Optimizing Kafka Performance Across Clusters at Twitter

Twitter processes trillions of events at the petabyte scale every day, using nearly 180 Kafka clusters in production to process data. A collaboration and optimization engagement with Intel and Twitter led to 1.25x[1] increase in throughput per core and 1.35x[1] increase in the number of connections per core.

Authors

Justin Long
Public Cloud Technologist, Intel

Yiming Zhang
Sr. Software Engineer, Twitter

Rodolfo De Vega
Cloud Software Engineer, Intel

Shylaja Kokoori
Software Engineer, Intel

Dory Simaan
Industry Technical Specialist, Intel

About Twitter, Inc.

Twitter, a microblogging and social networking service, connects billions of people around the world using short messages called Tweets. Twitter leverages Apache Kafka to process an incredible scale of data:

- As of May 2022, 10,033 tweets per second or 867 million tweets per day[2]
- Trillions of platform events per day at the petabyte scale
- Approximately 180+ Kafka clusters, the largest of which is up to 200 brokers

Twitter's microblogging and social networking service is unique in how it leverages real-time data for multiple business purposes, supporting various functions including data logging, timeline predictions, user analysis, machine learning and analytics, ad recommendations, and notifications. Twitter extensively uses Kafka to process the trillions of events generated by its platform for their real-time pipelines. Because every incremental performance gain has significant impact at scale, Twitter collaborated with Intel to optimize its Kafka clusters to increase throughput and increase the maximum number of connections.

Challenges

Running one of the largest usages of Kafka in the world comes with its own unique set of problems that can affect performance. Intel collaborated with Twitter on several challenges including:

Kafka broker-cluster size limitation. Increasing throughput while maintaining SLA on broker clusters is challenging due to certain limitations in Kafka 2.7. Encryption is universal when communicating across services but adds processing overhead. TLS cryptographic protocol uses AES-GCM which can be CPU intensive.

TLS 1.3 regression in Kafka 2.7. TLS 1.3 does not support renegotiation in Kafka 2.7, creating intermittent disconnections in brokers.

Understanding available features in Intel processors in the cloud. Software developers, in general, may not be completely aware of all features in Intel CPUs and not aware of Intel's optimizations contributed to open source software.

Solutions

Throughput performance (while maintaining latency SLA) of Kafka brokers improved by optimizing encryption with 3rd generation Intel® Xeon® Scalable processors (codename Ice Lake) with support for Vector Advanced Encryption Standard (VAES) and enabling these features in the corresponding version of OpenJDK. Since OpenJDK 11.0.11, support for VAES has been enabled. Twitter has since upgraded to OpenJDK 11.0.15, reducing the overhead caused by encryption.

Combined usage of VTune™ Profiler and Kafka logging of broker cluster performance, allowing identification of a bug in the Kafka source code responsible for the TLS 1.3 regression. Intel engineers submitted a patch and worked with the community to upstream in Apache Kafka.

Continued collaboration between Intel and Twitter engineers, allowing Twitter engineers to further optimize workloads. The ongoing collaboration showed the Twitter engineering team how to take advantage of accelerations in Intel hardware. Intel is continually making contributions and optimizations to open source software through informed collaborations with customers like Twitter.

Results

Over 9 months of collaboration, incremental improvements through performance testing, and engineering, internal testing achieved an overall 1.25x[1] increase in throughput per core, and 1.35x[1] increase in the maximum number of broker connections per core.

The following data was collected from Twitter engineering's test environment using a 10-node, 1 broker-per-node setup with 1 topic, 50 partitions, 10,000 producers, and 500 bytes per message:

Upgrade of Instance Type. Performance testing of newer instance types demonstrated significant benefits, and allowed Kafka to take advantage of performance advantages in 3rd Gen Intel Xeon Scalable processors. These instance types and their naming are unique to Twitter's own data center.

Instance JDK	Cores	Max throughput per broker
F5[3] JDK 11.0.15	52	300 MB/s per broker 5.8 MB/s per core
F6[4] JDK 11.0.15	72	450 MB/s per broker 6.25 MB/s per core
		1.19x increase[1]

Upgrade of Java runtime. Intel contributions to OpenJDK 11.0.15 have enabled specific cryptographic accelerations, taking advantage of VAES supported features on 3rd Gen Intel Xeon Scalable processors not available prior to OpenJDK version 11.0.8.

Instance JDK	Cores	Max throughput per broker
F6[4] JDK 8	72	350 MB/s per broker 4.9 MB/s per core
F6[4] JDK 11.0.15	72	450 MB/s per broker 6.25 MB/s per core
		1.27x increase[1]

Combined results

Instance JDK	Cores	Max throughput per broker	Max number of connections
F5[3] JDK 8	52	260 MB/s per broker 5.0 MB/s per core	80,000 1538.5 per core
F6[4] JDK 11.0.15	72	450 MB/s per broker 6.25 MB/s per core	150,000 2083.3 per core
		1.25x increase[1]	**1.35x increase**[1]

Lessons Learned

The collaboration between Intel and Twitter is a modern success story of hardware and software engineers working together to improve application performance. Intel is continually taking feedback from Twitter and engineering new features into upcoming generations of processors.

Where to Get More Information

IT@Intel built a modern and scalable cyber intelligence platform with Apache Kafka based on Confluent Platform. A white paper is available at:

https://www.intel.com/content/www/us/en/it-management/intel-it-best-practices/modern-scalable-cyber-intelligence-platform-kafka.html

Solution Ingredients

Apache Kafka 2.7, 3rd Gen Intel Xeon Scalable processor, VTune Profiler 2022.2.0.172, OpenJDK 11.0.15, TLS v1.3, crypto, encryption, cloud instance type.

intel.