Overview

In a SAS Viya environment, Cloud Analytic Services (CAS) uses a controller node to distribute work to worker nodes. In this type of distributed environment, it is important to monitor the performance of each of the nodes to ensure that no nodes are becoming overloaded and no nodes are slowing down. You should also monitor session processes on the CAS nodes to ensure that individual processes are not consuming excessive resources.

Monitoring functions in SAS Viya are provided by these facilities:

**Grid Monitor tab in the CAS Server Monitor application.**
- provides histograms that enable you to quickly view the CPU load, memory usage, and CAS process performance on all nodes in your CAS grid. These views enable you to easily identify overloaded nodes. You can also manage and control processes through monitoring tools such as the UNIX top command, a terminal window, and a capability to kill defunct processes. If you are a CAS administrator, one histogram is displayed for each node on the CAS grid.

If you are not an administrator, only the histogram for the grid controller node is displayed.

**CAS start-up or session options**
- returns performance metric information to a terminal each time a CAS action runs. The data provided by the -metrics CAS server option enables you to monitor the CPU load on the CAS grid and determine how efficiently the CAS grid is processing actions.
Monitoring: How To

Using the CAS Server Monitor

Access CAS Server Monitor
To log on to CAS Server Monitor, open a web browser and enter the following URL in the address field:

https://controller-machine:HTTPSPort

You must have an active CAS Server session in order to access CAS Server Monitor.
For more information, see “Access the Monitor” in SAS Viya Administration: Using CAS Server Monitor.

Monitor CAS Process Performance
1  In CAS Server Monitor, beneath the SAS Cloud Analytic Services banner, click .
2  Select Add View ➔ CAS Process CPU Usage.
   The Process CPU Usage panel on the window displays a set of histograms. There is one histogram for each machine and the corresponding CAS server process. The histogram in the upper left pertains to the CAS controller node. If you are not an administrator, only the histogram for the CAS controller node is displayed.
   Each histogram displays the percentage of CPU usage, from 0 to 100%.

   rdcesx04222 7083
   ---------------------------------- 100

   Use these histograms to note patterns of CPU usage among the CAS nodes.

3  Select Add View ➔ CAS Process Metrics
   The CAS Process Metrics panel on the window displays a set of histograms. There is a set of three histograms for each machine and the corresponding CAS server process. The histogram in the upper left pertains to the CAS controller node. If you are not an administrator, only the histogram for the CAS controller node is displayed.
   Each set of histograms displays the percentage of CPU used, amount of resident memory used, and amount of virtual memory used for the CAS process.

   %CPU  Resident Memory Size  Virtual Memory Size
   rdcesx04222 7083
   ---------------------------------- 100%  ----------------- 1 GB  ----------------- 4 GB

4  Click to stop metric collection.
Monitor CAS Session Performance

1. In CAS Server Monitor, select on the left side of the window.

2. Select Add View ➔ Add Session View.

The panel for the session displays a set of histograms, with one histogram for each machine on the CAS cluster. If you are not an administrator, only the histogram for the CAS controller node is displayed. The top half of the histogram displays the percentage of CPU load that is used by the session. The bottom half displays the amount of resident memory that is used for the session.

```
rdcgrd001 8123 : 100%

1 GB
```

Monitor Host Performance

CAS Server Monitor displays histograms that enable you to view CPU load and memory usage for all machines in the CAS cluster. Follow these steps:

1. In CAS Server Monitor, select on the left side of the window.

2. To view CPU load, select Add View ➔ Host CPU Load Average.

The Host CPU Load Average panel on the window displays a set of histograms. There is one histogram for each machine in the CAS cluster. The histogram in the upper left pertains to the CAS controller node. If you are not an administrator, only the histogram for the CAS controller node is displayed.

Each histogram displays the CPU load on the machine, using the same format as the Linux xload command. Each division on the histograms represents one load average point. The highest point on each histogram is displayed to the right of the histogram.

```
rdcase04222
```

Use these histograms to note usage patterns among the CAS nodes. For example, if the load on a worker node machine is significantly and consistently higher than the load on other machines, use the Show Processes function to check for other running processes or defunct processes. See Monitor Process Information on page 4 for instructions about this function.

3. To view the memory usage, select Add View ➔ Host Memory Usage.

The Host Memory Usage panel on the window displays a set of histograms. There is one histogram for each machine in the CAS cluster. The histogram in the upper left pertains to the CAS controller node. If you are not an administrator, only the histogram for the CAS controller node is displayed.

Each histogram displays the percentage of memory that is used on the machine, from 0 to 100%. The percentage of memory used is displayed in green, at the top of the histogram. The percentage of virtual memory that is used is displayed in orange, at the bottom of the histogram.
Use these histograms to note patterns of memory usage among the CAS nodes. For example, if memory usage is consistently high on a machine, its memory capacity might need to be increased.

4 Click the Stop button to stop metric collection.

Monitor Process Information

1 Perform one of these actions in CAS Server Monitor:
   - Select on the left side of the window and open one of the views from the Add View or Add Session View menus. Click to the right of a histogram. Select Show Processes. This option is available only if you are an administrator.
   - Click and select the Grid Nodes tab. Click on the right side of a node’s row and select Show Processes.

2 The Processes window appears. The window displays this information:
   - Metrics for the selected node, including uptime, the number of processes, memory usage, CPU load, and file usage.
   - A histogram of the CPU load for the node.
   - A table containing the output from the top command for the selected node. The output includes metrics such as CPU usage, time, and the threads for each process. If you are the process owner, the window displays information about all processes. If you are not the process owner, you can view information only about your own processes.

   If you are the process owner, you can open a terminal window to terminate processes that are causing problems. See Open a Terminal Window on a Grid Node on page 4 for information.

Open a Terminal Window on a Grid Node

After using the monitoring functions of CAS Server Monitor to identify problems with the CAS nodes, you might want to use commands to end processes on a node. If you are the process owner, you can launch a terminal window to manage processes on the node. Follow these steps:

1 Perform one of these actions in CAS Server Monitor:
   - Select on the left side of the window and use the Add View menu to display the Host CPU Load Average, Host Memory Usage, or CAS Process CPU Usage views. Click on the right side of the histogram for a node. Select Launch Terminal. This action is available only if you are an administrator.
   - Click and select the Grid Nodes tab. Click on the right side of a node’s row and select Launch Terminal.

2 A terminal window appears on the selected machine. Use the window to manage processes on the machine.

3 Enter the exit command to close the terminal window.
Change the Monitoring Display Options

When you are viewing the histograms in the Grid Monitor view in CAS Server Monitor, you can control how the histograms are displayed.

- To change how quickly the graph data is refreshed, move the slider next to the **Speed** label. The default refresh rate is approximately 3.5 seconds. If you move the slider all the way to the left, the refresh rate is approximately 6.5 seconds. If you move the slider all the way to the right, the refresh rate is approximately 1.5 seconds.

- To change the size of the histograms, move the slider next to the **Size** label.

- The default layout for a histogram view is a grid. To change to a single column, click the **column icon** in the banner for a view. To return to a grid layout, click the **grid icon**.

To change the default view for the Grid Monitor view, select **user ID ⇒ Settings** in the upper right of the CAS Server Monitor window. You can select a default monitor view and a layout.

Using CAS Session Options

View Performance Metrics for a CAS Action

To view performance metrics data when you execute a CAS action, start the CAS server with the -metrics option, or set the cas.metrics configuration option to **true**.

To start displaying performance metrics for a running server, set the metrics session option to **true**. If you enable the collection of metrics, a standard set of data is returned to the log each time that a CAS action is completed. The same data is displayed by both the server and the client, but the names of the metrics are different. See Reference on page 6 for a list of the metrics that are displayed.

Here is an example of the metrics that are displayed for a CAS action:

```
NOTE: Executing action 'tkimstat.summary'
NOTE: Action 'tkimstat.summary' used (Total process time):
NOTE:       real time               0.024989 seconds
NOTE:       cpu time                0.165974 seconds (664.19%)
NOTE:       total nodes             4 (96 cores)
NOTE:       total memory            377.85G
NOTE:       memory                  22.53M (0.01%)
{   disposition = { ... },
    messages = { ... },
    results = { ... },
    performance = {
        elapsedTime = 0.024989,
        cpuUserTime = 0.132979,
        systemCores = 96,
        systemTotalMemory = 405711519744,
        cpuSystemTime = 0.032995,
        memoryOS = 45793280,
        memory = 23621664,
        memoryQuota = 47366144,
        systemNodes = 4,
```
Evaluate CPU Utilization for an Action

If you specify that performance metrics are collected when CAS actions are executed, you can use these metrics to evaluate the CPU utilization of your CAS environment.

The `cpu time` server metric is displayed in both the number of seconds and a percentage. Here is an example:

```
cpu time    0.165974 seconds (664.19%)
```

The percentage is calculated as \(\frac{\text{cpuUserTime} + \text{cpuSystemTime}}{\text{elapsedTime}}\). On a single-threaded system, the maximum value for this metric is 100%. However, for a multi-core system, the maximum value is \(100\% \times \text{number of cores}\). In this example, the system has 96 cores, so the maximum value is 9600%.

Reference

CAS Action Metrics

If you enable the collection of metrics for CAS actions, a standard set of data is returned each time that a CAS action is completed. The same data is displayed by both the server and the client. Here is the data that is displayed:

<table>
<thead>
<tr>
<th>Server Metric Name</th>
<th>Client Metric Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>real time</td>
<td>elapsedTime</td>
<td>The number of seconds in actual time that were used to run the action.</td>
</tr>
<tr>
<td>not supplied</td>
<td>cpuUserTime</td>
<td>The total number of seconds that were used by the action in user mode across all cores that were used to run the action.</td>
</tr>
<tr>
<td>not supplied</td>
<td>cpuSystemTime</td>
<td>The total number of seconds that were used by the action in system mode across all cores that were used to run the action.</td>
</tr>
<tr>
<td>cpu time</td>
<td>not supplied</td>
<td>CPU time is calculated and displayed in these formats:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{cpuUserTime} + \text{cpuSystemTime}), and is displayed in seconds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\frac{\text{cpuUserTime} + \text{cpuSystemTime}}{\text{elapsedTime}}), and is displayed as a percentage.</td>
</tr>
<tr>
<td>total nodes</td>
<td>systemNodes</td>
<td>The number of nodes in the cluster (total nodes displays both systemNodes and systemCores).</td>
</tr>
<tr>
<td>total nodes</td>
<td>systemCores</td>
<td>The number of cores in the cluster (total nodes displays both systemNodes and systemCores).</td>
</tr>
<tr>
<td>total memory</td>
<td>systemTotalMemory</td>
<td>The total memory that is available to the system (total memory is displayed in GB and systemTotalMemory is displayed in bytes).</td>
</tr>
<tr>
<td>memory</td>
<td>memory</td>
<td>The amount of memory that is used to execute the action.</td>
</tr>
<tr>
<td>Server Metric Name</td>
<td>Client Metric Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>not supplied</td>
<td>memoryOS</td>
<td>The amount of operating system memory that is used by the action.</td>
</tr>
<tr>
<td>not supplied</td>
<td>contextVoluntary</td>
<td>The number of times that a context switch occurred because a process released the processor before its time slice was completed.</td>
</tr>
<tr>
<td>not supplied</td>
<td>contextInvoluntary</td>
<td>The number of times that a context switch occurred because a process with a higher priority was present or because the current process exceeded its time slice.</td>
</tr>
<tr>
<td>not supplied</td>
<td>memoryQuota</td>
<td>The memory quota was used entirely by the action.</td>
</tr>
<tr>
<td>not supplied</td>
<td>dataMovementTime</td>
<td>The amount of time (in seconds) that was used to move the data between memory and the processors.</td>
</tr>
<tr>
<td>not supplied</td>
<td>dataMovementBytes</td>
<td>The number of bytes of data that were moved between memory and the processors.</td>
</tr>
</tbody>
</table>

See View Performance Metrics for a CAS Action on page 5 for information about displaying these metrics.