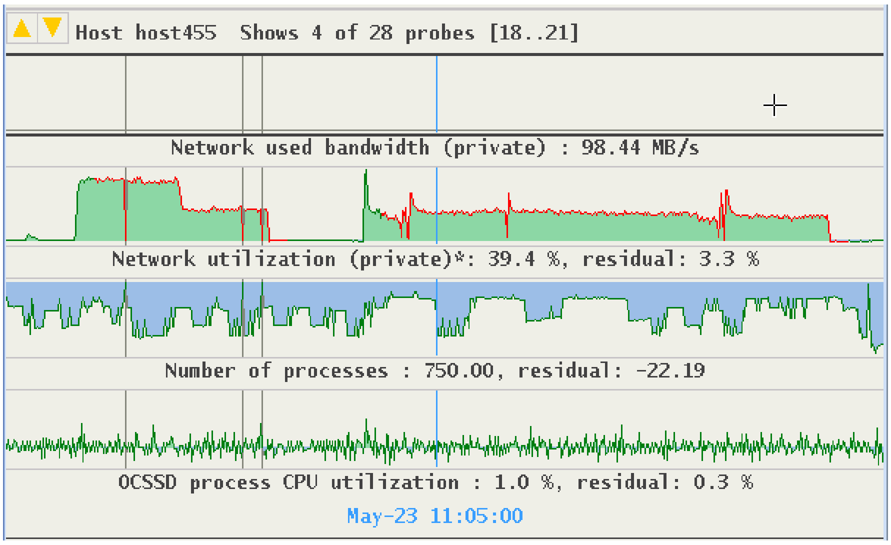
Network utilization (private)*: 39.4 %, expected: 36.1 %

Number of processes : 750.00, expected: 772.19

OCSSD process CPU utilization : 1.0 %, expected: 0.7 %

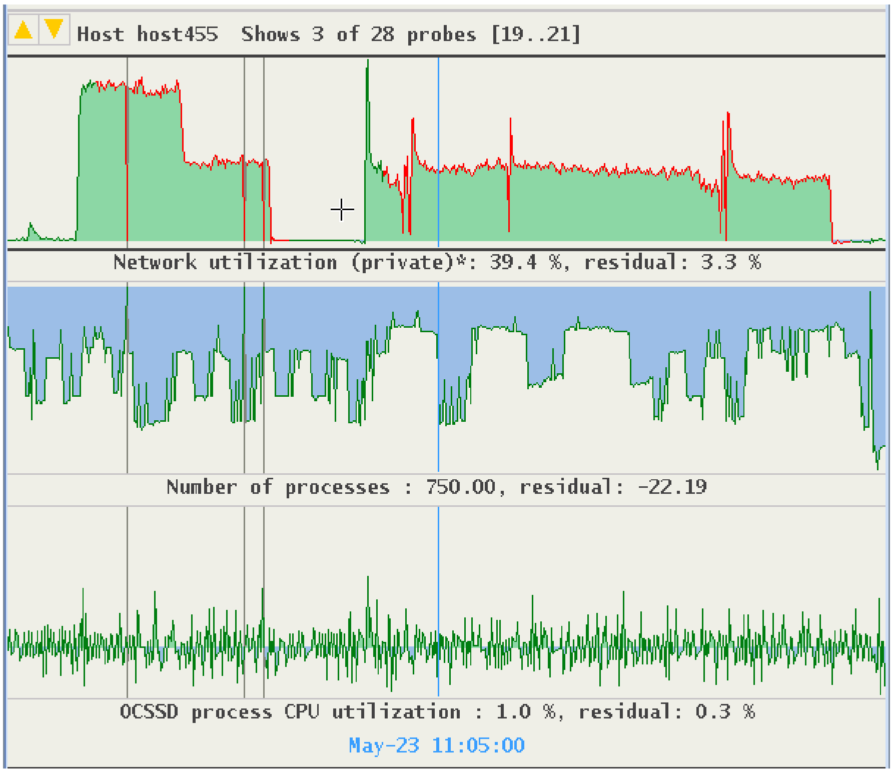
May-23 11:05:00

Display of residuals for this metric is empty, whereas display for "`Network utilization`" and "`Number of processes`" provide good visualization of differences between their expected and observed values:


Host host455 Shows 4 of 28 probes [18..21]
Network used bandwidth (private) : 98.44 MB/s
Network utilization (private)*: 39.4 %, residual: 3.3 %
Number of processes : 750.00, residual: -22.19
OCSSD process CPU utilization : 1.0 %, residual: 0.3 %
May-23 11:05:00

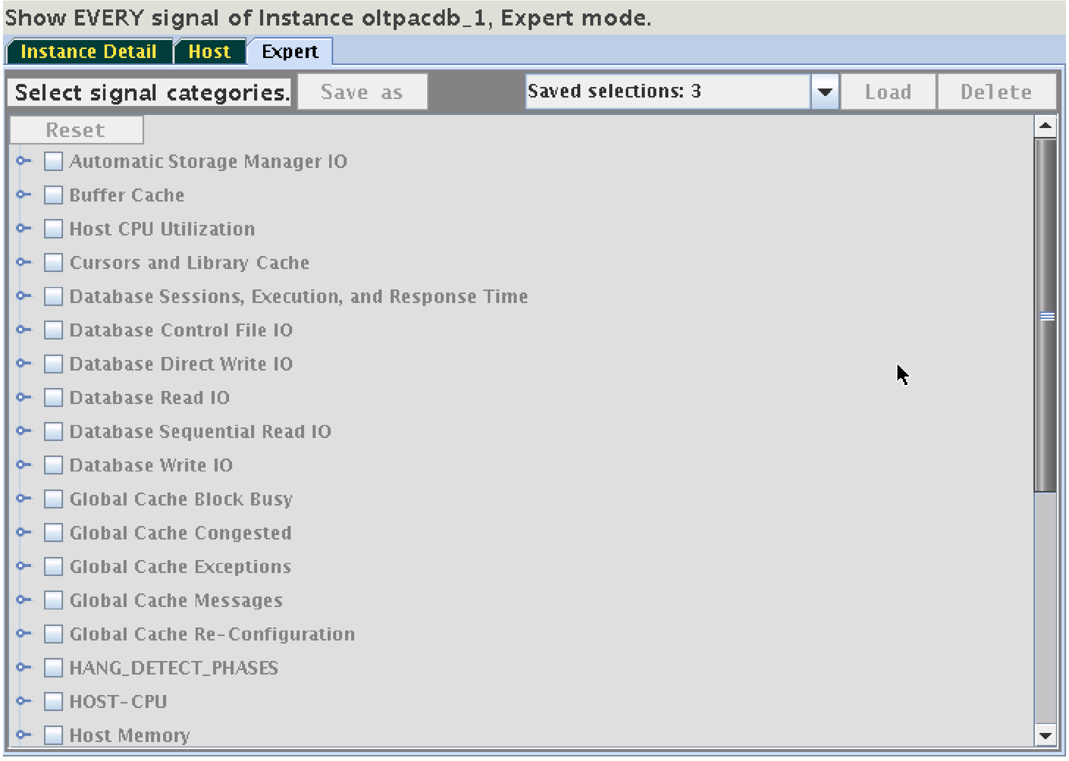
3.2 Resizing Expert Diagrams

Due to the auto-scaling important details in observed and predicted values may not be easily visible. A vertical zoom function is available by placing the cursor over the area of interest and press `<Control+Left mouse button>` to drag the cursor up or down. The maximum height of each graph is 128 pixels. The example below shows the previous image enlarged to display more detail in graphs:


Host host455 Shows 3 of 28 probes [19..21]
Network utilization (private)*: 39.4 %, residual: 3.3 %
Number of processes : 750.00, residual: -22.19
OCSSD process CPU utilization : 1.0 %, residual: 0.3 %
May-23 11:05:00

3.3 Selecting Custom Set of Probes

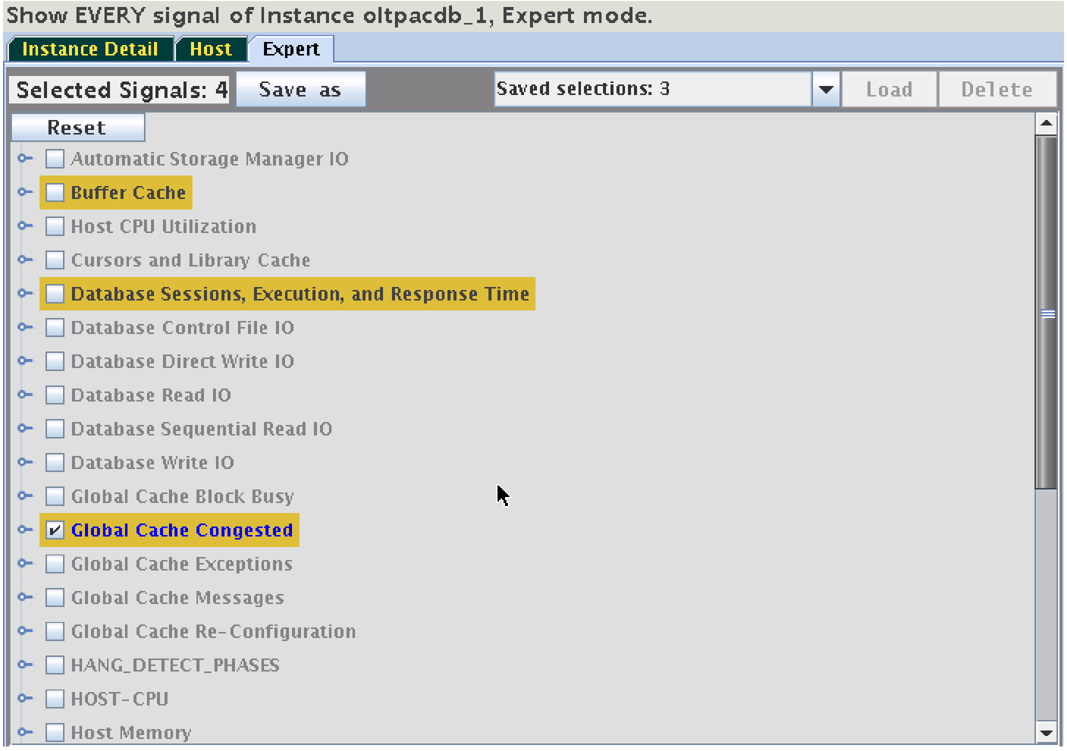
Probes monitored by CHA are grouped into *categories*. A custom set of probes may be specified, based on their categories, or on individual probes from different categories. Click on the "Expert" tab, or use `<Tab>` and `<Left/Right Arrow>` keys to navigate to the Expert panel:



Show EVERY signal of Instance oltpacdb_1, Expert mode.

- □ Automatic Storage Manager IO
- □ Buffer Cache
- □ Host CPU Utilization
- □ Cursors and Library Cache
- □ Database Sessions, Execution, and Response Time
- □ Database Control File IO
- □ Database Direct Write IO
- □ Database Read IO
- □ Database Sequential Read IO
- □ Database Write IO
- □ Global Cache Block Busy
- □ Global Cache Congested
- □ Global Cache Exceptions
- □ Global Cache Messages
- □ Global Cache Re-Configuration
- □ HANG_DETECT_PHASES
- □ HOST-CPU
- □ Host Memory

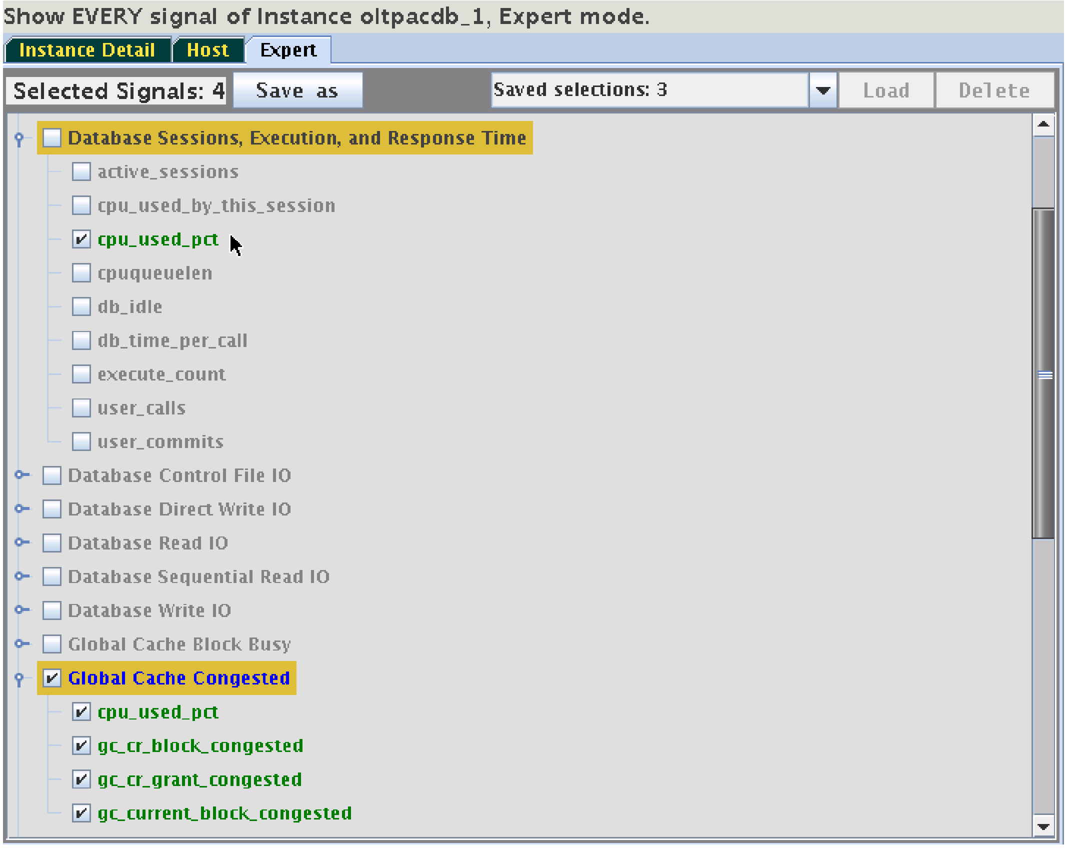
This panel provides access to a hierarchical tree with probe categories. The number of probe categories may vary between CHA models. To create a custom set of displayed probes, select either a category, or expand its tree and select individual probes from any category. For example, select the category "`Global Cache Congested`". Its name is highlighted in yellow and its checkbox is selected. Note two additional categories became highlighted, without their check boxes being selected. This means that one or more of the probes from "`Global Cache Congested`" are also their members:



Show EVERY signal of Instance oltpacdb_1, Expert mode.

- □ Automatic Storage Manager IO
- □ Buffer Cache
- □ Host CPU Utilization
- □ Cursors and Library Cache
- □ Database Sessions, Execution, and Response Time
- □ Database Control File IO
- □ Database Direct Write IO
- □ Database Read IO
- □ Database Sequential Read IO
- □ Database Write IO
- □ Global Cache Block Busy
- ☑ Global Cache Congested
- □ Global Cache Exceptions
- □ Global Cache Messages
- □ Global Cache Re-Configuration
- □ HANG_DETECT_PHASES
- □ HOST-CPU
- □ Host Memory

Expand the selected category and one of the other yellow highlighted categories to see which probe they share. In this case, it is the "`cpu_used_pct`":



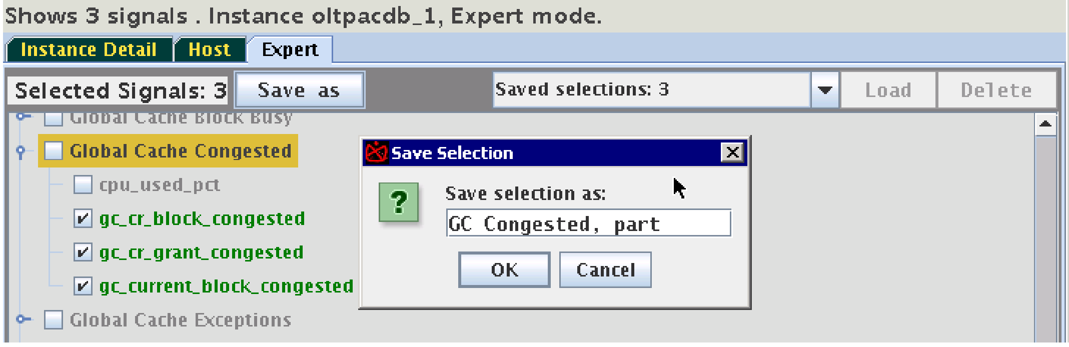
Show EVERY signal of Instance oltpacdb_1, Expert mode.

Selected Signals: 4

Saved selections: 3

- □ Database Sessions, Execution, and Response Time
  - □ active_sessions
  - □ cpu_used_by_this_session
  - ☑ cpu_used_pct
  - □ cpuqueuelen
  - □ db_idle
  - □ db_time_per_call
  - □ execute_count
  - □ user_calls
  - □ user_commits
- □ Database Control File IO
- □ Database Direct Write IO
- □ Database Read IO
- □ Database Sequential Read IO
- □ Database Write IO
- □ Global Cache Block Busy
- ☑ Global Cache Congested
  - ☑ cpu_used_pct
  - ☑ gc_cr_block_congested
  - ☑ gc_cr_grant_congested
  - ☑ gc_current_block_congested

By unchecking the probe "`cpu_used_pct`", only one category will remain highlighted. Switch to "Instance Detail" tab to see that now only the 3 signals of the category "`Global Cache Congested`" are being displayed on the instance tab. Selections may be stored for reuse. On the Expert tab use the button "Save As" to save the selection under a specified name:



Shows 3 signals . Instance oltpacdb_1, Expert mode.

Selected Signals: 3

Saved selections: 3

- □ Global Cache Block Busy
- □ Global Cache Congested
  - □ cpu_used_pct
  - ☑ gc_cr_block_congested
  - ☑ gc_cr_grant_congested
  - ☑ gc_current_block_congested
- □ Global Cache Exceptions

Save Selection

Save selection as: GC Congested, part

These saved selections will be available after a restart of CHAG on the same host. Use the drop down menu "Saved selections" to retrieve any of the saved sets of probes. Press on "Load" to activate it, or press on "Delete" to remove it from the persistent storage. Note, that selecting of the saved set without a corresponding Loading does not activate it.

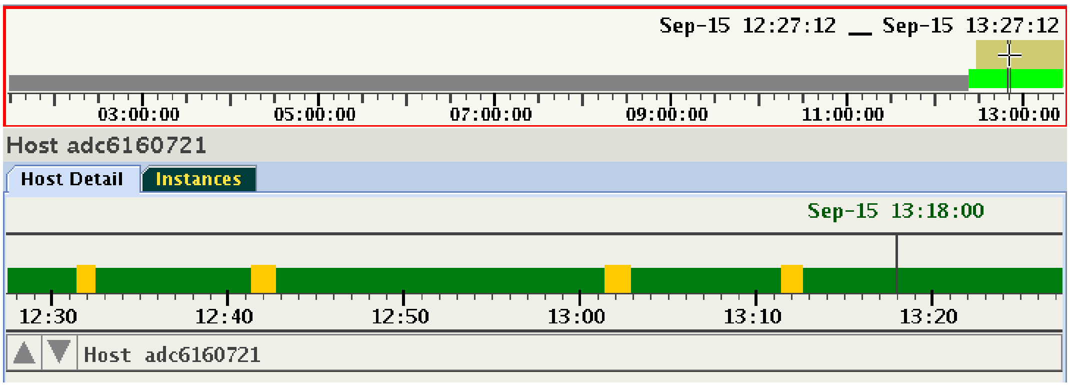
4. Live and Passive Sessions

CHAG can be run locally on one of the RAC cluster nodes and connected to the Grid Infrastructure Management Repository (GIMR,) or by reading exported GIMR data from a file. Note, remote GIMR connections are not supported as the SQL connection is not encrypted. When connected to a GIMR Database and actively receiving samples in real time from a live system, CHAG maintains in parallel 2 separate sessions:

1. Live Session (Primary): Receives in real time current metrics.
2. Past (Replay) Session: Displays statically a situation encountered at an earlier time.

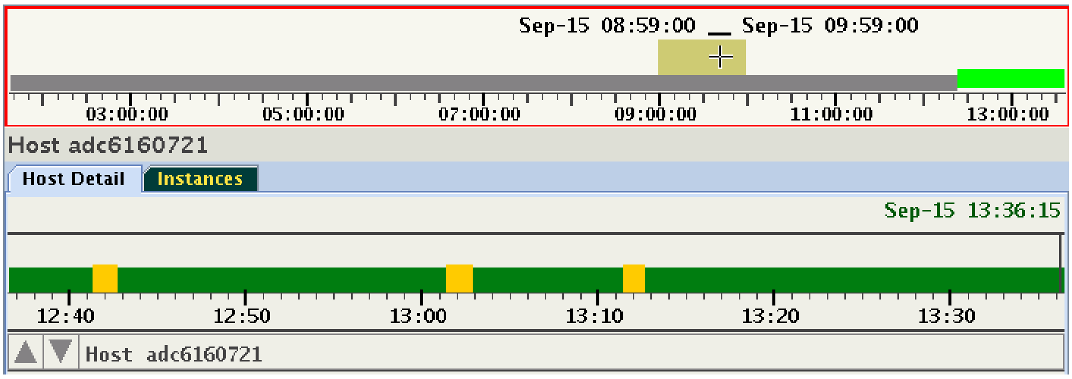
Access to a Replay Session is available via the System Timeline (Ticker Tape) *only* when CHAG is connected to a GIMR. When CHAG is started with "`-f <file>`" as its feed, Ticker Tape is not active. In such a case the primary session is passive, and CHAG does not have any possibility to retrieve data from the past.

Use Ticker Tape to locate the time period of interest. Place cursor over the Time Selector and press `<Shift>`. Note that Ticker Tape displays now information about the time range corresponding to the location of the Time Selector:

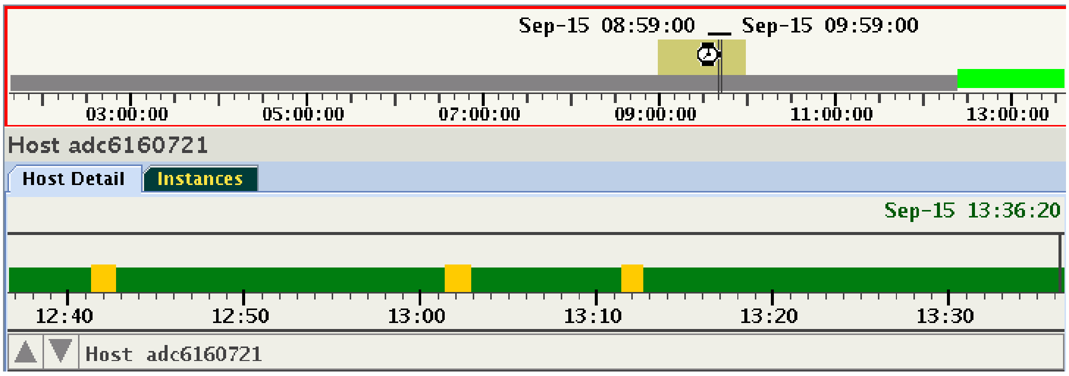


In this example, the selected time period is the hour from 12:27:12 to 13:27:12. The Time Selector width (time period) is customizable by using the "`-q <minutes>`" command line parameter.

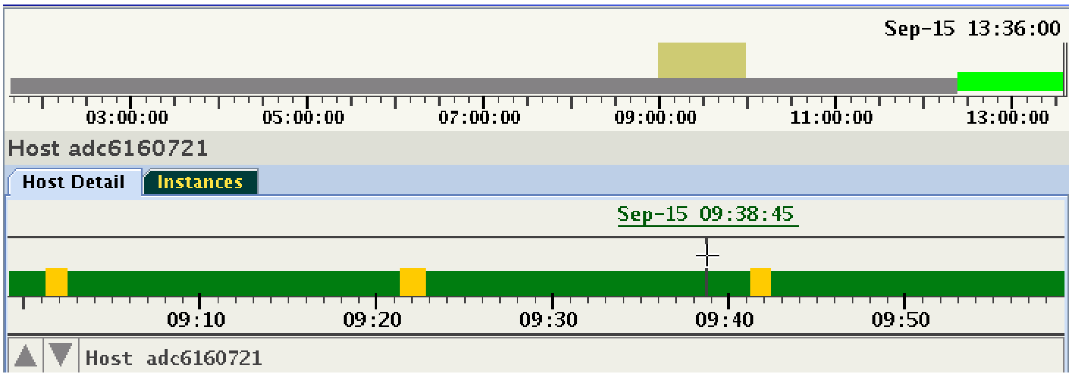
While holding down the `<Shift>`-key, press left mouse button and slide the Time Selector to an older time point. Time Selector may also be moved without using a mouse: Press `<Shift>` and while holding it, press `<left/right arrow>` to shift time selector by 30minutes. In this example Time Selector was moved to to approx. 9:00:



Release `<Shift>`. CHAG will issue a query to GIMR requesting a set of samples for the selected period of time. The time to retrieve this data can be substantial, especially when CHA monitors many databases in a large cluster. While the query and the parsing process of the data is in progress, a clock "Wait-Cursor" is displayed. In a background the current timelines of the live session continue to receive data and advance accordingly:



Once the query and parsing of the data is finished, CHAG displays the data of a "*Replay Session*". In this example this session covers one hour, approx. 9:00..10:00 o'clock:



This display never changes. User can investigate the data in the customary way without any time restriction. However, the "*Live Session*" is still active and in the background the data is constantly being collected. To restore the Live Session either slide the Time Selector to the right end, or with the cursor hovering over System Timeline press "=". Note that the Past Session data is discarded from memory once the session is switched to Live.

5. Console Commands

Start CHAG with the option -i:

```
  chag -i <other options and parameters>
```

The console will activate an interactive Command Line Interface (CLI). Enter a question mark (?) to see the list of available commands:

```
cha> ?
  list <item>:
     entities:   List of entities and time ranges of their relations
     inputs:     Lists input feeds
     kinds:      Kinds of Entities and number of their metrics
            <verbose>   lists metrics with every kind
     metrics:    List of all known metrics and their units of measure
            <details>   lists metrics with full name and description of value
     probes:     List of all probes (signals) for all entities
            <diff>         lists probes where number of predicted and observed values differ
            <nounit>       lists probes without unit of measure
            <noobserved>   lists probes without observed values
            <nopredicted>  lists probes without predicted values
            <missing>      lists probes with missing values in some samples
            <flagged>      lists probes flags set
     units:      List units of measure
            <diagnose>  shows conversion rules and values at conversion thresholds
  trace <item>       Switch tracing on/off, indicated by '+' or '-'
     f: (feed)          live feed activity
     d: (db)            toggle 'alter session' for trace event 10046 (ORA DB)
     i: (input)         copy incoming data to the log-file
     p: (probes)        shows internal or descriptive probe names
     r: (rootcause)     print to console CLOB's containing a RootCause of a Problem
     t: (topology)      shows changes in set of entities (topology)
  version:           Version of the (1) UI, (2) data stream and (3) Java Virtual Machine
  zoom (in|out):     Set or reset Magnify option

  quit:              Exit CHAG GUI.
```

For convenience, CHAG's CLI provides an abbreviation grammar. For example: Instead of typing "`version`", user may simply type '`v`':

```
cha> version
   CHA UI version:     V1.00.000
   Data version:       V0.17
   PL/SQL package:     V0.10.11.2
   Java version:       1.8.0_77 on Linux
cha> v
   CHA UI version:     V1.00.000
   Data version:       V0.17
   PL/SQL package:     V0.10.11.2
   Java version:       1.8.0_77 on Linux
cha>
```

When more than one command starts with the same prefix, they need to be disambiguated. For example "debug" versus "device" would require typing at least 3 letters to correctly identify the desired command. These commands provide summaries not available in graphical form.

Appendix A: List of Hot Keys

On any Place:

`^+`	`(control +)`	Enlarge (Zoom In). On many keyboards `'+'` might stand over `'='`, use "shift" to reach `'+'`
`^-`	`(control -)`	Shrink (Zoom Out)
`w`	`(lower case w)`	Toggle theme (Day/Night)

On a target panel (Host or Instance):

`a`	`(lower case a)`	Show all probes in high state.
`A`	`(upper case A)`	Show all existing probes. (NOTE: Set may be filtered in Expert Tab).
`c`	`(lower case c)`	Show all correlated probes (belonging to the same Probe Category).
`e`	`(lower case e)`	Toggle expert mode.
`r`	`(lower case r)`	Toggle values predicted(expected)/residuals.

p	(lower case p)	On state histogram: pin cursor line at the current position.
f	(lower case f)	On state histogram: follow (unpin).
->	(shift right-arrow)	Jump to the next occurrence of the selected Problem.
<-	(shift left-arrow)	Jump to the previous occurrence of the selected Problem.
^q	(control q)	Print the values of the focused-on probe. before and after the position marker into the console window.
<escape>		Remove time filter, unpin, deselect a Problem

Zoom Signal Histograms in Host/Instance panel, between 24..128 pixels (cursor has to be in the histogram area):

control+mouse left-button drag up:	makes the histograms smaller
control+mouse left-button drag down:	makes the histograms larger
control+arrow up:	makes the histograms smaller
control+arrow down:	makes the histograms larger

Signal navigation (active when not all signals visible on a panel):

Arrow Up/Down :	navigate between signals, scroll visible signals by one.
Page-Up/Page-Down:	navigate between signals, scroll entire page of signals.

On System Timeline (Ticker Tape):

= (equal sign) :	jump to the current timestamp from a "Past Session", resume "Live Session"
Shift + Left/Right Cursor:	shift time selector by 30min to the left or right

Entity navigation on all non-target panels:

<Enter>: selects the focused entity and displays the panel associated with the Entity.

Appendix B: Set of Persistent Settings.

1. Theme dark/light
2. Last time of start
3. Window size and position
4. Last selected set of probes
5. Every set of Named Custom Selections of probes

Appendix C: Accessibility aspects.

In order to enable accessible technology in Java on Windows, user has to follow the instructions given in:

Enabling and Testing Java Access Bridge

CHAG can be used without the help of the mouse. Every operation can be achieved using keys: Tab, Cursor Keys, Enter, several hot-keys described in appendix A. Some of the operations are available in combination with shift and control keys. No timeout exist on any of these operations, thus they can be used in conjunction with Sticky Keys and Slow Keys. The navigation contains components from standard Java "Swing" augmented by custom implementation of navigation in components designed for the unique, specialized displays in CHAG.

A magnification operation allows to enlarge text and components on panels. CHAG's displays are always high contrast, without use of images. They do not change with OS High Contrast mode, and CHAG's display or mode of operation does not affect the remaining desktop.

CHAG can operate with assistive technology software JAWS from Freedom Scientific, V17 and later.

Appendix J: Customizing Java run time system.

CHAG is written in Java, what ensures its platform independence. The script "chag" invokes the Java Virtual Machine (JVM) with Oracle classes. Knowledgeable users may consider customizing this script, or use environment variable "**_JAVA_OPTIONS**" to influence the way the JVM executes code.

JVM is the run-time process which interprets Java classes. All contemporary JVM's incorporate some method of on-the-fly translation of bytecode into native code. Dominating in this field is the Hot Spot. Except for beginning invocations of classes, in most cases Java methods run later in native code. Consequently they perform at speeds comparable to programs written in native languages, such as C.

Furthermore, on many platforms Java supports native 2D and 3D graphics with a *hardware acceleration* through the use of the Open GL libraries, what once more significantly improves display performance. It is highly recommended that Open GL will be configured for default use with JVM. Information about the Open GL library is available at: **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 Open GL:

https://docs.oracle.com/javase/8/docs/technotes/guides/2d/flags.html#opengl

https://docs.oracle.com/javase/8/docs/technotes/guides/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
```

CHAG 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 Open GL is not available::

```
Picked up _JAVA_OPTIONS: -Dsun.java2d.opengl=True -Dsun.java2d.d3d=true
Could not enable OpenGL pipeline for default config on screen 0
```

In this case, the system does not have a graphics card supporting OpenGL. The following is an example of a system with a graphics card supporting OpenGL:

```
OpenGL pipeline enabled for default config on screen 0
```

In the Java SE 6 release, you can get detailed information about the startup procedures of the OpenGL-based Java 2D pipeline by using the `J2D_TRACE_LEVEL` environment variable as follows.

On Windows:

```
# set J2D_TRACE_LEVEL=4
# java -Dsun.java2d.opengl=True YourApp
```

On Solaris OS and Linux:

```
# J2D_TRACE_LEVEL=4
# export J2D_TRACE_LEVEL
# java -Dsun.java2d.opengl=True YourApp
```

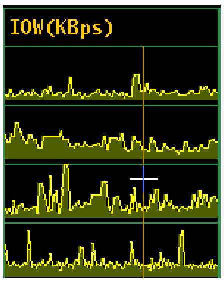
In the case of some graphics cards, OpenGL requires the option: `sun.java2d.opengl.fbobject=false`. See the section "3.1.5.5 *Diagnosing Rendering and Performance Issues*" in this document:

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

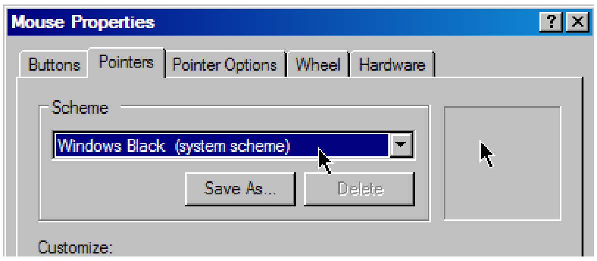
This link is the current and comprehensive description of potential issues with OpenGL and Java 2D drawing package in conjunction with specific hardware/driver versions.

Appendix W: 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:



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



Mouse Properties

Scheme: Windows Black (system scheme)

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