KIP-938: Add more metrics for measuring KRaft performance

Status

Current state: *Accepted*

Discussion thread: https://lists.apache.org/thread/wtx9qcr613gyqtm0bx8rlsckrg5pl276

JIRA: KAFKA-15183

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

Motivation

The motivation of this KIP is to add some more metrics for the purpose of measuring performance. Most of these metrics are focused on KRaft mode.

Public Interfaces

We propose adding the following new metrics:

Name	Context	Type	Mode	Description
`kafka.controller: type=KafkaController, name=TimedOutBrokerHeartbeatCount`	Controller	Long	KRaft only	The number of broker heartbeats that timed out on this controller since the process was started. Note that only active controllers handle heartbeats, so only they will see increases in this metric.
`kafka.controller: type=KafkaController, name=EventQueueOperationsStartedCount`	Controllers	Long	KRaft only	The total number of controller event queue operations that were started. This includes deferred operations.
`kafka.controller: type=KafkaController, name=EventQueueOperationsTimedOutCount`	Controllers	Long	KRaft only	The total number of controller event queue operations that timed out before they could be performed.
`kafka.controller: type=KafkaController, name=NewActiveControllersCount`	Controller	Long	KRaft only	Counts the number of times this node has seen a new controller elected. A transition to the "no leader" state is not counted here. If the same controller as before becomes active, that still counts.
`kafka.server: type=MetadataLoader, name=CurrentMetadataVersion`	Broker and Controller	Integer	KRaft only	Outputs the feature level of the current effective metadata version.
`kafka.server: type=MetadataLoader, name=HandleLoadSnapshotCount`	Broker and Controller	Long	KRaft only	The total number of times we have loaded a KRaft snapshot since the process was started.

`kafka.server: type=SnapshotEmitter, name=LatestSnapshotGeneratedBytes`	Broker and Controller	Long	KRaft only	The total size in bytes of the latest snapshot that the node has generated. If none have been generated yet, this is the size of the latest snapshot that was loaded. If no snapshots have been generated or loaded, this is 0.
`kafka.server: type=SnapshotEmitter, name=LatestSnapshotGeneratedAgeMs`	Broker and Controller	Long	KRaft only	The interval in miliseconds since the latest snapshot that the node has generated. If none have been generated yet, this is approximately the time delta since the process was started.
`kafka.server: type=ForwardingManager, name=QueueTimeMs`	Broker	Histogram	KRaft and ZK	A histogram describing the amount of time in milliseconds each admin request spends in the broker's forwarding manager queue, waiting to be sent to the controller. This does not include the time that the request spends waiting for a response from the controller.
`kafka.server: type=ForwardingManager, name=QueueLength`	Broker	Integer	KRaft and ZK	The current number of RPCs that are waiting in the broker's forwarding manager queue, waiting to be sent to the controller.
`kafka.server: type=ForwardingManager, name=RemoteTimeMs`	Broker	Histogram	KRaft and ZK	A histogram describing the amount of time in milliseconds each request sent by the ForwardingManager spends waiting for a response. This does not include the time spent in the queue.

Implementation Notes

Lockless

In order to avoid excessive performance impacts from these new metrics, none of them will require locks to read. (Except for any locks inside the metrics library, JMX implementation, and so on.)

Histogram details

Histograms have the standard Yammer MBean fields of 50thPercentile, 75thPercentile, 95thPercentile, 98thPercentile, 999thPercentile, 99thPercentile, Count, Max, Mean, Min, StdDev.

Rationale

TimedOutBrokerHeartbeats

This metric is useful to monitor because when broker heartbeats are timing out, that indicates a performance problem on the active controller.

EventQueueOperationsStarted

This is a rough measure of how busy the controller is. This lets us know how many operations per second different quorum controller clusters can perform.

EventQueueOperationsTimedOut

This is a rough measure of how much load we are shedding by means of timeouts. If we see this increase faster than TimedOutBrokerHeartbeats, we know that operations other than heartbeats are being impacted by timeouts.

NewActiveControllersCount

The main reason to monitor this metric is to make sure we are not electing too many new controllers per minute.

CurrentMetadataVersion

This metric simply reflects the current metadata version. It comes from the numeric "feature level" of the metadata version. It is useful for administrators with multiple clusters, who want to ensure that they're all up-to-date. It also is helpful to know at a glance when a metadata version transition occurred.

HandleLoadSnapshotCount

This metric counts the number of times we have loaded a metadata snapshot. This is an O(N) operation since it involves reloading the full metadata state. So it's helpful to know when this has occurred.

LatestSnapshotGeneratedBytes

This metric is useful to monitor the size of the snapshot generated by the cluster. In general, the larger the snapshot gets, the more resources the cluster will need.

LatestSnapshotGeneratedAgeMs

This metric is useful to monitor how long it has been since the node last generated a snapshot. If this time grows too large, it may indicate a potential problem, since loading times might also become very large.

ForwardingManager QueueTimeMs, QueueLength, RemoteTimeMs

These metrics are useful to understand how much of the latency of admin commands is due to waiting for the forwarding manager, versus other causes.

Compatibility, Deprecation, and Migration Plan

These will be newly exposed metrics and there will be no impact on existing kafka versions.

Test Plan

We will add junit tests to verify the new metrics.

Rejected Alternatives

Rather than adding NewActiveControllersCount, we could monitor existing metrics such as the metadata log epoch, or the ActiveCountroller metric that is either 0 or 1. However, these alternatives are not as good.

- The metadata log epoch may increase multiple times during a Raft election even when only one new leader results.
- Measuring transitions in the ActiveController metric is difficult in many metrics collection systems. It's also easy to lose track of a transition if the sampling period is too long.