Apache Kafka và Ứng dụng

Tưởng Thị Xuân Thu [a*]

[a] *Khoa Công nghệ Thông tin, Trường Đại học Ngoại ngữ - Tin học Thành phố Hồ Chí Minh*
[*] *Tác giả liên hệ: Email: thuttxt@huflit.edu.vn*

Tóm tắt

Trong Big Data, một khối lượng khổng lồ dữ liệu được sử dụng. Hai thách thức được đặt ra trong vấn đề này, thách thức đầu tiên là làm thế nào để thu thập khối lượng lớn dữ liệu và thách thức thứ hai là phân tích dữ liệu thu thập được. Apache Kafka [1] là một trong những giải pháp giúp giải quyết hai thách thức trên bằng cách thiết kế hệ thống phân tán xử lý dữ liệu hay tin nhắn quy mô lớn theo thời gian thực với khả năng sao chép và khả năng chịu lỗi cao đảm bảo tính an toàn dữ liệu tránh mất mát dữ liệu. Apache Kafka là phần mềm nguồn mở sử dụng với hai mục đích chính: xây dựng data pipeline hoạt động như một Gateway dùng để thiết lập kênh liên lạc giữa hai hệ thống để nhận luồng dữ liệu stream data theo thời gian thực một cách đáng tin cậy; xây dựng các ứng dụng luồng dữ liệu stream data theo thời gian thực để ánh xạ đến các stream data. Ứng dụng Apache Kafka trong nhiều lĩnh vực, có thể kể đến như Internet of Things (IoT) [2], thương mại điện tử, hệ thống message queue [3]…

Từ khoá: Big Data; Apache Kafka; stream data; IoT; data pipeline; message queue; stream processing; log aggregation; metrics collection; Confluent Platform; Apache Storm; Apache

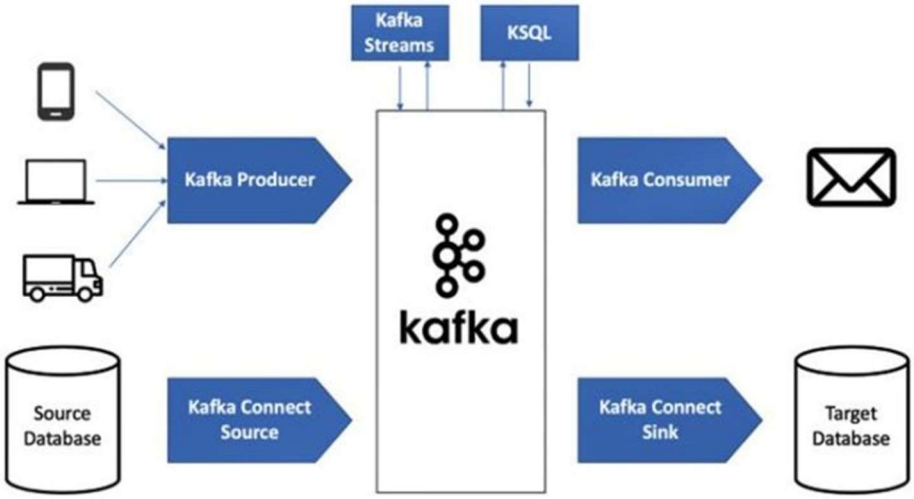
1. GIỚI THIỆU

Kafka được LinkedIn tạo ra vào năm 2011 viết bằng Scala và Java, sau đó Apache phát triển trở thành dự án Apache mã nguồn mở năm 2012, từ đó Kafka có tên là Apache Kafka. Đến hiện tại, Apache Kafka được phân phối chính thức và đầy đủ nhất bởi Confluent thông qua Confluent Platform.

Trong định nghĩa, Apache Kafka là một nền tảng distributed event streaming platform – nền tảng phân tán sự kiện và cũng là distributed messaging system – hệ thống phân tán dữ liệu message/data giữ vài trò chính như một message broker. Thứ nhất, khi đề cập Apache Kafka theo hướng Kafka stream cho mục đích cung cấp dịch vụ xử lý sự kiện theo thời gian thực với độ trễ thấp và thông lượng cao. Thế nên, Apache Kafka được hơn 80% doanh nghiệp trong "Top 100 of Fortune" tin dùng. Thứ hai, Apache Kafka là hệ thống phân tán dữ liệu message/data theo cơ chế public/subscribe, bên gửi - public dữ liệu được gọi là producer và bên nhận - subscribe dữ liệu theo các chủ để Topic sẽ được gọi là consumer, trong trường hợp bên nhận chưa nhận message/data vẫn được lưu trữ sao lưu trên một hàng đợi và cả trên ổ đĩa bảo đảm an toàn, đồng thời nó cũng được sao chép - replicate giúp phòng tránh mất dữ liệu. Nếu có nhu cầu xây dựng một phần mềm, một trang web hiển thị thông tin cho người dùng theo thời gian thực, Apache Kafka chính là một lựa chọn tối ưu. Chúng ta có thể sử dụng Apache Kafka để nhập và lưu trữ dữ liệu trong quá trình phát trực tiếp; hay cũng có thể sử dụng như một phần mềm message broker để kết nối hai ứng dụng/nền tảng giao tiếp với nhau. Những tiện ích chúng ta có thể xem xét sử dụng Apache Kafka:

- Khả năng mở rộng cho phép dữ liệu có thể phân phối trên nhiều máy chủ, điều đó có thể nâng cấp mở rộng khi có nhu cầu mà không cần phải dừng hệ thống.
- Khả năng xử lý tách luồng dữ liệu, vì thế đồ trễ thấp làm cho tốc độ xử lý trở nên nhanh hơn rất nhiều.
- Khả năng chịu lỗi và độ bền cao với các gói dữ liệu có thể được sao chép và phân phối trên nhiều máy chủ server khác nhau. Thế nên, khi có sự cố xảy ra, dữ liệu đã được sao lưu trên các server khác mà không bị lỗi và mất dữ liệu.

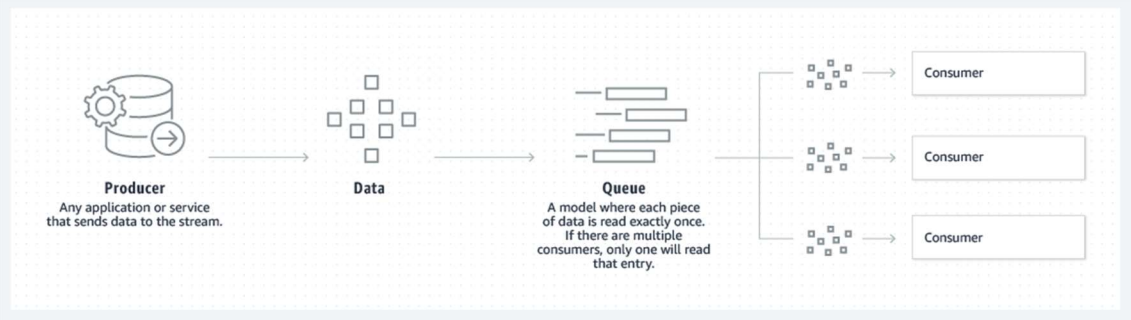
2. KIẾN TRÚC VÀ CƠ CHẾ HOẠT ĐỘNG CỦA APPACHE KAFKA



Hình 1. Mô hình tổng quan Apache Kafka

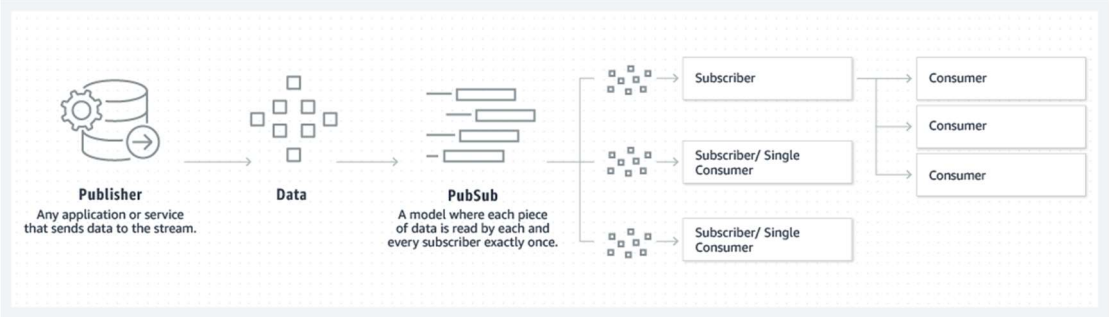
Apache Kafka sử dụng kết hợp hai mô hình chính là queuing và publish-subscribe giúp cung cấp nhiều lợi ích tốt nhất cho người sử dụng.

- Mô hình Queuing – được mô tả qua Hình 2: cho phép dữ liệu được xử lý phân tán trên nhiều máy chủ consumer server và tạo ra khả năng nâng cấp mở rộng cao.



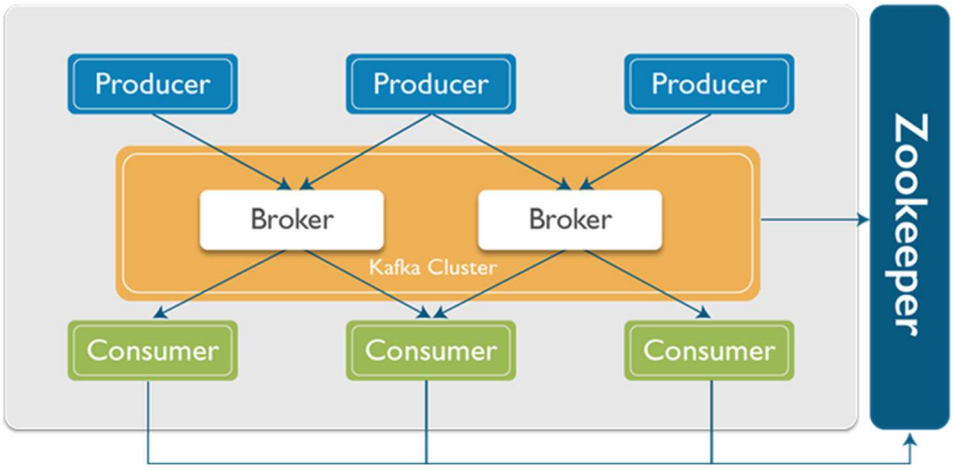
Hình 2. Mô hình Queue

- Mô hình Publish-Subscribe – được mô tả qua Hình 3: tiếp cận cùng lúc nhiều subscribe và các message sẽ được gửi đến nhiều subscribe.



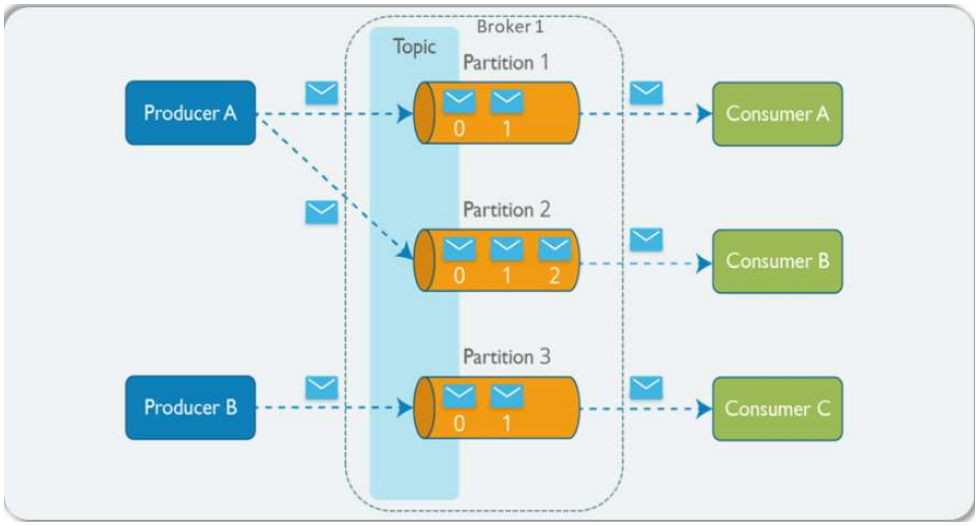
Hình 3. Mô hình Publis-Subscribe

Kiến trúc tổng quát Apache Kafka được biểu diễn qua - Hình 1 cho thấy việc truyền dữ liệu message/data theo Topic, khi cần truyền dữ liệu cho các ứng dụng khác nhau thì sẽ tạo ra các Topic khác nhau. Trước khi xử lý dữ liệu, Apache Kafka sẽ thực hiện phân loại và lưu trữ message dựa theo chủ đề Topic.



Hình 4. Kiến trúc Apache Kafka đơn giản

Nhìn vào Hình 4 – biểu diễn kiến trúc Apache Kafka để thấy rõ chức năng của các thành phần tham gia như Producer có nhiệm vụ lưu, phân loại Message theo Topic, gửi publish dữ liệu message/data vào các chủ đề Topic thích hợp. Sau đó, khi dữ liệu được gửi đến các phần đã được chia thành các Parttion của Topic được lưu trữ trên Broker. Trong đó Kafka Cluster là một nhóm các server và mỗi nhóm server này sẽ được gọi là Broker; Zookeeper được dùng để quản lý và bố trí Broker. Về phần các Consumer sẽ được Apache Kafka sử dụng để đăng ký nhận subscribe vào Topic, chúng được định danh thành từng nhóm theo tên group name, có thể có nhiều Consumer đặt cùng trong một Topic để cùng đọc một Topic. Trong một Kafka Cluster, Partition là nơi lưu trữ dữ liệu cho một Topic. Một Topic có thể có một hay nhiều Partition. Trên mỗi Partition thì dữ liệu lưu trữ cố định và được gán cho một. Mỗi Topic có thể có một hoặc nhiều Partition khác nhau. Mỗi Partition sẽ được gán cho một ID gọi là Offset. Trong một Apache Kafka Cluster biểu diễn qua Hình 5 – Kiến trúc Apache Kafka chi tiết, một Partition có thể sao chép replicate ra nhiều bản. Trong đó có một bản Leader chịu trách nhiệm đọc ghi dữ liệu và các bản còn lại gọi là Follower. Một khi bản Leader bị lỗi thì sẽ có một bản Follower lên làm Leader thay thế. Nếu muốn dùng nhiều Consumer đọc song song dữ liệu của một Topic thì Topic đó cần phải có nhiều Partition.

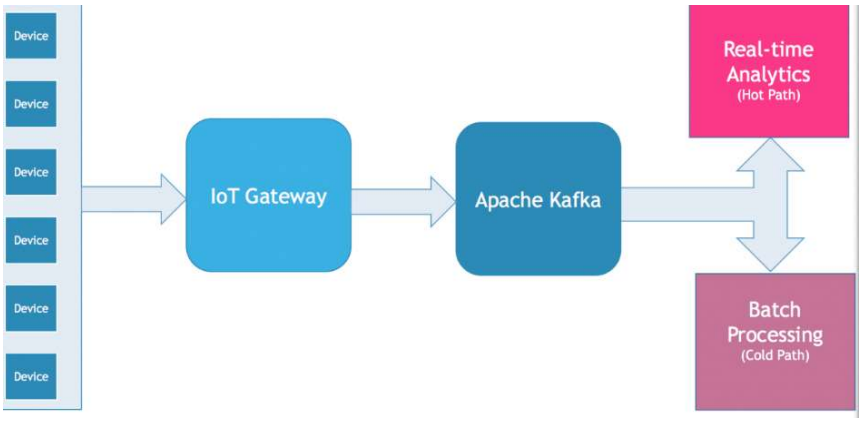


Hình 5. Kiến trúc Apache Kafka chi tiết

3. ỨNG DỤNG CỦA APACHE KAFKA

3.1. Ứng dụng Apache Kafka trong Internet of Things (IoTs)

Nhiều loại cảm biến của thiết bị IoT có khả năng tạo ra nhiều điểm dữ liệu thu thập ở tần suất cao. Một bộ điều nhiệt có thể tạo ra một vài byte dữ liệu mỗi phút trong khi một chiếc xe được kết nối hoặc một tuabin gió tạo ra hàng gigabyte dữ liệu chỉ trong vài giây. Các bộ dữ liệu khổng lồ này được đưa vào pipeline xử lý dữ liệu để lưu trữ, chuyển đổi, xử lý, truy vấn và phân tích. Minh họa với hệ thống sưởi, thông gió và điều hòa không khí (HVAC) được kết nối sẽ báo cáo ghi nhận nhiệt độ môi trường, nhiệt độ mong muốn, độ ẩm, chất lượng không khí, tốc độ quạt và chỉ số tiêu thụ năng lượng. Một ví dụ khác, trong khu mua sắm lớn, những điểm dữ liệu thường được thu thập từ hàng trăm HVAC vì các thiết bị này có thể không đủ mạnh để chạy toàn bộ ngăn xếp TCP, chúng sử dụng các giao thức như Z-Wave và ZigBee để gửi dữ liệu đến một Gateway trung tâm có khả năng tổng hợp các điểm dữ liệu và đưa chúng vào hệ thống. Gateway đẩy dữ liệu được đặt thành cụm Apache Kafka, nơi dữ liệu có nhiều đường dẫn. Các điểm dữ liệu cần được theo dõi trong thời gian thực qua đường dẫn nóng. Trong HVAC, điều quan trọng là theo dõi các số liệu như nhiệt độ, độ ẩm và chất lượng không khí trong thời gian thực để có biện pháp khắc phục. Các điểm dữ liệu này có thể đi qua cụm Apache Storm và Apache Spart để xử lý dữ liệu theo thời gian thực.

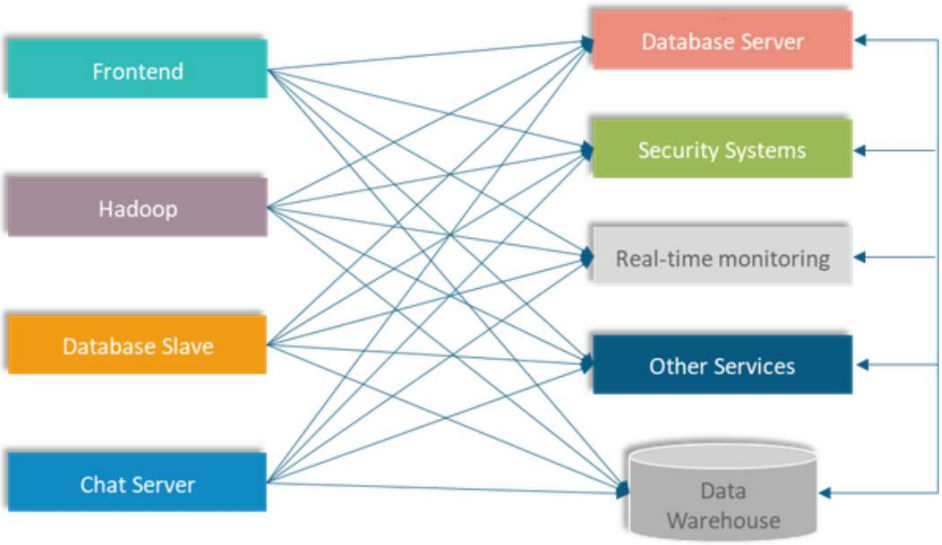


Hình 6. Ứng dụng Apache Kafka trong IoT

Hình 6 – biểu diễn mô hình ứng dụng Apache Kafka trong IoT. Apache Kafka hoạt động như lớp nhập dữ liệu hiệu năng cao xử lý số lượng lớn các tập dữ liệu, thành phần pipeline xử lý dữ liệu chịu trách nhiệm phân tích dữ liệu trực tuyến tức thời – real time analytics và xử lý dữ liệu bach processing chịu trách nhiệm trở thành người đăng ký của Apache Kafka.

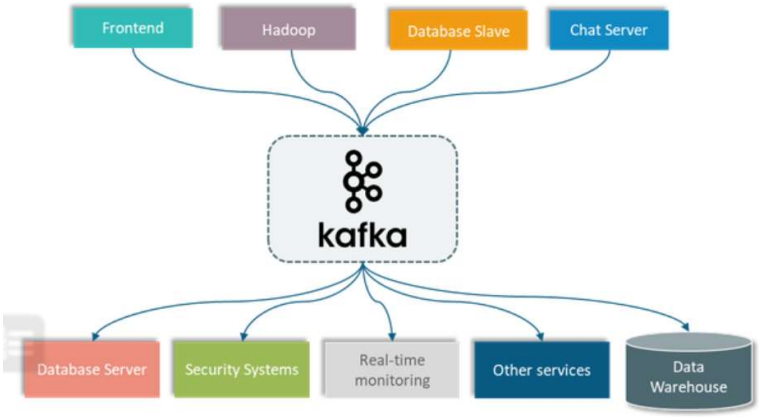
3.2. **Ứng dụng Apache Kafka cho thương mại điện tử**

Một hệ thống thương mại điện tử sẽ có rất nhiều máy chủ server thực hiện nhiều tác vụ khác nhau. Tất cả server này đều sẽ giao tiếp với cơ sở dữ liệu của máy chủ server – database server để đọc ghi dữ liệu. Vì vậy sẽ có rất nhiều data pipeline kết nối từ rất nhiều server khác nhau đến database server này, sẽ làm hệ thống phức tạp khủng khiếp do gia tăng số lượng hệ thống server tham gia vào được mô phỏng theo mô hình trong Hình 7 – Mô hình truyền thống thương mại điện tử.



Hình 7. Mô hình truyền thống hệ thống thương mại điện tử

Giải pháp dùng Apache Kafka để tách rời các data pipeline giữa các hệ thống để làm cho việc giao tiếp giữa các hệ thống trở nên đơn giản hơn và dễ quản lý hơn nhìn rõ trong Hình 8 – mô hình biểu diễn ứng dụng Apache Kafka vào hệ thống thương mại điện tử.



Hình 8. Ứng dụng Apache Kafka vào hệ thống thương mại điện tử

3.3. **Triển khai Apache Kafka đám mây – Cloud Kafka**

Tất cả hệ điều hành có khả năng chạy JVM đều có thể được sử dụng để triển khai cụm Apache Kafka. Apache Kafka được phát triển bằng Java và việc triển khai nó được quản lý bởi Apache ZooKeeper, có thể bắt đầu với dịch vụ Apache Kafka được quản lý trên đám mây – Cloud Kafka được biểu diễn qua Hình 9 – Mô hình Apache Kafka đám mây. IBM Bluemix có Message Hub, một dịch vụ nhắn tin trên đám mây được quản lý hoàn toàn dựa trên Apache Kafka. Cloud Kafka là một nền tảng phát trực tuyến trên đám mây cho Apache Kafka. Aiven.io cung cấp Apache Kafka được lưu trữ cùng với InfluxDB, Grafana và Elaticsearch; Heroku hiện có giúp chúng ta có thể tận dụng Apache Kafka trên Heroku.

Hình 9. Mô hình Apache Kafka đám mây – Apache Kafka Cloud

4. KẾT LUẬN

Apache Kafka là một dự án mã nguồn mở của Apache, đã được đóng gói hoàn chỉnh, có khả năng chịu lỗi cao, hiệu năng rất tốt, dễ dàng mở rộng hệ thống mà không cần dừng hệ thống. Nó đang dần thay thế cho các hệ thống message truyền thống. Có nhiều tiện ích và ứng dụng cao nhờ vào đặt điểm và cơ chế hoạt động của nó, như: một hệ thống message queue thay thế cho ActiveMQ hay RabbitMQ; một hệ thống rất thích hợp trong việc xử lý dòng dữ liệu theo thời gian thực, khi dữ liệu của một Topic được thêm mới ngay lập tức dữ liệu được ghi vào hệ thống và truyền đến bên nhận, dữ liệu có thể được lưu trữ an toàn cho đến khi bên nhận sẵn sàng nhận nó; một hệ thống theo dõi hành động người dùng để gửi public vào một Topic và thông tin đó sẽ được phân tích và xử lý sau; hay lưu lại trạng thái của hệ thống để có thể phục hồi hệ thống khi cần.

TÀI LIỆU THAM KHẢO

[1] https://kafka.apache.org/documentation

[2] https://www.confluent.io.

[3] https://activemq.apache.org