

KIP-488: Clean up Sum,Count,Total Metrics

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: *Accepted*

Discussion thread: http://mail-archives.apache.org/mod_mbox/kafka-dev/201907.mbox/%3CCAyirGtNhLb8YnmmAUgiwQcjSP-mYCMESB7dAVK6mqQDVdnwLg%40mail.gmail.com%3E

JIRA: [KAFKA-8696](#) - Getting issue details... STATUS

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Kafka has a family of metrics consisting of:

- `org.apache.kafka.common.metrics.stats.Count`
- `org.apache.kafka.common.metrics.stats.Sum`
- `org.apache.kafka.common.metrics.stats.Total`
- `org.apache.kafka.common.metrics.stats.Rate.SampledTotal`
- `org.apache.kafka.streams.processor.internals.metrics.CumulativeCount`

These metrics are all related to each other, but their relationship is obscure (and one is redundant) (and another is internal).

I've recently been involved in a *third* recapitulation of trying to work out which metric does what. It seems like it's time to clean up the mess and save everyone from having to work out the mystery for themselves.

Public Interfaces

The affected public interfaces are:

sampled count metric:

- (deprecated) `org.apache.kafka.common.metrics.stats.Count`
- (new) `org.apache.kafka.common.metrics.stats.WindowedCount`

sampled sum metric:

- (deprecated) `org.apache.kafka.common.metrics.stats.Rate.SampledTotal`
- (deprecated) `org.apache.kafka.common.metrics.stats.Sum`
- (new) `org.apache.kafka.common.metrics.stats.WindowedSum`

non-sampled count metric:

- (internal: removed) `org.apache.kafka.streams.processor.internals.metrics.CumulativeCount`
- (new) `org.apache.kafka.common.metrics.stats.CumulativeCount`

non-sampled sum metric:

- (deprecated) `org.apache.kafka.common.metrics.stats.Total`
- (new) `org.apache.kafka.common.metrics.stats.CumulativeSum`

Proposed Changes

The existing metrics cover four quadrants of a matrix:

	count	sum
--	-------	-----

sampled	Count	SampledTotal
		Sum
non-sampled	(internal) CumulativeCount	Total

It's immediately apparent that there's no consistency in naming, that there's a missing quadrant, and that one quadrant is redundantly covered.

The proposal is simple:

	count	sum
sampled	WindowedCount	WindowedSum
non-sampled	CumulativeCount	CumulativeSum

Under this proposal, the metrics are clearly and regularly named and all quadrants are covered uniquely. There is no ambiguity in the names, and the structure of the names also indicate a pattern that would guide users to select the correct metric for their needs.

Compatibility, Deprecation, and Migration Plan

Existing metrics are deprecated in favor of unambiguously named ones. They will be made to subclass the new metrics to avoid code duplication, but this won't cause any code compatibility issues, since they'll still inherit the same interfaces.

Rejected Alternatives

- "Running" or "Total" instead of "Cumulative": After some discussion and some research, "Cumulative" appears to be the technically correct term: "In a cumulative moving average, the data arrive in an ordered datum stream, and the user would like to get the average of all of the data up until the current datum point." (https://en.wikipedia.org/wiki/Moving_average)
- "Sampled" instead of "Windowed": Sampling is the implementation, and in the current Metrics framework, it implies that the metric is windowed, but the name bears no such connotation. Since the distinction we wish to draw is the these metrics drop old data, as opposed to the cumulative ones, we choose a name that actually means it will drop old data.
- "Moving" instead of "Windowed": Any stat that is continuously updated is moving, whether it is windowed or not.
- "Simple" or "SimpleWindowed" instead of "Windowed": These options have the benefit that they specify the weighting function (uniform) in addition to implying windowing, but the term "simple" is itself jargon. It's also not necessary, as the absence of a weighting function in the name can also imply that the weighting is uniform. If we want to add a metric with some other function in the future, we can always name it like ExponentiallyWeightedWindowedBlahBlahBlah to differentiate it from WindowedBlahBlahBlah.