KIP-864: Add End-To-End Latency Metrics to Connectors

Status

Current state: *"Under Discussion"*

Discussion thread: *here*

JIRA: *here*

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

Draft implementation: https://github.com/apache/kafka/pull/12647

Motivation

Currently, Sink and Source Connectors include latency metrics covering only the time expend interacting with the external systems —

- `put-batch-latency` metric measures the time to sink a batch of records, and
- `poll-batch-time` metric measures the time to poll a batch of records from the external system.

At the moment it's difficult to understand another latency aspects, e.g.:

- how much latency has the connector introduced in the process, or
- how long after being written a record is processed.

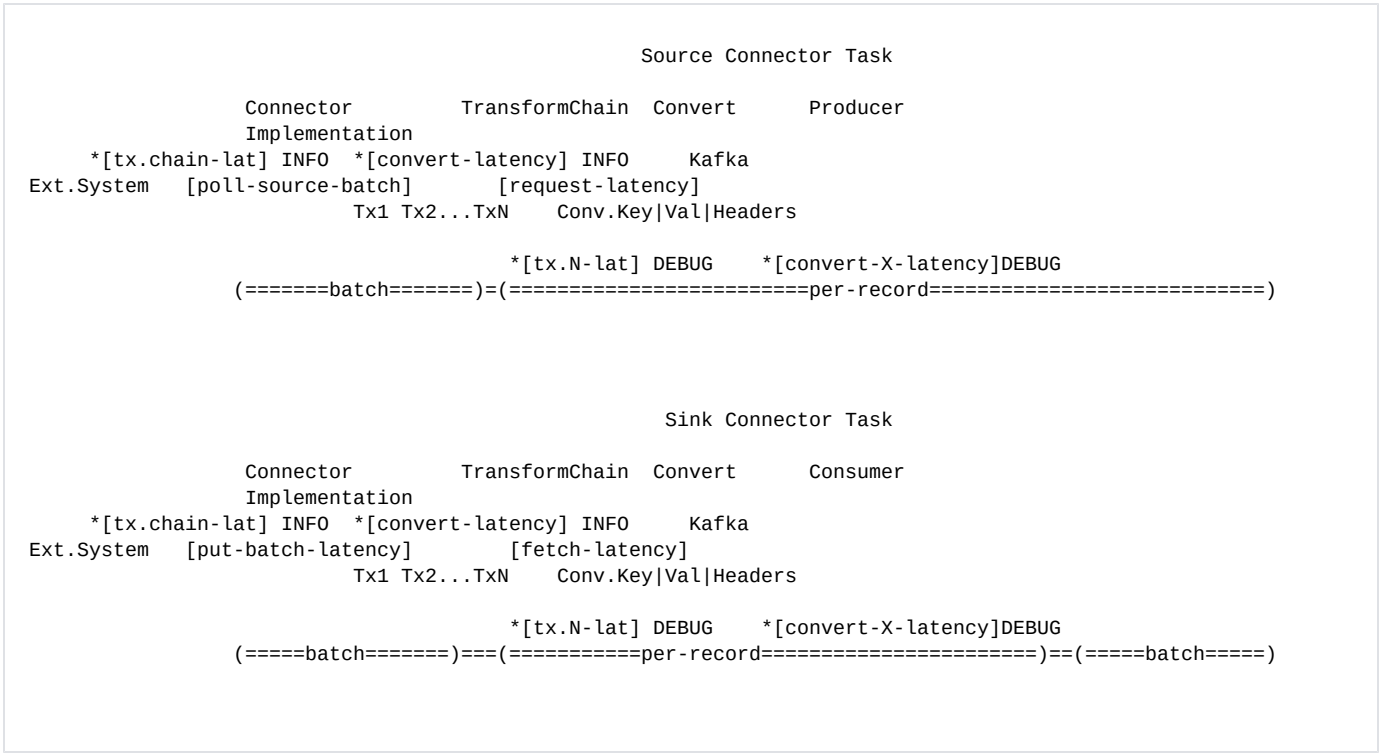
In order to observe the connector's performance and measure its complete end-to-end latency from sources to sinks, this KIP is proposing the following additional measurements:

- In the source connector:
  - After polling, there are transformations and conversions that happen before the records are sent to Kafka.
- In the sink connector:
  - Record latency: `wall-clock time - record timestamp` to evaluate how *late* records are processed.
  - Convert and transform time before sending records to a external system

With these enhanced metrics available, operators/developers could:

- Monitor and alert when sink processing is happening after accepted latency (e.g. > 5secs)
  - Current workaround:
    - Inserting records timestamp into the target system to handle the calculations there.
- Observe processing lifecycle and monitor on spikes caused by convert and transforms and reduce the time to remediation
  - Current workaround:
    - Infer connector's latency by capturing the timestamp of the source system and comparing it with the Kafka record timestamp; which is not trivial.
    - With the current poll/put and the produce/fetch latencies are not enough to infer the time taken in between.
- Performance testing transforms in non-production environments to measure how latency increases related to throughput.

Proposed Changes



(*) New metrics

Source Connectors have the following stages:

- task implementation polling records: gather *batch of records* from external system
- transform: apply transformation chain *individually*
- convert: convert records to `ProducerRecord` *individually*
- send records: send records to Kafka topics *individually*

The stage for polling records already has a latency metric: `poll-batch-time` .

`transform-chain-source-record-time` and `convert-source-record-time` metric will measure the transformations applied and conversion from `SourceRecord` into `ProducerRecord` .

The stage for sending records can be monitored with Producer sender metrics, e.g. `request-latency-avg/max`

Sink Connectors have the following stages:

- consumer polling record: gather *batch of consumer records*
- convert: convert *records individually* to generic `SinkRecord` ,
- transform: apply transformation chain
- process: put *record batches* into a external system.

The processing stage already has a latency metric: `put-batch-time`

To measure sink's record latency (i.e. processing time - event time), it's proposed to measure the difference between record timestamp and current system time (wall-clock) just before the convert stage as it is when records are iterated already.

Convert latency `convert-sink-record-time` and `transform-chain-sink-record-time` measures the convert and transformation per-record.

Polling can be monitored with Consumer fetch metrics, e.g. `fetch-latency-avg/max`

Predicates as implemented via `PredicatedTransformation` will be also measured.

The per-record metrics will definitely be added to Kafka Connect as part of this KIP, but their metric level will be changed pending the performance testing described in Unable to render Jira issues macro, execution error. , and will otherwise only be exposed at lower level (DEBUG instead of INFO, and TRACE instead of DEBUG)

Public Interfaces

The following metrics would be added at the Task Level:

kafka.connect:type=sink-task-metrics,connector="{connector}",task="{task}"

Sensor / Recording Level	Attribute name	Description
sink-record-latency **INFO**	• sink-record-latency-max-ms	The maximum latency of a record, measured by comparing the record timestamp with the system time (i.e. wallclock) when it has been received by the Sink task right after consumer poll and before conversions.
	• sink-record-latency-avg-ms	The average latency of a record, measured by comparing the record timestamp with the system time (i.e. wallclock) when it has been received by the Sink task right after consumer poll and before conversions.
convert-sink-record-time **INFO**	• convert-sink-record-time-avg-ms	The average time taken by this task to convert sink records, including key, value, and headers conversion.
	• convert-sink-record-time-max-ms	The maximum time taken by this task to convert sink records, including key, value, and headers conversion.
convert-sink-record-key-time **DEBUG**	• convert-sink-record-key-time-avg-ms	The average time taken by this task to convert sink record keys.
	• convert-sink-record-key-time-max-ms	The maximum time taken by this task to convert sink record keys.
convert-sink-record-value-time **DEBUG**	• convert-sink-record-value-time-avg-ms	The average time taken by this task to convert sink record values.
	• convert-sink-record-value-time-max-ms	The maximum time taken by this task to convert sink record values.
convert-sink-record-headers-time **DEBUG**	• convert-sink-record-headers-time-avg-ms	The average time taken by this task to convert sink record headers.

	• convert-sink-record-headers-time-max-ms	The maximum time taken by this task to convert sink record headers.
transform-chain-sink-record-time **INFO**	• transform-chain-sink-record-time-avg-ms	The average time taken by this task to apply all the transforms included in this task.
	• transform-chain-sink-record-time-max-ms	The maximum time taken by this task to apply all the transforms included in this task.

kafka.connect:type=source-task-metrics,connector="{connector}",task="{task}"

Sensor / Recording Level	Attribute name	Description
convert-source-record-time **INFO**	• convert-source-record-time-avg-ms	The average time taken by this task to convert source records, including key, value, and headers conversion.
	• convert-source-record-time-max-ms	The maximum time taken by this task to convert source records, including key, value, and headers conversion.
convert-source-record-key-time **DEBUG**	• convert-source-record-key-time-avg-ms	The average time taken by this task to convert source record keys.
	• convert-source-record-key-time-max-ms	The maximum time taken by this task to convert source record keys.
convert-source-record-value-time **DEBUG**	• convert-source-record-value-time-avg-ms	The average time taken by this task to convert source record values.
	• convert-source-record-value-time-max-ms	The maximum time taken by this task to convert source record values.
convert-source-record-headers-time **DEBUG**	• convert-source-record-headers-time-avg-ms	The average time taken by this task to convert source record headers.
	• convert-source-record-headers-time-max-ms	The maximum time taken by this task to convert source record headers.
transform-chain-source-record-time **INFO**	• transform-chain-source-record-time-avg-ms	The average time taken by this task to apply all the transforms included in this task.
	• transform-chain-source-record-time-max-ms	The maximum time taken by this task to apply all the transforms included in this task.

kafka.connect:type=sink-task-transform-metrics,connector="{connector}",transform="{transform_alias}",task="{task}"

Sensor / Recording Level	Attribute name	Description
transform-sink-record-time (?) **DEBUG**	• transform-sink-record-time-avg-ms	The average time taken by this task to apply specific transform included in this task.
	• transform-sink-record-time-max-ms	The maximum time taken by this task to apply specific transform included in this task.

kafka.connect:type=source-task-transform-metrics,connector="{connector}",transform="{transform_alias}",task="{task}"

Sensor / Recording Level	Attribute name	Description
transform-source-record-time **DEBUG**	• transform-source-record-time-avg-ms	The average time taken by this task to apply specific transform included in this task.
	• transform-source-record-time-max-ms	The maximum time taken by this task to apply specific transform included in this task.

Where `alias` is the Transform alias name used in configuration:

```
"transforms": "routeRecords",
"transforms.routeRecords.type": "org.apache.kafka.connect.transforms.RegexRouter"
```

"routeRecords" in this example.

More granular metrics are recorded at **DEBUG** level to avoid performance impact.

`TransformationChain` and `ConnectorConfig` will change their following APIs to support keeping transform alias to record metrics:

ConnectorConfig:

```
-    public <R extends ConnectRecord<R>> List<Transformation<R>> transformations() {
+    public <R extends ConnectRecord<R>> LinkedHashMap<String, Transformation<R>> transformations() {
```

TransformChain:

```
-    public TransformationChain(List<Transformation<R>> transformations, RetryWithToleranceOperator
retryWithToleranceOperator) {
+    public TransformationChain(LinkedHashMap<String, Transformation<R>> transformations,
RetryWithToleranceOperator retryWithToleranceOperator) {
```

Compatibility, Deprecation, and Migration Plan

ConnectConfig and TransformationChain users will have to migrate to the new interfaces. Though these APIs are used internally on Worker instantiations of Tasks and not meant for external usage.

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.