Metric Quantization

Learn about how Sumo aggregates metric data points over time buckets.

Sumo ingests individual metric data points from your metric sources. In metric visualizations, rather than charting individual data points, Sumo presents the aggregated value of the data points received during an interval.

Quantization is the process of aggregating metric data points for time series over an interval, for example an hour or a minute, using a particular aggregation function: \texttt{avg}, \texttt{min}, \texttt{max}, \texttt{sum}, or \texttt{count}.

Quantization terminology

This section defines the quantization-related terms we use in Sumo.

**Buckets**

We use the term bucket to refer to the intervals across which Sumo quantizes your metrics.

When you run a metric query, Sumo divides up your metric query time range into contiguous buckets based the selected quantization level on the metric query page, or upon the interval you specify in the \texttt{quantize} operator. For example, given this query:

\texttt{cpu | quantize to 15m}

Sumo divides your time range into 15 minute buckets.

For each bucket, Sumo uses a rollup type, described below, to aggregate the values of all the data points in the bucket. The aggregated values are displayed in your metric visualization.

By default, Sumo uses the \texttt{avg} rollup type. You can specify another rollup type by using the \texttt{quantize} operator, as described in Quantize with rollup type specified below.

**Rollup types**

We use the term rollup to refer to the aggregation function Sumo uses when quantizing metrics. This table describes the different rollup types you can select when running a query.

<table>
<thead>
<tr>
<th>Rollup type</th>
<th>Description</th>
<th>When to use</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
avg  Calculates the average value of the data points for a time series in each bucket.  Useful for gauges (metrics that may go up or down over time), for example CPU usage.

min  Calculates the minimum value among the data points for a time series in each bucket.  Useful for metrics whose values decrease over time. For example, Minimum Response Time.

max  Calculates the maximum value among the data points for a time series in each bucket.  Useful for metrics whose values increase over time. For example, Bytes Served.

sum  Calculates the sum of the values of the data points for a time series in each bucket.  Useful for metrics that measure the difference between consecutive data points of a time series.

count  Calculates the count of data points for a time series in each bucket.  Useful for calculating the number data points generated by a set of time-series.

---

### Quantization at ingestion

Upon ingestion, Sumo quantizes raw metric data points to one minute and one hour resolutions for all rollup types: `avg`, `min`, `max`, `sum`, and `count`. This data is stored in 1 minute and 1 hour rollup tables in Sumo. The raw data is stored in a table referred to as the baseline table. For information about retention times, see [Metric Ingestion and Storage](#).

---

### Automatic quantization at query time

This section describes how Sumo quantizes metrics when you run a metric query without the `quantize` operator.

If you do not use the `quantize` operator in your metric query, Sumo automatically determines an optimal quantization interval, based on the age of the data you are querying and the number of data points. The quantization interval is shown at the top of the metric query tab. Sumo will quantize the metric data points using the `avg` rollup type, for each bucket in your time range.

![Metric Query Tab](image)

The age of the metrics in time range governs the minimum quantization interval (based on what rollups are available for the query time range). Sumo retains only the last 7 days of raw metric data, and only the last 30 days of 1 minute rollups. So, when you query metrics that are more than 7 days old, Sumo must quantize the data to at least 1 minute, because that's the minimum resolution rollup available given the age of the data. Similarly, when you query metrics older
than 30 days, Sumo will must quantize to 1 hour, because that’s the minimum resolution rollup available for metrics over 30 days old.

In addition to the age of data you query, the volume of data points per series is a factor in automatic quantization and may dictate a longer quantization interval. Sumo selects a quantization interval that does not result in too many data points on the visualization. UI-based queries restrict results to contain about 800 points per series, so Sumo chooses a quantization level that keep results within that limit.

If you want, you can override the automatic quantization interval. Click the displayed quantization interval to display a pulldown list of intervals, from one second to one day (1s, 10s, 1m, 15 min, 1h, and 1d).

Sumo sets the actual quantization interval to be as close to your selection as possible. If it is not possible to set the actual interval to the targeted interval—typically because too many data points would be produced to reasonably show on the chart—Sumo displays a message like the following:

**Examples of how Sumo chooses quantization interval and rollup table to use**

This section shows examples of how Sumo chooses the quantization interval and rollup table to use for a query if you do not specify those options using the `quantize` operator.

**Example 1. Time range is -15m**

We calculate a target quantization interval by dividing the time range (in seconds) by the maximum number of data points we want to present per time series, 800.

\[
\frac{(15 \text{ minutes} \times 60 \text{ seconds per minute})}{800 \text{ buckets}} = 1.125 \text{ seconds per bucket}
\]

So, we need to quantize to at least 1.25s for the visualization to be usable.

Sumo effectively rewrites the query to this:

`cpu-idle | quantize to 5s`

Now, we look at what rollups are available for the last 15 minutes: raw and 1m. Sumo chooses between the two based on rollup granularity. The 1m rollup is less granular that our desired quantization interval, so Sumo will quantize to 5 seconds using raw.

**Example 2. Time range is -1d**

We calculate a target quantization interval by dividing the time range (in seconds) by the maximum number of data points we want to present per time series, 800.

\[
\frac{(24 \text{ hours} \times 60 \text{ minutes per hour} \times 60 \text{ seconds per minute})}{800 \text{ buckets}} = 108 \text{ seconds per bucket.}
\]

So, we need to quantize to at least 108s for the visualization to be usable.
Sumo effectively rewrites the query to this:

```
  cpu-idle | quantize to 15m
```

### Explicit quantization at query time

When you run a metric query, you can optionally use the `quantize` operator to specify a quantization interval and rollup type, or both.

When you run a query with the `quantize` operator, the way that Sumo quantizes your metric data points depends on:

- The rollup type you specify, if any, in the `quantize` clause of your query. Rollup types include `avg`, `min`, `max`, `sum`, and `count`. (Specifying rollup type is optional for the `quantize` operator.)
- The operator, if any, that follows the `quantize` clause of your query.

#### Quantize with rollup type specified

If your metric query uses the `quantize` operator and specifies a rollup type, Sumo will only quantize metric data points accordingly. For example, given this query,

```
  cpu | quantize to 15m using sum
```

Sumo will quantize only to the `sum` rollup type.

#### Quantize with no rollup type specified

If your metric query uses the `quantize` operator without specifying a rollup type, internally, Sumo produces all rollups, `avg`, `min`, `max`, `sum`, and `count`. We refer to these internal results as "descriptive" points. Here is an example of a descriptive point.

```
{min=2, max=4, sum=6, count=2, avg=3}
```

How Sumo processes the descriptive points produced depends on the operator immediately following the `quantize` clause.

**quantize operator is followed by an aggregation operator**

An aggregation operator following a `quantize` operator converts descriptive points to simple points by discarding all but one rollup before performing aggregations.

- The `min` operator discards all rollups except `min` before performing the aggregation.
- The `max` aggregate operator discards all rollups except `max` before performing the aggregations.
- Other aggregation operators discard all rollups except `avg` before performing the aggregation.

Here are some examples of queries and the rollups that are selected in each case:
<table>
<thead>
<tr>
<th>Query</th>
<th>What Happens</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu</td>
<td>quantize to 1m</td>
</tr>
<tr>
<td>cpu</td>
<td>quantize to 1m</td>
</tr>
<tr>
<td>cpu</td>
<td>quantize to 1m</td>
</tr>
<tr>
<td>cpu</td>
<td>quantize to 1m</td>
</tr>
<tr>
<td>cpu</td>
<td>quantize to 1m</td>
</tr>
<tr>
<td>cpu</td>
<td>quantize to 1m</td>
</tr>
</tbody>
</table>

**quantize operator is followed by a parse operator**

The descriptive points might be passed through without change. For example, the `parse` operator changes time series metadata but lets data points through unchanged. For example,

```plaintext
... | quantize to 5s | parse field=_sourceHost - as cluster,instance | ..
```

**quantize operator is followed by another quantize operator**

A `quantize` operator following a `quantize` operator uses all the rollup information present in a descriptive point meaningfully. For example,

```plaintext
... | quantize to 15s | quantize to 1m | ...  
```

The first `quantize` operator receives a stream of simple points, representing baseline data, and produces a stream of descriptive points.

To illustrate this transformation, suppose a given time series contains the following three simple points in a given 15s bucket from 12:00:30 to 12:00:45:

<table>
<thead>
<tr>
<th>timestamp</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00:33</td>
<td>4.0</td>
</tr>
<tr>
<td>12:00:39</td>
<td>6.0</td>
</tr>
<tr>
<td>12:00:40</td>
<td>5.0</td>
</tr>
</tbody>
</table>

It will produce the following descriptive point for the above bucket:
This stream of descriptive points is then fed to the second quantize operator, which, again, produces descriptive points. Let’s assume it gets the following 4 descriptive points within a given one minute bucket from 12:00:00 to 12:01:00:

```
timestamp | min | max | sum | count | avg
---------------------------------------------------
12:00:00  | 1.0 | 7.0 | 8.0 | 2     | 4.0
12:00:15  | 1.0 | 1.0 | 1.0 | 1     | 1.0
12:00:30  | 4.0 | 6.0 | 15.0| 3     | 6.0
12:00:45  | 5.0 | 5.0 | 15.0| 3     | 5.0
```

It will produce the following descriptive point for the above one minute bucket:

```
timestamp | min | max | sum | count | avg
---------------------------------------------------
12:00:30  | 1.0 | 7.0 | 39.0| 9     | 4.3
```