

KIP-847: Add ProducerIdCount metrics

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

Status

Current state: *Accepted* ([voting thread](#))

Discussion thread: [here](#)

JIRA: [here](#)

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

Motivation

[KIP-664: Provide tooling to detect and abort hanging transactions](#) provided tooling to get visibility into transactional and idempotent producers that the broker keeps track of. This KIP proposes to add `ProducerIdCount` metrics that enable easy monitoring of transactional and idempotent producer counts on the broker.

Producer ids are used by idempotent and transaction producers. The brokers keep a small amount of metadata (e.g. producer id, epoch, sequence number, etc.) in memory for every partition that the idempotent producer produced to. This metadata is maintained on every replica and it's recovered from logs and snapshots even if brokers restart. The producer id and its metadata is removed after it's been inactive for a certain time controlled by the `transactional.id.timeout.ms` configuration setting, the default is 7 days. The [KIP-98 - Exactly Once Delivery and Transactional Messaging](#) has details on producer ids and related protocols and data structures.

In idempotent producers, a new producer id is created when `KafkaProducer` is created. A badly written application may frequently create new `KafkaProducer` objects. This is not optimal in general, but specifically for idempotent producers, doing so would pollute broker memory with producer ids and related metadata. Even though the metadata for each producer id is small, creating too many producer ids could run brokers out of memory.

The `ProducerIdCount` metric reflects the total count of producer ids in all partitions maintained at each broker. The metric can be used to set up alerts so that the abovementioned pattern can proactively be detected and action could be taken before too many producer ids run the broker out of memory.

Public Interfaces

We propose adding a new broker metric

Name	Description
<code>kafka.server:type=ReplicaManager, name=ProducerIdCount</code>	The total number of active transactional / idempotent producer ids in all partitions maintained in the broker.

Proposed Changes

Add the new metric to the `ReplicaManager` class.

Compatibility, Deprecation, and Migration Plan

- No migration plan is needed because the metric is new

Rejected Alternatives

Have a partition level metric as well - this doesn't seem to be needed as we can use [KIP-664: Provide tooling to detect and abort hanging transactions](#) for detailed debugging, once alerted on total producer id count on the broker.

Name the metric `ProducerCount` - may be misleading as the producers without producer ids are not counted.

Have 2 metrics `IdempotentProducerCount` and `TransactionalProducerCount` - currently we don't keep track which producer id is idempotent and which is transactional, adding that would add some complexity and potential runtime overhead, currently there doesn't seem to be a monitoring scenario that requires distinguishing between the two.