IBM Event Streams Performance Report

9 August 2019

For Event Streams 2019.2.1 running on IBM Cloud Private 3.1.2

Disclaimer

DISCLAIMERS

The performance data contained in this report was measured in a controlled environment. Results obtained in other environments may vary significantly.
You should not assume that the information contained in this report has been submitted to any formal testing by IBM.
Any use of this information and implementation of any of the techniques are the responsibility of the licensed user. Much depends on the ability of the licensed user to evaluate the data and to project the results into their own operational environment.

WARRANTY AND LIABILITY EXCLUSION

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.
In Germany and Austria, notwithstanding the above exclusions, IBM's warranty and liability are governed only by the respective terms applicable for Germany and Austria in the corresponding IBM program license agreement(s).

ERRORS AND OMISSIONS

The information set forth in this report could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; any such change will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time and without notice.

LOCAL AVAILABILITY

References in this report to IBM products or programs do not imply that IBM intends to make these available in all countries in which IBM operates. Consult your local IBM representative for information on the products and services currently available in your area.

ALTERNATIVE PRODUCTS AND SERVICES

Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

TRADEMARKS AND SERVICE MARKS

The following terms used in this publication are trademarks of their respective companies in the United States, other countries or both:
- IBM Corporation : IBM
Other company, product, and service names may be trademarks or service marks of others.

EXPORT REGULATIONS

You agree to comply with all applicable export and import laws and regulations.

Overview

This report contains indicative numbers that demonstrate the ability of IBM Event Streams to handle different levels of messaging traffic. The values were achieved using example workloads. It is not a definitive guide to peak performance capabilities, but aims to show what Event Streams can handle based on examples. Performance will always depend on numerous factors including message throughput, message size, hardware, configuration settings, and so on.

This report will be updated on a regular basis. This is the first version.

Testing was based on Event Streams version 2019.2.1 running on IBM Cloud Private 3.1.2.

Workloads

The table below illustrates the workloads measured during the creation of this report.

Workload	Description
Resilient	Demonstrates scaling as cluster size increases, favoring resilience over throughput, 3-way replication, all acknowledgements
Fast	Demonstrates scaling as cluster size increases, favoring throughput over resilience, no replication, leader acknowledgements
Payload size	Demonstrates the effect of message size on throughput

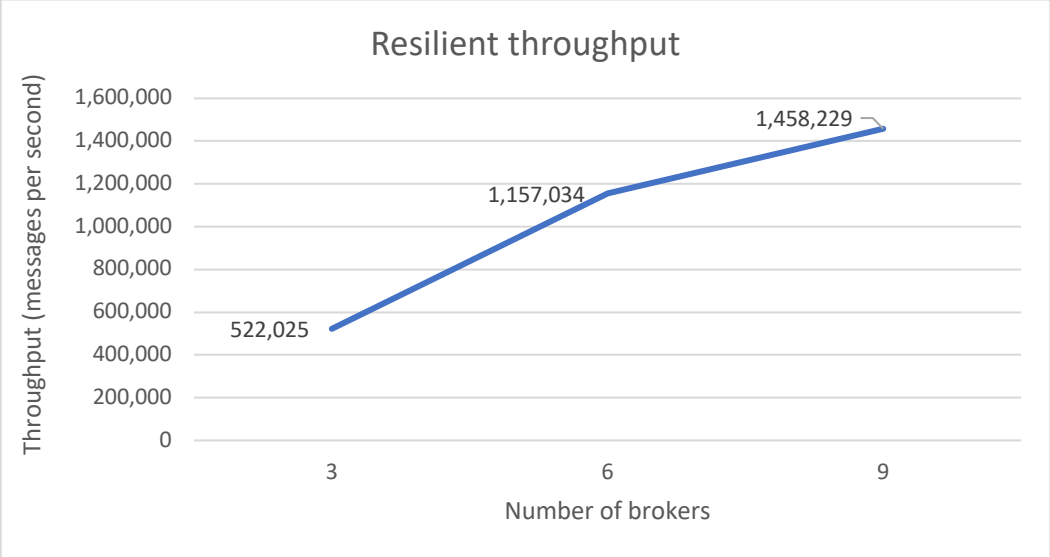
Workload – Resilient

This workload measures throughput at increasing cluster sizes favoring resilience over throughout:

- Replication and acknowledgements protect messages against failures
- Messages are replicated across 3 brokers (replication factor 3)
- Acknowledgements from all brokers (acks=all)

A single topic is used for the messages. The number of partitions is double the number of brokers and the partition leadership and replicas are spread evenly across the brokers. The message payload is 128 bytes long. All consumers are in the same consumer group and each message is consumed by one consumer. The workload is generated by the Apache Kafka kafka-producer-perf-test.sh and kafka-consumer-perf-test.sh tools.

Brokers	Producers	Consumers	Partitions	Throughput (messages/second)
3	6	6	6	522,025
6	12	12	12	1,157,034
9	18	18	18	1,458,229



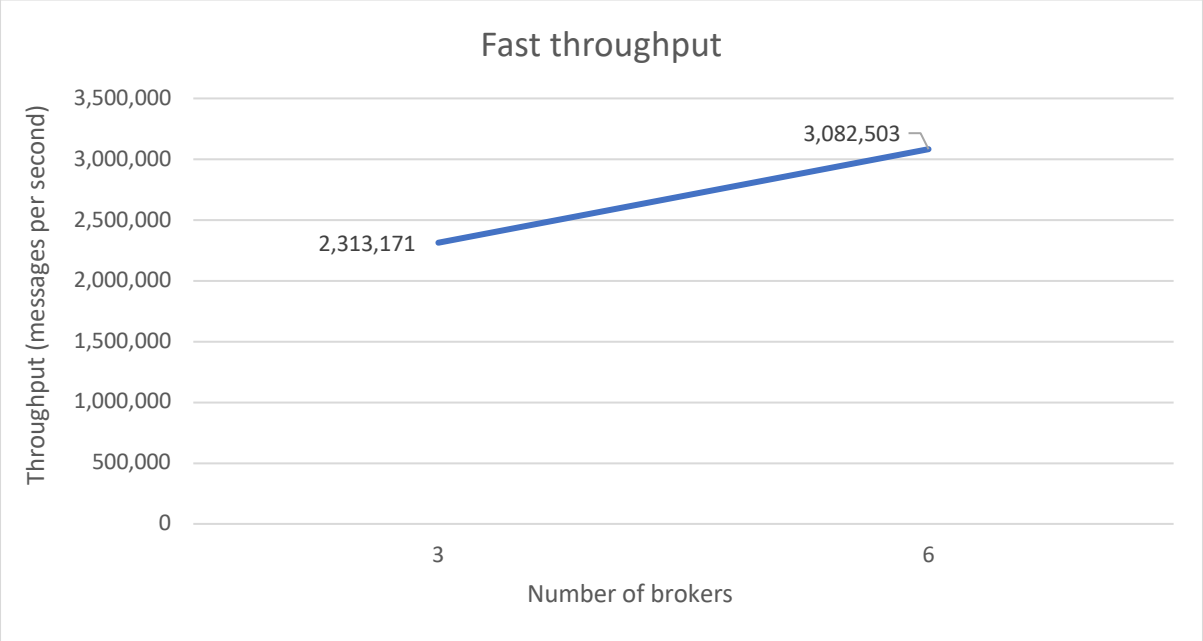
Workload – Fast

This workload measures throughput at increasing cluster sizes, favoring throughput over resilience:

- Lower overhead gives additional throughput but less resilience to failures
- The messages are not replicated across brokers (replication factor 1)
- Acknowledgements from 1 broker (acks=1)

A single topic is used for the messages. The number of producers and consumers are balanced to achieve constant message throughput. The number of partitions is set to match the number of consumers to evenly distribute message processing across the consumer group. The message payload is 128 bytes long. All consumers are in the same consumer group and each message is consumed by one consumer. The workload is generated by the Apache Kafka kafka-producer-perf-test.sh and kafka-consumer-perf-test.sh tools.

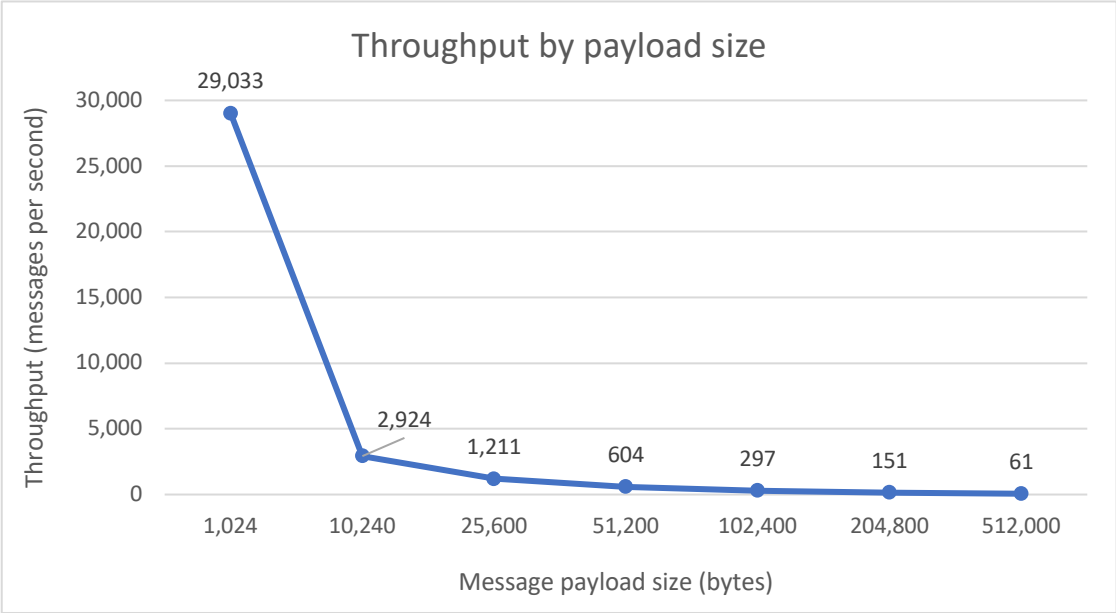
Brokers	Producers	Consumers	Partitions	Throughput (messages/second)
3	12	24	24	2,313,171
6	24	48	48	3,082,503



Workload – Payload size

This test case measures the effect of increasing payload size on throughput. A single topic with one partition and no replication is used for the messages. One producer and no consumers are used. The workload is generated by the Apache Kafka kafka-producer-perf-test.sh tool.

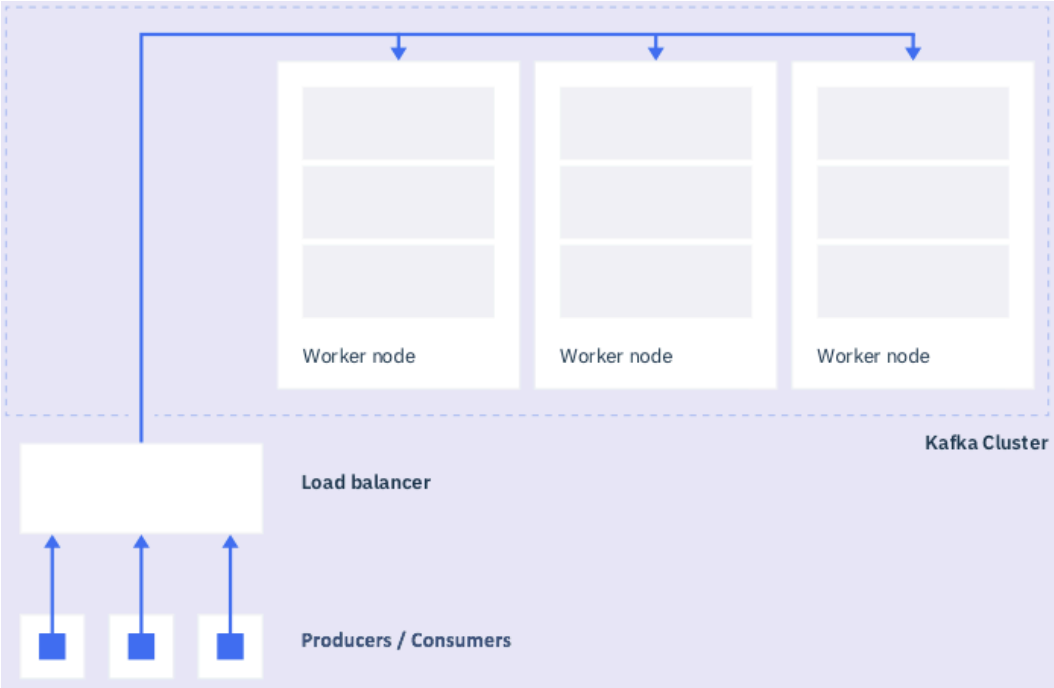
Message size (bytes)	Throughput (messages/second)	Throughput (MB/second)
1,024	29,033	28.36
10,240	2,924	28.56
25,600	1,211	29.56
51,200	604	29.48
102,400	297	29.02
204,800	151	29.49
512,000	61	30



Recommended deployment guidance

The performance of IBM Event Streams is affected by the environment in which it is deployed. For example, the replication of message data between brokers can consume a lot of network bandwidth so isolating replication traffic from application traffic can benefit performance.

The following diagram illustrates the deployment of IBM Event Streams:



Test configurations

Hardware

- IBM BladeCenter blade with 24 x 6 core CPUs: Intel® Xeon® X5670 at 2.93GHz
- 32GB RAM
- 2 x 10Gb ethernet adapters, two separate networks (one for internal traffic and one for external traffic)
- Persistent storage: DS8880 Storage system connected via a DH8 SAN Volume Controller with an 8Gbit fiber connection, mounted via NFS within internal network

Software

IBM Event Streams

- Ubuntu 16.04
- IBM Cloud Private 3.1.2
- IBM Event Streams 2019.2.1 based on Apache Kafka 2.2

Producers and consumers

- Apache Kafka 2.2
- OpenJDK 64-Bit 1.8.0_191

Apache Kafka tuning

The following Apache Kafka tuning parameters were applied:

Name	Value	Explanation
num.replica.fetchers	Same as number of brokers	Increases the I/O parallelism for replication
num.io.threads	24	Increases the I/O parallelism for processing requests
num.network.threads	9	Increases the I/O parallelism for network traffic
log.cleaner.threads	6	Provides enough bandwidth for log deletion to ensure Kafka logs do not exceed available disk space.

Event Streams tuning

The following resource configurations were applied when deploying the helm chart:

Name	Value
kafka.resources.limits.cpu	4000m
kafka.resources.requests.cpu	4000m
kafka.resources.limits.memory	8096Mi
kafka.resources.requests.memory	8096Mi
kafka.metricsProxyResources.limits.cpu	2000m
kafka.metricsProxyResources.requests.cpu	2000m
kafka.metricsProxyResources.limits.memory	3072Mi
kafka.metricsProxyResources.requests.memory	3072Mi
global.resources.collector.limits.cpu	2000m
global.resources.collector.requests.cpu	2000m
global.resources.collector.limits.memory	2000Mi
global.resources.collector.requests.memory	2000Mi
global.resources.proxy.limits.cpu	2000m
global.resources.proxy.requests.cpu	2000m