

KIP-551: Expose disk read and write metrics

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Implementation](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)
 - [External Monitoring](#)

Status

Current state: Accepted

Discussion thread: [here](#)

JIRA: [KAFKA-9292](#)

Motivation

It's often helpful to know how many bytes Kafka is reading and writing from the disk. The reason is because when disk access is required, there may be some impact on latency and bandwidth.

We currently don't have a metric that measures this directly. It would be useful to add one.

Public Interfaces

Two new metrics will be added.

Full Name	Type	Description
kafka.server:type=KafkaServer, name=TotalDiskReadBytes	64-bit gauge	The total number of bytes read by the broker process. This includes reads from all disks. It does not include reads that came out of page cache.
kafka.server:type=KafkaServer, name=TotalDiskWriteBytes	64-bit gauge	The total number of bytes written by the broker process. This includes writes from all disks.

For now, these metrics will be present only on Linux. They will not be created when running the broker on other operating systems, such as MacOS and Windows. In the future, we may add an implementation for these metrics for MacOS and Windows.

Implementation

These metrics will be collected from the `/proc/self/io` file on Linux. `TotalDiskReadBytes` comes out of the line marked `"read_bytes:"`. `TotalDiskWriteBytes` comes out of taking the line marked `"write_bytes:"` and subtracting it from the value in the line marked `"cancelled_write_bytes:"`. This reflects the fact that write bytes which were cancelled and not performed should not be counted.

The broker will read this file from the `/proc` filesystem whenever the metric needs to be accessed. We will not re-read the file more than once a millisecond even if the gauge is accessed multiple times during that period.

Compatibility, Deprecation, and Migration Plan

This change adds a new metric, so there is no need to deprecate or migrate any interfaces.

Rejected Alternatives

External Monitoring

We could use an external program to monitor these metrics. This makes a lot of sense for metrics like overall CPU use. However, the number of bytes the broker process read and write is a metric closely tied to the broker process itself, so it makes more sense for that process itself to monitor it.