Apache Kafka

From Wikipedia, the free encyclopedia

Apache Kafka is an open-source stream processing platform developed by the Apache Software Foundation written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. Its storage layer is essentially a "massively scalable pub/sub message queue architected as a distributed transaction log,"[3] making it highly valuable for enterprise infrastructures to process streaming data. Additionally, Kafka connects to external systems (for data import/export) via Kafka Connect and provides Kafka Streams, a Java stream processing library.

The design is heavily influenced by transaction logs.[4]

Apache Kafka [1]

Developer(s)	Apache Software Foundation
Initial release	January 2011[2]
Stable release	0.10.2.1 / April 26, 2017
Repository	git-wip-us.apache.org/repos/asf/kafka.git
Development status	Active
Written in	Scala, Java
Operating system	Cross-platform
Type	Stream processing, Message broker
License	Apache License 2.0
Website	kafka.apache.org

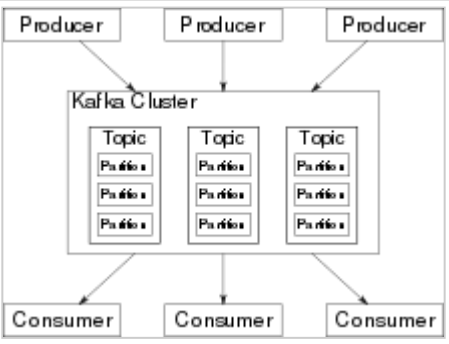
Contents

- 1 History
- 2 Description
- 3 Kafka performance
- 4 Enterprises that use Kafka
- 5 See also
- 6 References
- 7 External links

History

Apache Kafka was originally developed by LinkedIn, and was subsequently open sourced in early 2011. Graduation from the Apache Incubator occurred on 23 October 2012. In November 2014, several engineers who worked on Kafka at LinkedIn created a new company named Confluent[5] with a focus on Kafka. It was named by Jay Kreps after the author Franz Kafka, since it is "a system optimized writing" and he liked Kafka's work.[6]

Description



Overview of Kafka

Kafka stores Messages which come from arbitrary many Processes called "Producers". The Data can thereby be partitioned in different "partitions" within different "topics". Within a partition the messages are indexed and stored together with a timestamp. Other processes called "Consumers" can query messages from partitions. Kafka runs on a cluster of one or more servers and the partitions can be distributed across cluster nodes.

Kafka performance

Due to its widespread integration into enterprise-level infrastructures, monitoring Kafka performance at scale has become an increasingly important issue. Monitoring end-to-end performance requires tracking metrics from brokers, consumer, and producers, in addition to monitoring ZooKeeper which is used by Kafka for coordination among consumers.[7][8] There are currently several monitoring platforms to track Kafka performance, either open-source, like LinkedIn's Burrow, or paid, like Datadog. In addition to these platforms, collecting Kafka data can also be performed using tools commonly bundled with Java, including JConsole.[9]

Enterprises that use Kafka

The following is a list of notable enterprises that have used or are using Kafka:

- Betfair[10]
- Cisco Systems[11]
- CloudFlare[12]
- Conviva[13]
- Daumkakao[14]
- eBay[15]
- Hyperledger Fabric[16]
- HubSpot[17]
- Netflix[18]
- PayPal[19]
- Shopify[20]
- Sift Science[21]
- Spotify[22]
- Ticketmaster[23]
- Uber[24]
- Walmart[25]

See also

- Apache ActiveMQ
- Apache Flink
- Apache Qpid
- Apache Samza
- Apache Spark Streaming
- Data Distribution Service
- Enterprise Integration Patterns
- Enterprise messaging system
- Streaming analytics
- Event-driven SOA
- Message-oriented middleware
- Service-oriented architecture
- StormMQ

References

1. "Mirror of Apache Kafka at GitHub]"(https://github.com/apache/kafka). *github.com*. Retrieved 6 March 2017.
2. "Open-sourcing Kafka, LinkedIn's distributed message queue"(https://blog.linkedin.com/2011/01/11/open-source-linkedin-kafka). Retrieved 27 October 2016.
3. Monitoring Kafka performance metrics(https://www.datadoghq.com/blog/monitoring-kafka-performance-metrics) Datadog Engineering Blog, accessed 23 May 2016/
4. The Log: What every software engineer should know about real-time data's unifying abstraction(http://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-datas-unifying) LinkedIn Engineering Blog, accessed 5 May 2014
5. Primack, Dan. "LinkedIn engineers spin out to launch 'Kafka' startup Confluent"(http://fortune.com/2014/11/06/linkedin-kafka-confluent/) *fortune.com*. Retrieved 10 February 2015.
6. "What is the relation between Kafka, the writer, and Apache Kafka, the distributed messaging system?" (https://www.quora.com/What-is-the-relation-between-Kafka-the-writer-and-Apache-Kafka-the-distributed-messaging-system) *Quora*. Retrieved 2017-06-12.
7. "Monitoring Kafka performance metrics"(https://www.datadoghq.com/blog/monitoring-kafka-performance-metrics/#broker-metrics). 2016-04-06. Retrieved 2016-10-05.
8. Mouzakitis, Evan (2016-04-06)."Monitoring Kafka performance metrics"(https://www.datadoghq.com/blog/monitoring-kafka-performance-metrics/#toc-why-zookeeper). *datadoghq.com*. Retrieved 2016-10-05.
9. "Collecting Kafka performance metrics - Datadog"(https://www.datadoghq.com/blog/collecting-kafka-performance-metrics/#jconsole). 2016-04-06. Retrieved 2016-10-05.
10. "Exchange Market Data Streaming with Kafka"(http://betsandbits.com/2015/07/22/exchange-market-data-streaming-with-kafka/).

11. "OpenSOC: An Open Commitment to Security"(http://blogs.cisco.com/security/opensoc-an-open-commitment-to-secu ity). *Cisco blog*. Retrieved 2016-02-03.
12. "More data, more data"(https://blog.cloudflare.com/more-data-more-data/)
13. "Conviva home page"(http://www.conviva.com/). *Conviva*. 2017-02-28. Retrieved 2017-05-16.
14. Doyung Yoon. "S2Graph : A Large-Scale Graph Database with HBase"(http://apachebigdata2015.sched.org/event/de6a bfbd8f0b9e66b1c03feb2b9e2078?iframe=yes&w=i:100;&sidebar=yes&bg=no)
15. "Kafka Usage in Ebay Communications Delivery Pipeline"(https://www.youtube.com/watch?v=Vh2RcwtzBRY).
16. "Cryptography and Protocols in Hyperledger Fabric"(https://www.zurich.ibm.com/~cca/talks/20170106-blockchain-rw c.pdf) (PDF). January 2017. Retrieved 2017-05-05.
17. "Kafka at HubSpot: Critical Consumer Metrics"(http://product.hubspot.com/blog/kafka-at-hubspot-part-1-critical-cons mer-metrics/).
18. Cheolsoo Park and Ashwin Shankar."Netflix: Integrating Spark at Petabyte Scale"(http://apachebigdata2015.sched.org/ event/3ztw/netflix-integrating-spark-at-petabyte-scale-cheolsoo-park-netflix-and-ashwin-shankar-netflix).
19. Shibi Sudhakaran of PayPal."PayPal: Creating a Central Data Backbone: Couchbase Server to Kafka to Hadoop and Back (talk at Couchbase Connect 2015)"(http://www.couchbase.com/nosql-resources/presentations/paypal-creating-a-c entral-data-backbone-couchbase-server-to-kafka-to-hadoop-and-back.html). *Couchbase*. Retrieved 2016-02-03.
20. "Shopify - Sarama is a Go library for Apache Kafka"(https://github.com/Shopify/sarama).
21. "Concurrency and At Least Once Semantics with the New Kafka Consumer"(http://blog.siftscience.com/2016/concurre ncy-and-at-least-once-semantics-with-the-new-kafka-consumer/)
22. Josh Baer. "How Apache Drives Spotify's Music Recommendations"(http://apachebigdata2015.sched.org/event/2a65daf 0baa4cfbc227a8cb74a9103a2?iframe=no&w=i:100;&sidebar=yes&bg=no)
23. Patrick Hechinger. "CTOs to Know: Meet Ticketmaster's Jody Mulkey"(http://www.builtinla.com/2015/11/04/ctos-kno w-meet-ticketmasters-jody-mulkey).
24. "Stream Processing in Uber"(http://www.infoq.com/presentations/uber-stream-processing). *InfoQ*. Retrieved 2015-12-06.
25. "Apache Kafka for Item Setup"(https://medium.com/walmartlabs/apache-kafka-for-item-setup-3fe8f4ba5967). *medium.com*. Retrieved 2017-06-12.

External links

- Apache Kafka website
- Discussion of project's design
- Github mirror
- Apache Kafka presentation by Morten Kjetland
- LinkedIn open sourcing announcement

Categories: Enterprise application integration | Free software | Free software programmed in Scala | Java platform | Message-oriented middleware | Service-oriented architecture-related products

---