

# KIP-281: ConsumerPerformance: Increase Polling Loop Timeout and Make It Reachable by the End User

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

## Status

**Current state:** *Accepted*

**Discussion thread:** [\[DISCUSS\] KIP-281: ConsumerPerformance: Increase Polling Loop Timeout and Make It Reachable by the End User](#)

**Vote thread:** [\[VOTE\] KIP-281: ConsumerPerformance: Increase Polling Loop Timeout and Make It Reachable by the End User](#)

**Vote results:**

- Binding +4: Jason Gustafson, Ismael Juma, Rajini Sivaram and Guozhang Wang
- Non-binding +4: Ted Yu, Moshe Blumberg, Alex Dunayevsky, and Dhruvil Shah
- No -1 votes.

JIRA:



Unable to render Jira issues macro, execution error.

**PULL REQUEST:** <https://github.com/apache/kafka/pull/4818>

## Motivation

ConsumerPerformance fails to consume all messages on topics with large number of partitions due to a relatively short default polling loop timeout (1000 ms) that is not reachable and modifiable by the end user.

**Demo:** Create a topic of 10 000 partitions, send a 50 000 000 of 100 byte records using *kafka-producer-perf-test* and consume them using *kafka-consumer-perf-test* (ConsumerPerformance). You will likely notice that the number of records returned by the *kafka-consumer-perf-test* is many times less than expected 50 000 000.

As the result, in some rough cases, it may take a long enough time to process/iterate through the records polled in batches, thus the time may exceed the default hardcoded polling loop timeout (1 second), forcing the utility to stop it's execution much earlier than it should to.

This leads to confusion, poor and unexpected test results and is probably not what we want. Thus:

- Increasing polling loop timeout is probably a good thing to handle this case
- Increasing it too much by default (say, 5 minutes) may lead to a slower feedback from the tool in some cases
- Making polling loop timeout optionally changeable is a good thing since it would give us more control for the corner cases
- Printing a warning message in case the timeout is exceeded should also be a good thing to know what's going on

## Public Interfaces

Providing the command line tool **kafka-consumer-perf-test.sh** with an optional **--timeout** parameter:

parameter name	defaults to	type	time unit	usage
--timeout	10000	Long	ms	optional

## Proposed Changes

- Increasing polling loop timeout in ConsumerPerformance implementation up to 10 000 ms
- Making polling loop timeout reachable and modifiable by the end user from the command line *kafka-consumer-perf-test* tool

- Print a warning message if the polling loop timeout is exceeded

**Proposed changes in code available in this PR:** <https://github.com/apache/kafka/pull/4818>

## Compatibility, Deprecation, and Migration Plan

- Positive impact on existing users since the utility generates more accurate results
- The proposed parameter is optional, will not break anything

## Rejected Alternatives

None.