

# KIP-92 - Add per partition lag metrics to KafkaConsumer

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

## Status

**Current state:** *Accepted*

**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

Currently KafkaConsumer only has a metric of max lag across all the partitions. It would be useful to know per partition lag as well.

## Public Interfaces

There is no programmatic public interface change. We are only adding new metrics.

## Proposed Changes

Add per partition lag metrics to KafkaConsumer. The metric names would be:

```
"TOPIC-PARTITION_ID.record-lag"  
"TOPIC-PARTITION_ID.record-lag-avg"  
"TOPIC-PARTITION_ID.record-lag-max"
```

The way to get the metrics is the same as getting other metrics. For example:

```
TopicPartition tp = new TopicPartition("topic", 0);  
HashMap<String, String> tags = new HashMap<>();  
tags.put("client-id", "metricTestConsumer0");  
  
double currentLag = kafkaConsumer.metrics().get(new MetricName(tp + ".records-lag", "consumer-fetch-manager-metrics", "", tags)).value()  
double averageLag = kafkaConsumer.metrics().get(new MetricName(tp + ".records-lag-avg", "consumer-fetch-manager-metrics", "", tags)).value()  
double maxLag = kafkaConsumer.metrics().get(new MetricName(tp + ".records-lag-max", "consumer-fetch-manager-metrics", "", tags)).value()
```

When the consumer does not own a partition anymore, the lag metric will be removed.

Notice that the per partition lag metrics have an average value reported while the max lag across all the partitions only has a max value reported. This is because the average value of max lag from all the partitions may be an average from different partitions at different times, so it is not very useful. But the average value of per partition lag at different times is more meaningful.

## Compatibility, Deprecation, and Migration Plan

The change is fully backwards compatible.

## Rejected Alternatives

None