

## Charting Time-Series Data

Cloudera Manager enables you to enter a query for a time series, chart the time-series data, view time series if your query produced multiple time series, and save the results as a dashboard. The following sections have more details on the terminology used, how to query for time-series data, details, editing charts, and modifying chart properties.

### Terminology

The list below describes the terminology used when creating Charts.

#### Entity

A Cloudera Manager component that has metrics associated with it, such as a service, role, or cluster. **Metric**

A property that can be measured to quantify the state of an entity or activity, such as the number of open file descriptors or CPU utilization percentage. For a list of all the metrics supported by Cloudera Manager, see the Cloudera Manager Metrics reference documentation.

#### Time Series

A list of (time, value) pairs that is associated with some (entity, metric) pair such as, (date, fd\_open), (hostname, cpu\_percent). In more complex cases, the time series can represent operations on other time series. For example, (datanode-1 , cpu\_user + cpu\_system).

#### Facet

A display grouping of a set of time series. By default, when a query returns multiple time series, they are displayed in individual charts. Facets allow you to display the time series in separate charts in a single chart, or grouped by various attributes of the set of time series.

#### Related Information

Cloudera Manager Metrics

## Building a Chart with Time-Series Data

Use the Chart Builder to build a chart with time-series data.

### Procedure

1. Select Charts Chart Builder.
2. Display time series in one of the following ways.
  - Select a recently used statement: Click the to the right of the Build Chart button to display a list of recently run statements and select a statement. The statement text displays in the text box and the chart(s) that display that time series will display.
  - Select from the list of Chart Examples:
    - a. Click the question mark icon to the right of the Build Chart button to display a list of examples with descriptions.
    - b. Click Try it to create a chart based on the statement text in the example.
  - Type a new statement: Press Spacebar in the text box. tsquery statement components display in a drop-down list. These suggestions are part of type ahead, which helps build valid queries. Scroll to the desired component

and click Enter. Continue choosing query components by pressing Spacebar and Enter until the statement is complete.

#### Example

For example, the query `SELECT jvm_heap_used_mb where clusterId = 1` could return a set of charts as shown in the following:

### Configuring Time-Series Query Results

A time-series query returns one or more time series or scalar values. By default a maximum of 1000 series are returned.

Before you begin

Minimum Required Role: Configurator (also provided by Cluster Administrator, Limited Cluster Administrator, Full Administrator)

#### Procedure

1. To change this value, select Administration Settings.

Cloudera Manager Charting Time-Series Data

2. In the Advanced category, set the Maximum Number Of Time-Series Streams Returned Per Time-Series Query or the Maximum Number of Time-Series Streams Returned Per Heatmap property.
3. Click Save Changes.

### Using Context-Sensitive Variables in Charts

When editing charts from a service, role or host status or charts page, or when adding a chart to a dashboard, context-sensitive variables (each beginning with '\$') will be displayed below the query box. For example, you might see variables similar to those in the query below:

Notice the `$HOSTNAME` portion of the query string. `$HOSTNAME` is a variable that will be resolved to the host's name value based on the page before the query is actually issued. In this case, `$HOSTNAME` will be `cloudera.com`.

The chart below shows an example of the output of a similar query.

Context-sensitive variables are useful since they allow portable queries to be written. For example, you may be on the host status page or any role status page to display the appropriate host's swap space. These variables are used in queries that are part of user-defined dashboards since those dashboards have no service context.

### Chart Properties

By default, the time-series data retrieved by the `tsquery` is displayed on its own chart, using a default size, and a default minimum and maximum for the Y-axis. You can change the chart type, change the chart scale and size, and set X- and Y-axis ranges.

Cloudera Manager Charting Time-Series Data

#### Changing Scale

You can set the scale of the chart to linear, logarithmic, and power.

### Changing Dimensions

You can change the size of your charts by modifying the values in the Dimension fields. They increment when you click the up or down arrows, and you can type values in as long as they are valid. If you have multiple charts, depending on the dimensions you specify and the size of your browser, charts may appear in rows of multiple charts. If the Resize Proportionally checkbox is checked, the width and height of the chart will be modified automatically to maintain the chart's width and height. The following chart shows the same query as the previous chart, but with All Combined selected (to show all time series in a single chart) and with the Dimension values increased to expand the chart.

### Changing Axes

You can change the Y-axis range using the Y Range minimum and maximum fields.

The X-axis is based on clock time, and by default shows the last hour of data. You can use the duration link ( ) to set the time range.

### Changing the Chart Type

You can change a chart from one type to another.

#### Procedure

1. Click ChartsChart Builder to access the chart, or click the chart on the Cloudera Manager home page and then View Entity Chart.
2. Click one of the chart types:
  - Line - Displays the points in the time series as continuous line.
  - Stack Area - Displays the points in the time series as continuous line and the area under the line filled in.
  - Bar - Displays each the value of the metric averaged over a second as a bar.
  - Scatter - Displays the points in the time series as dots.
  - Heatmap - Displays a metric thermometer and grid of colored squares. The thermometer displays buckets that represent a range of metric values and a color coding for the bucket. Each square represents an entity and the color of the square represents the value of a metric within a range. The following heatmap shows the last value of the resident memory for the NodeManager, ImpalaD, DataNode, and RegionServer roles.
  - Histogram - Displays the time series values as a set of bars where each bar represents a range of metric values and the height of the bar represents the number of entities whose value falls within the range. The following histogram shows the number of roles in each range of the last value of the resident memory.
  - Table - Displays the time series values as a table with each row containing the data for a single time value. Note: Heatmaps and histograms render charts for a single point as opposed to time series charts that render a series of points. For queries that return time series, Cloudera Manager will generate the heatmap or histogram based on the last recorded point in the series, and will issue the warning:

“Query returned more than one value per stream. Only the last value was used.” To eliminate this warning, use a scalar returning function to choose a point. For example, use `select last(cpu_percent)` to use the last point or `select max(cpu_percent)` to use the maximum value (in the selected time range).

## Grouping (Faceting) Time Series

A time-series plot for a service, role, or host may actually be a composite of multiple individual time series. If you have many facets, you can combine time series based their attributes.

For example, the query `SELECT jvm_heap_used_mb where clusterId = 1` returns time-series data for JVM heap used. Each time series has hostname, role type, metric, and entity name attributes. By default, all time series are displayed all on a single chart.

To change the organization of the chart data, click one of the facets in the facet section of the chart. The number in parentheses indicates how many charts will be displayed for that facet. As shown in Figure 1, if the `serviceName` facet is selected for the JVM heap query, the time series is grouped into multiple charts, one for each service name. The charts for service types with multiple roles contain multiple lines (for example, HDFS) while services that have only one role (for example, ZooKeeper) contain just a single line. If a chart contains multiple lines, each entity is identified by a different color line.

## Displaying Chart Details

You can interact with a chart to display various chart details.

When you move your mouse over a chart, its background turns gray, indicating that you can access details about the chart.

- Moving the mouse to a data point on a line, stack area, or bar chart shows the details about that data point in a pop-up tooltip.
- Click a line, stack area, scatter, or bar chart to expand it into a full-page view with a legend for the individual charted entities as well more fine-grained axes divisions.
  - If there are multiple entities in the chart, you can
    - \* Check and uncheck the legend item to hide or show the time series for the entities on the chart.
    - \* If there are service, role, or host instances in the chart, click the View link to display the instance’s Status page.
  - Click the Close button to return to the regular chart view.
- Heatmap - Clicking a square in a heatmap displays a line chart of the time series for that entity.
- Histogram -
  - Mousing over the upper right corner of a histogram and clicking opens a pop-up containing the query that generated the chart, an expanded view of the chart, a list of entity names and links to the entities whose metrics

## Cloudera Manager Charting Time-Series Data

are represented by the histogram bars, and the value of the metric for each entity. For example, the following histogram

displays the following:

- Clicking a bar in the expanded histogram displays a line chart of the time series from which the histogram was generated:

Clicking the < Back link at the bottom left of the line chart returns to the expanded histogram.

## Editing a Chart

You can edit a chart from the custom dashboard and save it back into the same or another existing custom dashboard.

About this task

Editing a chart only affects the copy of the chart in the current dashboard - if you have copied the chart into other dashboards, those charts are not affected by your edits.

Procedure

1. Move the cursor over the chart, and click the gear icon at the top right.
2. Click Open in Chart Builder. This opens the Chart Builder page with the chart you selected already displayed.
3. Edit the chart's select statement and click Build Chart. Related Information Dashboard Types

## Saving a Chart

After you edit a chart, you can save it to a new or existing custom dashboard.

Before you begin

Minimum Required Role: Configurator (also provided by Cluster Administrator, Limited Cluster Administrator)

Users with Viewer, Limited Operator, or Operator user roles can edit charts and view the results of their edits. They can also save charts to a dashboard.

Procedure

1. Modify the chart's properties and click Build Chart.
2. Click Save to open the Save Chart dialog box, and select one of the following:
  - Update chart in current dashboard: .
  - Add chart to another dashboard.
  - Add chart to a new custom dashboard.
3. Click Save Chart.
4. Click View Dashboard to go to the dashboard where the chart has been saved. Saving a chart only affects the copy of the chart in the dashboard where you save it - if you have previously copied the chart into other dashboards, those charts are not affected by your edits.

## Obtaining Time-Series Data Using the API

You can obtain time-series data using the Cloudera Manager API. For details about using a ts obtain time-series data, see the /timeseries API documentation at [http:// cmServerHost:7180/timeseries.html](http://cmServerHost:7180/timeseries.html).

### Procedure

1. Click ChartsChart Builder to access the chart, or click the chart on the Cloudera Manager home page and then click View Entity Chart.
2. To see the API call that returns the time-series data for an existing chart, click the blue down-arrow at the upper- right corner of the chart and click Export JSON. A new web browser window opens, displaying the time-series data in JSON format. The query string of the URL for that window displays the API call that retrieved the time-series data.

## Dashboards

A dashboard is a set of charts. This section covers creating, configuring, and managing dashboards. For more information, see [Related Information](#).

### Charting Time-Series Data

## Dashboard Types

A default dashboard is a predefined set of charts that you cannot change. In a default dashboard you can:

- Display chart details.
- Edit a chart and then save back to a new or existing custom dashboard. A custom dashboard contains a set of charts that you can change. In a custom dashboard you can:
  - Display chart details.
  - Edit a chart and then save back to a new or existing custom dashboard.
  - Save a chart, make any modifications, and then save to a new or existing dashboard.
  - Remove a chart. When you first display a page containing charts it has a custom dashboard with the same charts as a default dashboard.

## Creating a Dashboard

When you create a dashboard, you specify a name and optionally a duration.

### Procedure

1. Do one of the following:
  - Select Charts New Dashboard.
  - Select Charts Manage Dashboards and click Create Dashboard.
  - Save a chart to a new dashboard.
2. Specify a name and optionally a duration. Conventional dashboard names follow the patterns given by one of the following Java regular expressions:

```
Pattern.compile("^(.+):(\\d):" + STATUS_VIEW_SUFFIX + "$");
```

```
Pattern.compile("^(.+):" + STATUS_VIEW_SUFFIX + "$");
```

Examples of expected names are: HOST:STATUS\_VIEW, MGMT:STATUS\_VIEW, or HDFS:5:STATUS\_VIEW. If the dashboard name does not match the expected pattern, a warning will be displayed.

3. Click Create Dashboard.

## Managing Dashboards

You can create, clone, edit, export, import, and remove dashboards. To manage dashboards, select Charts Manage Dashboards.

- Create Dashboard - create a new dashboard.
- Clone - clones an existing dashboard.
- Edit - edit an existing dashboard.
- Export - exports the specifications for the dashboard as a JSON file.
- Import Dashboard - reads an exported JSON file and recreates the dashboard.
- Remove - deletes the dashboard.

## Configuring Dashboards

You can change the time scale of a dashboard, switch between default and custom dashboards, and delete a dashboard.

### Setting the Time Scale of a Dashboard

By default the time scale of a dashboard is 30 minutes. To change the time scale, click a dropdown menu at the top-right of the dashboard.

### Setting the Dashboard Type

To set the dashboard type, click and select one of the following:

- Custom - displays a custom dashboard.
- Default - displays a default dashboard.
- Reset - resets the custom dashboard to the predefined set of charts, discarding any customizations.

## Saving Charts to Dashboards

You can save the charts and their configurations (type, dimension, and y-axis minimum and maximum) to a new dashboard or to an existing dashboard.

Minimum Required Role: Configurator (also provided by Cluster Administrator, Limited Cluster Administrator, and Full Administrator)

If your tsquery statement resulted in multiple charts, those charts are saved as a unit (either a new dashboard or an existing dashboard). You cannot edit the individual plots in that set of charts, but you can edit the set. An edit button appears for the set that you saved - typically on the last chart in the set.

You can edit a copy of the individual charts in the set, but the edited copy does not change the original set.

dashboard from which it was copied.

### **Saving Charts to a New Dashboard**

You can save new or existing charts to a new dashboard.

#### **Procedure**

1. Optionally, modify the chart properties.
2. If the chart was created with the Chart Builder, optionally type a name for the chart in the Title field.
3. Do one of the following:
  - New chart: Click Save.
  - Existing chart: Move the cursor over the chart, and click the icon at the top right.
4. Optionally, edit the chart name.
5. Select the Add chart to a new custom dashboard option.
6. Enter a dashboard name.
7. Click Save Chart. The new dashboard appears on the menu under the top-level Charts tab. ##### Saving Charts to an Existing Dashboard

You can save new or existing charts to an existing dashboard.

#### **Procedure**

1. Optionally, modify the chart properties.
2. If the chart was created with the Chart Builder, optionally type a name for the chart in the Title field.
3. Do one of the following:
  - New chart: Click Save.
  - Existing chart: Move the cursor over the chart, and click the icon at the top right.
4. Optionally, edit the chart name.
5. Select the Add chart to an existing custom or system dashboard option.
6. Select a dashboard from the Dashboard Name drop-down list.
7. Click Save Chart. The chart is added (appended) to the dashboard you select.

### **Adding a New Chart to the Custom Dashboard**

You can add new charts to the custom dashboard on the Status tab of a service, host, or role.

#### **Procedure**

1. Click and select one of the following:
  - Add From Charts Library: Displays the charts page.
    - Select one or more charts.



- Add From Chart Builder: Displays the Add Chart To Dashboard page, with variables preset for the specific service, role, or host where you want to add the dashboard.
  - a. Click the question mark icon to the right of the Build Chart button and select a metric from the List of Metrics, type a metric name or description into the Basic text field, or type a query into the Advanced field.
  - b. Click Build Chart. The charts that result from your query are displayed, and you can modify their chart type, combine them using facets, change their size and so on.
- 2. Click Add. Note: If the query you've chosen has resulted in multiple charts, all the charts are added to the dashboard as a set. Although the individual charts in this set can be copied, you can only edit the set as a whole.

### Removing a Chart from a Custom Dashboard

You can remove a chart from a custom dashboard.

#### Procedure

1. Move the cursor over the chart and click the icon at the top right.
2. Click Remove.

### Moving and Resizing Charts

You can move or resize the charts on a dashboard, the cluster home page, or a service page.

#### Cloudera Manager Charting Time-Series Data

#### Procedure

1. Click the Edit Layout link. The dashboard changes to an editing view where each chart's display is replaced with a placeholder box that is the same size as the actual chart.
2. Change the size of a chart by dragging the lower-right corner of the chart.
3. Drag charts to change their relative positions.
4. Click Finish Editing to save the new layout.

### tsquery Language

The tsquery language is used to specify statements for retrieving time-series data from the series datastore.

Below are some common queries using the tsquery language:

- Retrieve time series for all metrics for all DataNodes:

```
select * where roleType=DATANODE
```

- Retrieve cpu\_user\_rate metric time series for all DataNodes:

```
select cpu_user_rate where roleType=DATANODE
```

- Retrieve the jvm\_heap\_used\_mb metric time series divided by 1024 and the jvm\_heap\_committed metric time series divided by 1024 for all roles running on the host named “my host”:

```
select jvm_heap_used_mb/1024, jvm_heap_committed_mb/1024 where category=
ROLE and hostname="my host"
```

- Retrieve the jvm\_total\_threads and jvm\_blocked\_threads metric time series for all entities for which Cloudera Manager collects these two metrics:

```
select jvm_total_threads,jvm_blocked_threads
```

#### Collections

A collection is a type of data. The field can take the value:

- ENTITY\_DATA -
- IMPALA\_QUERIES -
- IMPALA\_QUERY\_DETAILS -
- YARN\_APPLICATIONS -

#### tsquery Syntax

A tsquery statement has a specific structure.

```
SELECT [metric expression]FROM collection WHERE [predicate]
```

Note the following properties of tsquery statements:

- The statement select \* is invalid.
- Tokens are case insensitive. For example, Select, select, and SeLeCt are all equivalent to SELECT.
- Multiple statements can be concatenated with semi-colons. Thus example 3 in the tsquery Language topic can be written as:

```
select jvm_heap_used_mb/1024 where category=ROLE and hostname=myhost; se
lect jvm_heap_committed_mb/1024 where category=ROLE and hostname=myhost
```

- The metric expression can be replaced with an asterisk (\*), as shown in example 1 of the tsquery Language topic. In that case, all metrics that are applicable for selected entities, such as DATANODE in example 1, are returned.
- The collection can be omitted.
- The predicate can be omitted, as shown in example 4 of the tsquery Language topic. In such cases, time series for all entities for which the metrics are appropriate are returned. For this query you would see the jvm\_new\_threads metric for NameNodes, DataNodes, TaskTrackers, and so on. Related Information

#### Metric Expressions

Predicates

## Metric Expressions

A metric expression generates the time series. It is a comma-delimited list of one or more metric expression statements.

A metric expression statement is the name of a metric, a metric expression function, or a scalar expression or more metric expression operators.

See the FAQ which answers questions concerning how to discover metrics and use cases for scalars. For a list of all the supported metrics, see the Cloudera Manager Metrics reference documentation.

Metric expressions support the binary operators: +, -, \*, /.

Here are some examples of metric expressions:

- `jvm_heap_used_mb, cpu_user, 5`
- `1000 * jvm_gc_time_ms / jvm_gc_count`
- `total_cpu_user + total_cpu_system`
- `max(total_cpu_user)`

## Metric Expression Functions

### FAQ

Cloudera Manager Metrics

## Metric Expression Functions

Metric expressions support the functions listed in the following table. A function can return a scalar or a time series. Functions that return scalars must be used for heatmap charts.

Function Returns

Scalar?

Description

`avg(metric expression)` N Computes a simple average for a time series.

`count_service_roles()` Y Returns the number of roles. There are three variants of this function.

- `count_service_roles(roleType, roleState)` - Returns the number of roles of the specified roleType and roleState. For example, `count_service_roles(datanode, running)` returns the number of running DataNodes.
- `count_service_roles(roleType)` - Returns the number of roles with the specified role Type.
- `count_service_roles()` - Return the number of roles. For example, `select events_critical where count_service_roles() > 100` returns the event\_critical metric when the number of roles is greater than 100.

Cloudera Manager Charting Time-Series Data

Function Returns

Scalar?

## Description

`dt(metric expression)` N Derivative with negative values. The change of the underlying metric expression per second. For example: `dt(jvm_gc_count)`.

`dt0(metric expression)` N Derivative where negative values are skipped (useful for dealing with metrics that only increase). The change of the underlying metric expression per second. For example: `dt0(jvm_gc_time_ms) / 10`.

`getClusterFact(string factName,  
double defaultValue)`

Y Retrieves a fact about a cluster. Currently supports one fact: `numCores`. If the number of cores cannot be determined, `defaultValue` is returned.

`getHostFact(string factName, double  
defaultValue)`

Y Retrieves a fact about a host. Currently supports one fact: `numCores`. If the number of cores cannot be determined, `defaultValue` is returned.

For example, `select dt(total_cpu_user) / getHostFact(numCores, 2) where category=HOST` divides the results of `dt(total_cpu_user)` by the current number of cores for each host. The following query computes the percentage of total user and system CPU usage each role is using on the host. It first computes the CPU seconds per second for the number of cores used by taking the derivative of the total user and system CPU times. It normalizes the results to the number of cores on the host by using the `getHostFact` function and multiplies the result by 100 to get the percentage.

```
select dt0(total_cpu_user)/getHostFact(numCores,1)*100,  
dt0(total_cpu_system)/getHostFact(numCores,1)*100  
where category=ROLE and clusterId=1
```

`greatest(metric expression, scalar  
metric expression)`

`N` Compares two metric expressions, one of which one is a scalar metric expression. Returns a time series where each point is the result of evaluating `max(point, scalar metric expression)`.  
`integral(metric expression)` N Computes the integral value for a stream and returns a time-series data point is the integral value of the corresponding data point from the original stream. For example, `select integral(maps_failed_rate)` will return the count of the failed number of maps.

`counter_delta(metric expression)` N Computes the difference in counter value for a stream and within which each data point is the difference in counter value of the corresponding data point from the counter value of previous data point in the original stream. For example: `select counter_delta(maps_failed_rate)` returns the count of the failed number of maps. This method is more accurate than the `integral()` function. However there are a few caveats:

- This function is only implemented for single time-series streams. For streams of cross-entity aggregates, continue to use the `integral()` function.

- If you apply this method for time-series streams which was created using a version of Cloudera Manager older than 5.7, Cloudera Manager fills in the older data points using the `integral()` function. `last(metric expression)` Y Returns the last point of a time series. For example, to use the last point of the `cpu_percent` metric time series, use the expression `select last(cpu_percent). least(metric expression, scalar metric expression)`

`N` Compares two metric expressions, of which one is a scalar metric expression. Returns a time series where each point is the result of evaluating `min(point, scalar metric expression)`. `max(metric expression)` Y Computes the maximum value of the time series. For example, `select max(metric expression)` Y Computes the minimum value of the time series. `moving_avg(metric expression, time_window_sec)`

`N` Computes the moving average for a time series over a time window `time_window_sec` specified in seconds (2, 0.1, and so on)

`stats(metric expression, stats name)` N Some time-series streams have additional statistics for time-series streams, cross-entity aggregates, and rate metrics. The following statistics are available for rollup and cross-entity aggregates: `max`, `min`, `avg`, `std_dev`, and `sample`. For rate metrics, the underlying counter value is available using the "counter" statistics. For example, `stats(fd_open_across_datanodes, max)` or `stats(swap_out_rate, counter)`. `sum(metric expression)` Y Computes the sum value of the time-series.

## Predicates

A predicate limits the number of streams in the returned series.

A predicate can take one of the following forms:

- `time_series_attribute operator value`, where
  - `time_series_attribute` is one of the supported attributes.
  - `operator` is one of `=` and `rlike`
  - `value` is an attribute value subject to the following constraints:
    - \* For attributes values that contain spaces or values of attributes of the form `xxxName` such as `displayName`, use quoted strings.
    - \* The value for the `rlike` operator must be specified in quotes. For example: `hostname rlike "host[0-3]+.*"`.
    - \* `value` can be any regular expression as specified in regular expression constructs in the Java Pattern class documentation.
- `scalar_producing_function(metric_expression) comparator number` , where
  - `scalar_producing_function` is any function that takes a time series and produces a scalar. For example, `min` or `max`.
  - `metric_expression` is a valid metric expression. For example, `total_cpu_user + total_cpu_system`.
  - `comparator` is a comparison operator: `<`, `<=`, `=`, `!=`, `>=`, `>`.
  - `number` is any number expression or a number expression with units. For example, `5`, `5mb`, `5s` are all valid number expressions. The valid

units are:

- \* Time - ms (milliseconds), s (seconds), m (minutes), h (hours), and d (days).
- \* Bytes -b (bytes), kb or kib (kilobytes), mb or mib (megabytes), gb or gib (gigabytes), tb or tib (terabytes), and pb or pib (petabytes)
- \* Bytes per second - Bytes and Time: bps, kbps, kibps, mbps, mibps, and so on. For example, 5 kilobytes per second is 5 kbps.
- \* Bytes time - Bytes and Time combined: bms, bs, bm, bh, bd, kms, ks, and so on. For example, 5 kilobytes seconds is 5 ks or 5 kis. You use the AND and OR operators to compose compound predicates. Example Statements with Compound Predicates

1. Retrieve all time series for all metrics for DataNodes or TaskTrackers.

```
select * where roleType=DATANODE or roleType=TASKTRACKER
```

2. Retrieve all time series for all metrics for DataNodes or TaskTrackers that are running on host named "myhost".

```
select * where (roleType=DATANODE or roleType=TASKTRACKER) and hostname=
myhost
```

3. Retrieve the total\_cpu\_user metric time series for all hosts with names that match the regular expression "host[0-3]+.\*"

```
select total_cpu_user where category=role and hostname rlike "host[0-3]+
.*"
```

#### Example Statements with Predicates with Scalar Producing Functions

1. Return the entities where the last count of Java VM garbage collections was greater than 10:

```
select jvm_gc_count where last(jvm_gc_count) > 10
```

2. Return the number of open file descriptors where processes have more than 500Mb of mem\_rss:

```
select fd_open where min(mem_rss) > 500Mb
```

#### Discovering possible predicates

You can use Predicates ('where' clauses) to specify filters on a wide range of aspects when some cases, this large selection makes it challenging to choose the right filters for the in describes a way to find out more about applicable filters and values in the context of a spe In a tsquery, the main way to filter results is by defining conditions on time series entity set of applicable attributes are determined by the entity type(s) being queried. The Chartin describes this data model in more detail.

You can find more information in the following sections of the tsquery Language topic:

- Predicates: explains what can be used in ‘where’ clauses, including and in addition to attributes.
- Time Series Attributes: a subset of the attributes you can use, with a description for each.
- Time Series Entities and their Attributes: a subset of the time series entity types available, with the applicable attributes listed. Important: The tables of entities and attributes cannot be complete for any given cluster, as service integrations (including third-party ones) can extend both. To show the attributes applicable to a specific query, along with the values they can have, on a running cluster, you can enter it into Chart Builder without the ‘where’ clause. The chart facets displayed are the attributes, while selecting a facet shows one chart for each currently valid value. For more information, see [Charting Time-Series Data](#) and the following figures:

In addition to the above approaches, you can also use the Cloudera Manager API to request the data from a running cluster as follows:

- Mapping of entity types to the metrics they have, as a JSON document: GET /timeseries/schema (The response is large, approximately 50 MB.) This further clarifies the relationship between entity types, attributes, and metrics, while providing insight into the metrics applicable to each entity type.
- A detailed explanation of attributes (similar to the Time Series Attributes table): GET /timeseries/entityTypeAttributes
- Definitions of the entity types referred to in the schema document, along with applicable attribute names (similar to the Time Series Entities and their Attributes table): GET /timeseries/entityTypes The results obtained from the API include all entity types and attributes tracked by the running cluster, including those added by service integrations.

### Filtering by Day of Week or Hour of Day

You can add an expression to the predicate of a tsquery statement that limits the stream to a specific day of the week or to a range of hours in each day.

**By Day** - Limits the stream to selected days of the week.

The day in () expression takes an argument with a comma-separated list of days of the week, The days of the week are numbered 1 through 7; 1 = Monday, 2 = Tuesday, and so on. Use the following expression to limit the stream to events that occurred only on Wednesday:

For example, the following expression limits the stream to events that occurred only on Wednesday:

**By Hour** - Limits the stream to a range of hours each day.

The hour in expression takes an argument with a range of hours separated by a colon and enclosed in brackets. Valid values are integers 0-23:

hour in [#: #]

For example, the following expression limits the stream to events that occur only between 9:

hour in [9:17]

Add the day or time range expression after the WHERE clause. Do not use the AND keyword. For

```
select fd_open where category = ROLE and roleType = SERVICEMONITOR day in (1
,2,3,4,5)
```

You can also combine day in and hour in expressions. Always put the day expression before the  
The following example limits the stream to weekdays between 9:00 a.m. and 5:00 p.m.:

```
select fd_open where category = ROLE and roleType = SERVICEMONITOR day in (1
,2,3,4,5) hour in [9:17]
```

## Time Series Attributes

You can use time series attributes when you build a predicate.

Attribute names and most attribute values are case insensitive. displayName and serviceType  
values are case sensitive.

### Name Description

active Indicates whether the entities to be retrieved must be active. A nonactive entity is  
removed or deleted from the cluster. The default is to retrieve only active entities (that is,  
access time series for deleted or removed entities, specify active=false in the query. For example,

```
SELECT fd_open WHERE roleType=DATANODE and active=false
```

agentName A Flume agent name.

applicationName One of the Cloudera Manager monitoring daemon names.

cacheId The HDFS cache directive ID.

category The category of the entities returned by the query: CLUSTER, DIRECTORY, DISK, FILESYSTEM,  
FLUME\_SOURCE, FLUME\_CHANNEL, FLUME\_SINK, HOST, HTABLE, IMPALA\_QUERY\_STREAM, JOURNAL,  
EAM, NETWORK\_INTERFACE, ROLE, SERVICE, USER, YARN\_APPLICATION\_STREAM, YARN\_APPLICATION,  
\_QUEUE.

Some metrics are collected for more than one type of entity. For example, total\_cpu\_user is  
entities of category HOST and ROLE. To retrieve the data only for hosts use:

```
select total_cpu_user where category=HOST
```

The ROLE category applies to all role types (see roleType attribute). The SERVICE category applies to  
service types (see serviceType attribute). For example, to retrieve the committed heap for a  
use:

```
select jvm_committed_heap_mb where category=ROLE and hostname="host1"
```

clusterDisplayName The user-defined display name of a cluster.

clusterName The cluster ID. To specify the cluster by its display name, use the clusterDisplayName

componentName A Flume component name. For example, channel1, sink1.



device A disk device name. For example, sda.

entityName A display name plus unique identifier. For example: HDFS-1-DATANODE-692d141f436cbe83f887.

expired A Boolean that indicates whether an HDFS cache directive expired.

groupName A user group name.

hbaseNamespace The name of the HBase namespace.

hostId The canonical identifier for a host in Cloudera Manager. It is unique and immutable. 5222-2f7e-4895-ae51-cd43b91f1e7a.

hostname A hostname.

hregionName The HBase region name. For example, 4cd887662e5c2f3cd5dd227bb03dd760.

hregionStartTimeMs Milliseconds from UNIX epoch since Cloudera Manager monitoring started on HBase region.

Name Description

tableName The name of an HBase table.

iface A network interface name. For example, eth0.

logicalPartition A Boolean indicating whether or not the disk is a logical partition. Applies to partition entity types.

mountpoint A mount point name. For example, /var, /mnt/homes.

nameserviceName The name of the HDFS nameservice.

ownerName The owner username.

partition A partition name. Applies to partition entity types.

path A filesystem path associated with the time-series entity.

poolName A pool name. For example, hdfs cache pool, yarn pools.

queueName The name of a YARN queue.

rackId A Rack ID. For example, /default.

roleConfigGroup The role group that a role belongs to.

roleName The role ID. For example, HBASE-1-REGIONSERVER-0b0ad09537621923e2b460e5495569e7.

roleState The role state: BUSY, HISTORY\_NOT\_AVAILABLE, NA, RUNNING, STARTING, STOPPED, STOPPING, UNKNOWN

roleType The role type: ACTIVITYMONITOR, AGENT, ALERTPUBLISHER, BEESWAX\_SERVER, CATALOGSERVER, DATANODE, EVENTSERVER, FAILOVERCONTROLLER, HBASE\_INDEXER, HBASE\_EREESTSERVER, HBASETHRIFTSERVER, HIVEMETASTORE, HIVESERVER2, HOSTMONITOR, HTTPFS, HUESERVER, IMPALAD, JOBHISTORY, JOBTRACKER, JOURNALNODE, KT\_RENEWER, LLAMA, MASTER, NAVIGATOR, REGIONSERVER, SERVICEMONITOR, NAMENODE, NODEMANAGER, REPORTSMANAGER, SECONDARYNAMENODE, SERVER, SOLR\_SERVER, SQOOP\_SERVER, STATESTORE, TASKTRACKER.

rollup The time-series store table rollup type.

schedulerType The scheduler type associated with the pool service.

serviceDisplayName The user-defined display name of a service entity.

serviceName The service ID. To specify a service by its display name use the serviceDisplayName.

serviceState The service state: HISTORY\_NOT\_AVAILABLE, NA, RUNNING, STARTING, STOPPED, STOPPING, UNKNOWN

serviceType The service type: ACCUMULO, FLUME, HDFS, HBASE, HIVE, HUE, IMPALA, KS\_INDEXER, MAEDUCE, MGMT, OOZIE, SOLR, SPARK, SQOOP, YARN, ZOOKEEPER.

solrCollectionName The Solr collection name. For example, my\_collection.

solrReplicaName The Solr replica name. For example, my\_collection\_shard1\_replica1.

solrShardName The Solr shard name. For example, shard1.  
systemTable A boolean indicating whether the HBase table is a system table or not.  
tableName The name of a table.  
userName The name of the user.  
version The version of the cluster. The value can be any of the supported Cloudera Runtime m

## Time Series Entities and their Attributes

The following table shows the entities and associated attributes that can appear in the prece  
tsquery statement.

### Entity Attributes

All Roles roleType, hostId, hostname, rackId, serviceType, serviceName  
All Services serviceName, serviceType, clusterId, version, serviceDisplayName, clusterDisplay

### Cloudera Manager Charting Time-Series Data

### Entity Attributes

Agent roleType, hostId, hostname, rackId, serviceType, serviceName, clusterId, version, age  
serviceDisplayName, clusterDisplayName  
Cluster clusterId, version, clusterDisplayName  
Directory roleName, hostId, path, roleType, hostname, rackId, serviceType, serviceName, clus  
agentName, hostname, clusterDisplayName  
Disk device, logicalPartition, hostId, rackId, clusterId, version, hostname, clusterDisplay  
File System hostId, mountpoint, rackId, clusterId, version, partition, hostname, clusterDis  
Flume Channel serviceName, hostId, rackId, roleName, flumeComponent, roleType, serviceType,  
agentName, serviceDisplayName, clusterDisplayName  
Flume Sink serviceName, hostId, rackId, roleName, flumeComponent, roleType, serviceType, cl  
agentName, serviceDisplayName, clusterDisplayName  
Flume Source serviceName, hostId, rackId, roleName, flumeComponent, roleType, serviceType, c  
agentName, serviceDisplayName, clusterDisplayName  
HDFS Cache Pool serviceName, poolName, nameserviceName, serviceType, clusterId, version, gro  
serviceDisplayName, clusterDisplayName  
HNamespaces serviceName, namespaceName, serviceType, clusterId, version, serviceDisplayName,  
Host hostId, rackId, clusterId, version, hostname, clusterDisplayName  
HRegion htableName, hregionName, hregionStartTimeMs, namespaceName, serviceName, tableName,  
serviceType, clusterId, version, roleType, hostname, roleName, hostId, rackId , serviceDispl  
clusterDisplayName  
HTable namespaceName, serviceName, tableName, serviceType, clusterId, version, serviceDispla  
clusterDisplayName  
Network Interface hostId, networkInterface, rackId, clusterId, version, hostname, clusterDis  
Rack rackId  
Service serviceName, serviceType, clusterId, serviceDisplayName  
Solr Collection serviceName, serviceType, clusterId, version, serviceDisplayName, clusterDis  
Solr Replica serviceName, solrShardName, solrReplicaName, solrCollectionName, serviceType, c  
roleType, hostId, hostname, rackId, roleName, serviceDisplayName, clusterDisplayName  
Solr Shard serviceName, solrCollectionName, solrShardName, serviceType, clusterId, version,

clusterDisplayName  
Time Series Table tableName, roleName, roleType, applicationName, rollup, path  
User userName  
YARN Pool serviceName, queueName, schedulerType

## FAQ

The following topic covers frequently asked questions about charting time-series data.  
How do I compare information across hosts?

1. Click Hosts in the top navigation bar and click a host link.
2. In the Charts pane, choose a chart, for example Host CPU Usage and select and then Open in Chart Builder.
3. In the text box, remove the where entityName=\$HOSTID clause and click Build Chart.
4. In the Facets list, click hostname to compare the values across hosts.
5. Configure the time scale, minimums and maximums, and dimension. For example:

How do I compare all disk IO for all the DataNodes that belong to a specific HDFS service?  
Use a query of the form:

```
select bytes_read, bytes_written where roleType=DATANODE and serviceName=hdfs1
```

replacing hdfs1 with your HDFS service name. Then facet by metricDisplayName and compare all DataNode byte\_reads and byte\_writes metrics at once.

When would I use a derivative function?

Some metrics represent a counter, for example, bytes\_read. For such metrics it is sometimes to see the rate of change instead of the absolute counter value. Use dt or dt0 derivative function.

When should I use the dt0 function?

Some metrics, like bytes\_read represent a counter that always grows. For such metrics a negative rate means that the counter has been reset (for example, process restarted, host restarted, Use dt0 for these metrics.

How do I display a threshold on a chart?

Suppose that you want to retrieve the latencies for all disks on your hosts, compare them, and set a threshold on the chart to easily detect outliers. Use the following query to retrieve the data and the threshold:

```
select await_time, await_read_time, await_write_time, 250 where category=disk
```

Then choose All Combined (1) in the Facets list. The scalar threshold 250 will also be rendered on the chart:

I get the warning "The query hit the maximum results limit". How do I work around the limit? There is a limit on the number of results that can be returned by a query. When a query returns more time-series streams than the limit a warning for "partial results" is issued. To circumvent this problem, reduce the number of metrics you are trying to retrieve or see the topic Configuring Queries.

Series Query Results.

You can use the `rlike` operator to limit the query to a subset of entities. For example, inst

```
select await_time, await_read_time, await_write_time, 250 where  
category=DISK
```

you can use

```
select await_time, await_read_time, await_write_time, 250 where  
category=DISK and hostname rlike "host1[0-9]?.cloudera.com"
```

The latter query retrieves the disk metrics for ten hosts.

How do I discover which metrics are available for which entities?

- Type Select in the text box and then press Space or continue typing. Metrics matching the letters you type display in a drop-down list.
- Select Charts Chart Builder , click the question mark icon to the right of the Build Chart button and click the List of Metrics link
- Retrieve all metrics for the type of entity:

```
select * where roleType=DATANODE
```

## Metric Aggregation

In addition to collecting and storing raw metric values, the Cloudera Manager Service Monitor produce a number of aggregate metrics from the raw metric data.

Where a raw data point is a timestamp value pair, an aggregate metric point is a timestamp p statistics including the minimum, maximum, average, and standard deviation of the data point aggregate.

Individual metric streams are aggregated across time to produce statistical summaries at dif For example, an individual metric stream of the number of open file descriptors on a host w time to the ten-minute, hourly, six-hourly, daily and weekly data granularities. A point in will include the maximum number of open file descriptors seen during that hour, the minimum, on. When servicing a time-series request, either for the Cloudera Manager UI or API, the Ser Monitor automatically choose the appropriate data granularity based on the time-range reques Cross-Time Aggregate Example

Consider the following `fd_open` raw metric values for a host:

```
9:00, 100 fds  
9:01, 101 fds  
9:02, 102 fds  
... 9:09, 109 fds
```

The ten minutely cross-time aggregate point covering the ten-minute window from 9:00 - 9:10 following statistics and metadata:

```
min: 100 fds  
min timestamp: 9:00  
max 109 fds
```

```
max timestamp 9:09
mean 104.5 fds
standard deviation: 3.02765 fds
count: 10 points
sample: 109 fds
sample timestamp: 9:09
```

The Service Monitor and Host Monitor also produce cross-entity aggregates for a number of entities. Cross-entity aggregates are produced by considering the metric value of a particular metric across all entities of the same type at a particular time. For each stream considered, two metrics are produced: one for statistics such as the minimum, maximum, average and standard deviation across all considered entities, and another for the identities of the entities that had the minimum and maximum values. The second tracks the sum of the metric values across all considered entities.

An example of the first type of cross-entity aggregate is the `fd_open_across_datanodes` metric. This metric contains aggregate statistics on the `fd_open` metric value for all the DataNodes in the service. This metric contains statistics for all the DataNodes within that rack, and so on. An example of the second type of cross-entity aggregate is the `total_fd_open_across_datanodes` metric. For an HDFS service this metric tracks the total number of file descriptors open by all the DataNodes in the service. For a rack this metric tracks the number of file descriptors open by all the DataNodes within the rack, and so on. Note that unlike the first type of cross-entity aggregate, this second type of cross-entity aggregate is a simple timestamp, value and standard deviation statistics.

#### Cross-Entity Aggregate Example

Consider the following `fd_open` raw metric values for a set of ten DataNodes in an HDFS service at timestamp 9:09:

```
datanode-0, 200 fds
datanode-1, 201 fds
datanode-2, 202 fds
...
datanode-9, 209 fds
```

The cross-entity aggregate `fd_open_across_datanodes` point for that HDFS service at that time contains the following statistics and metadata:

```
min: 200 fds
```

#### Cloudera Manager Charting Time-Series Data

```
min entity: datanode-0
max: 209 fds
max entity: datanode-9
mean: 204.5 fds
standard deviation: 3.02765 fds
count: 10 points
sample: 209 fds
sample entity: datanode-9
```

Just like every other metric, cross-entity aggregates are aggregated across time. For example, the cross-entity aggregate of `fd_open_across_datanodes` for an HDFS service will include the maximum `fd_open` value across all DataNodes in the service at a particular time.

DataNode in that service over that hour, the average value over the hour, and so on. A point total\_fd\_open\_across\_datanodes for an HDFS service will contain statistics on the value of total\_fd\_open\_across\_datanodes for that service over the hour.

### Presentation of Aggregate Data

Aggregate data points returned from the Cloudera Manager API appear as shown in this section. A cross-time aggregate:

```
{
  "timestamp" : "2014-02-24T00:00:00.000Z",
  "value" : 0.014541698027508003,
  "type" : "SAMPLE",
  "aggregateStatistics" : {
    "sampleTime" : "2014-02-23T23:59:35.000Z",
    "sampleValue" : 0.0,
    "count" : 360,
    "min" : 0.0,
    "minTime" : "2014-02-23T18:00:35.000Z",
    "max" : 2.9516129032258065,
    "maxTime" : "2014-02-23T19:37:36.000Z",
    "mean" : 0.014541698027508003,
    "stdDev" : 0.17041289765265377
  }
}
```

A raw cross-entity aggregate:

```
{
  "timestamp" : "2014-03-26T00:50:15.725Z",
  "value" : 3288.0,
  "type" : "SAMPLE",
  "aggregateStatistics" : {
    "sampleTime" : "2014-03-26T00:49:19.000Z",
    "sampleValue" : 7232.0,
    "count" : 4,
    "min" : 1600.0,
    "minTime" : "2014-03-26T00:49:42.000Z",
    "max" : 7232.0,
    "maxTime" : "2014-03-26T00:49:19.000Z",
    "mean" : 3288.0,
    "stdDev" : 2656.7549127961856,
    "crossEntityMetadata" : {
      "maxEntityDisplayName" : "cleroy-9-1.ent.cloudera.com",
      "minEntityDisplayName" : "cleroy-9-4.ent.cloudera.com",
      "numEntities" : 4.0
    }
  }
}
```

```
}
}
```

## Cloudera Manager Charting Time-Series Data

A cross-time, cross-entity aggregate:

```
{
  "timestamp" : "2014-03-11T00:00:00.000Z",
  "value" : 3220.818863879957,
  "type" : "SAMPLE",
  "aggregateStatistics" : {
    "sampleTime" : "2014-03-10T22:28:48.000Z",
    "sampleValue" : 7200.0,
    "count" : 933,
    "min" : 1536.0,
    "minTime" : "2014-03-10T21:02:17.000Z",
    "max" : 7200.0,
    "maxTime" : "2014-03-10T22:28:48.000Z",
    "mean" : 3220.818863879957,
    "stdDev" : 2188.6143063503378,
    "crossEntityMetadata" : {
      "maxEntityDisplayName" : "cleroy-9-1.ent.cloudera.com",
      "minEntityDisplayName" : "cleroy-9-4.ent.cloudera.com",
      "numEntities" : 3.9787037037037036
    }
  }
}
```

These differ from non-aggregate data points by having the `aggregateStatistics` structure. Note that the point structure will always be the same as the `aggregateStatistics` mean field. The Cloudera Manager charting interface displays aggregate statistics in a number of ways. First, aggregate statistics are made available in the chart's popover when dealing with aggregate data. Second, it is possible to turn on and turn off the display of maximum time-series streams in line charts of aggregate data. These streams are displayed as horizontal lines as a visual indication of the underlying metric values data range over the time considered, and the range of the data. These lines are displayed by default for single stream line charts of aggregate data. For all other charts, they can be turned on and turned off using the chart popover.

## Accessing Aggregate Statistics Through tsquery

You can use the `stats` function to access aggregate statistics directly in `tsquery`. For example, `select stats(fd_open_across_datanodes, max) where category = service and service = "hdfs-service"` will return a single time-series stream containing the just the maximum `stats(fd_open_across_datanodes)` stream. The following statistics are available through the `stats` function: `mean`, `stdDev`, and `sample`.

Related Information

[tsquery Language](#)

## Filtering Metrics

Metric Filters limit the amount of metric data reported to the Cloudera Manager server to avoid overloading the server with too much data.

### About this task

Metrics Filters allow you to limit the amount of metric data sent to the Cloudera Manager Server. In some clusters, some services, such as Kudu, send a high volume of non-essential metrics data to the Cloudera Manager Server, which can overload it, causing gaps in the data reported from these metrics in charts/dashboards and potentially limiting the ability for Cloudera Manager to effectively monitor cluster health. You can configure Metric Filters that limit the amount of data sent to the Service Monitor and the Cloudera Manager Server. You can configure Metric Filters for any service deployed in a cluster.

You can configure the filter in several ways:

- Limit the collected metrics to only include those required for Health Tests.
- Limit the collected metrics to only include metrics used in the charts on the main page ('Dashboard') of the service.
- Include specific metrics - only the specified metrics will be collected for this service
- Exclude specific metrics - the specified metrics will not be collected for this service

### Procedure

To configure a Metric Filter:

1. Log in to the Cloudera Manager Admin Console.
2. Navigate to the cluster where you want to add the filter.
3. Click ConfigurationMetric Filters. The Configuration page displays the Metric Filter parameter for all roles in the cluster. You can now choose whether to edit all the Metric Filters using the same values, or edit the filter for a specific role.
4. Do one of the following:
  - To configure the filter for a specific role, click the Edit Individual Values link and locate the parameter for the service. Follow the steps below.
  - To configure the same filter for all roles, edit the Default Group.
5. Select one of the following:
  - Include Only Health Test Metric Set
  - Include Only Default Dashboard Metric Set
6. To filter specific metrics:
  - a) Select either Include or Exclude from the 'Include/Exclude Custom Metrics' drop-down list. If you selected either of the Metric sets, and select Include, the custom metrics will be added to the selected Metrics sets. If you select Exclude, they will be excluded from the selected sets.
  - b) Click the "Plus" icon to add a metric.
  - c) Enter the metric name.
  - d) Click the "Plus" icon to add additional metrics. Note: You can find



the names of specific metrics you would like to include or exclude either from the Cloudera Manager Metrics Reference (see link below) or from a chart that is displaying the required metrics in the Cloudera Manager Admin Console, as follows:

- a. Go to the Status page for a cluster or service.
- b. In a chart, click the gear icon and select Open in Chart Builder. The query displays at the top of the page. Metric names are part of the query. For example, the following query:

```
select dfs_capacity, dfs_capacity_used, dfs_capacity_used_non_hdfs
where entityName=$SERVICENAME
```

contains the following metric names:

- dfs\_capacity
- Dfs\_capacity\_used
- dfs\_capacity\_used\_non\_hdfs

7. Click Save Changes.
8. Refresh the cluster:
  - a) Go the cluster's Status page.
  - b) Click ActionsRefresh Cluster. This action does not restart the cluster. The cluster configuration is refreshed. Note: You can configure Cloudera Manager to automatically refresh the cluster after you update Metric configurations by enabling the Enable auto refresh for metric configurations parameter. This parameter is set at the role level, and, when set to true, Cloudera Manager refreshes only the role affected by a change to the metric filter or collection enable settings, not the entire cluster.

#### Results

Note: If you do not select any options, all metrics for the service will be sent to the Service Monitor.