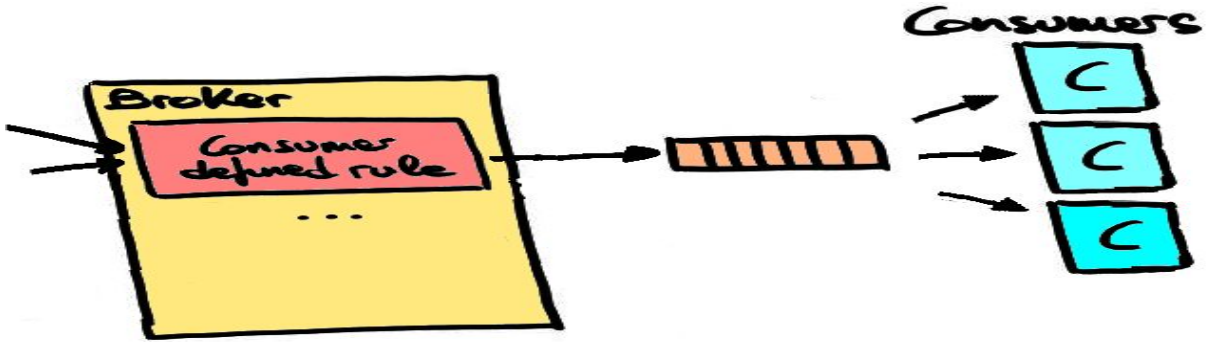
Topic Based Classification Publisher Subscriber System



By: Hanifa Hotelwala, Bijay Ranabhat, and Shahraiz Niazi

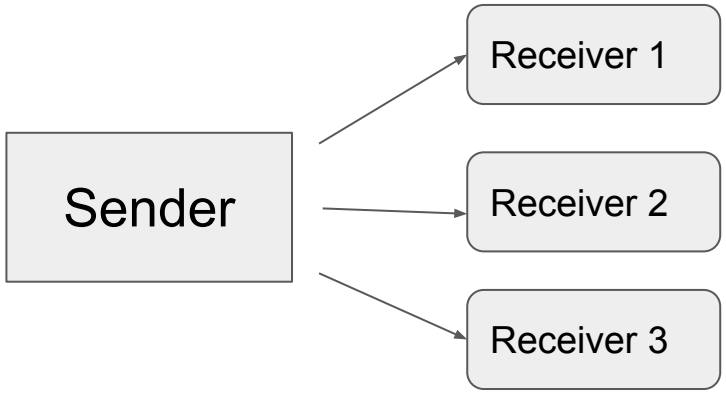
Objectives

- Understand real time data streaming using a topic-based Pub-Sub system
- Understand Publisher-Subscriber design pattern
- Learn benefits of ActiveMQ and Apache Kafka for a Pub-Sub system

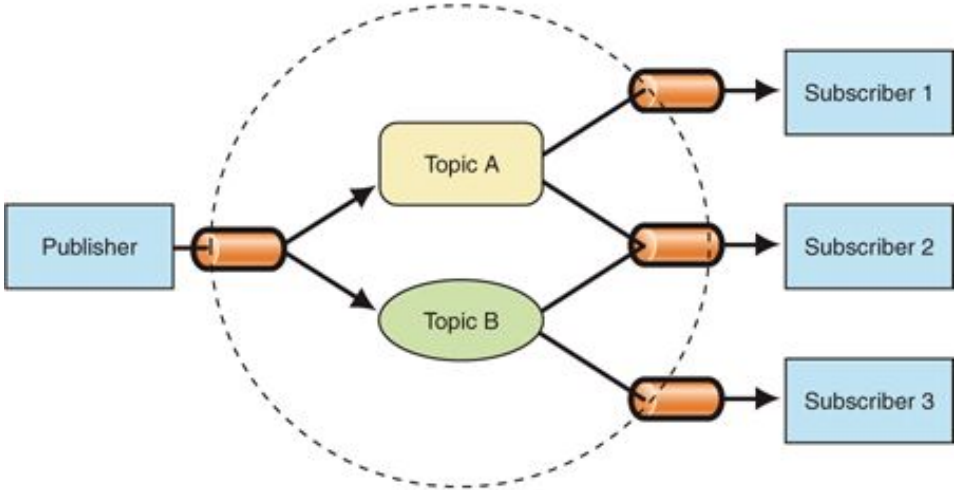
What is a Publisher-Subscriber System?

- Messaging design pattern
- Basic set-up:
  - Sender
  - Receiver
- Improves basic design pattern by introducing scalability and modularity

Basic design pattern

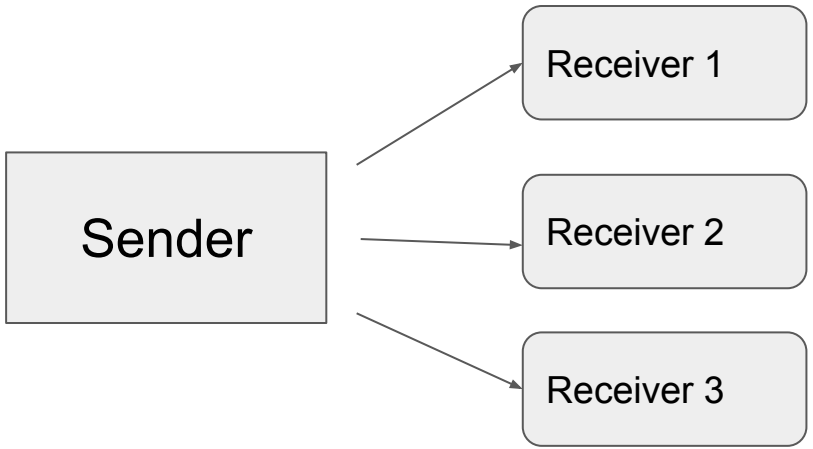


Publisher Subscriber design pattern



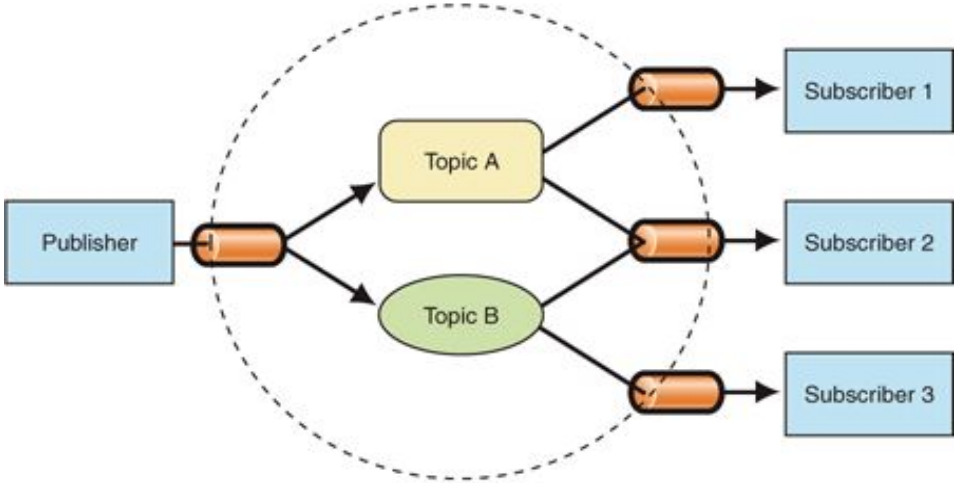
Basic Design Patterns

- Can not send messages without knowing who the receivers are
- Increasing # of senders must be done manually in program
- Not modular
- Not scalable



Solution: Pub-sub system

- Extends the communication infrastructure by including a topic for each message
- Enables listening applications to receive messages from only specified topics
- Creates a mechanism that sends messages to all interested receivers

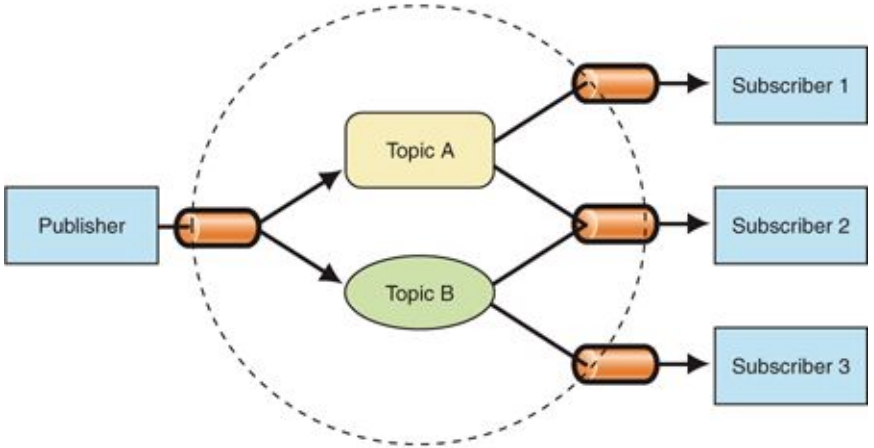


Publisher

- **Producers** publish information on a software bus
  - publish() - generates event
- **Software Bus (PubSub server)**
  - Holds various events
  - Defines a uniform message structure by organizing each event created

Subscriber

- **Consumers** subscribe to the specific information they want to receive from that bus
  - subscribe() / unsubscribe()
- Once subscribed, origin of the sources is unknown and publisher is not notified



Topic-Based Pub-sub System

- **Subscriber**
  - Finds **specific** topics through **keywords**
- **Publishers**
  - Role = same
  - Creates topics/subjects
    - All information is stored in the software bus
- Message Queueing
  - Establishes FIFO scheme
  - Programming abstractions
  - Topic Abstractions
    - Easier to understand
    - Enforces a systematic platform to enable message queueing

Benefits of a Pub-Sub

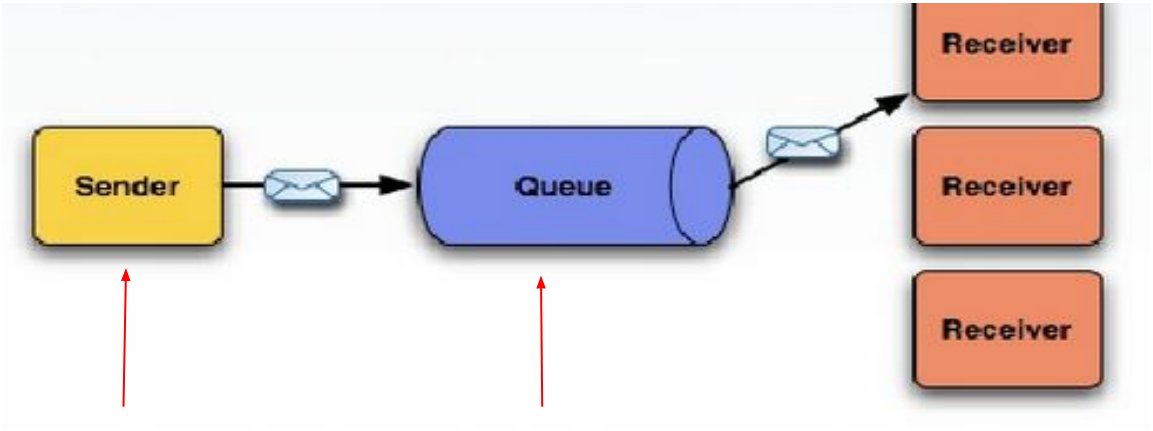
- In modern cloud architecture applications are decoupled into smaller pieces independent from one another
  - Easier to develop, maintain, and deploy
- Provides instant event notification for distributed applications
- Ensures reliability and scalability

Code Implementation - Screencast

3-4 minutes of quick overview of the code and the implemented model

Apache ActiveMQ

- ActiveMQ is an open-source, messaging software which can serve as the backbone for an architecture of distributed applications built upon messaging
- It is used to reliably communicate between two distributed processes



ActiveMQ - use cases

- Supports a variety of Cross Language Clients and Protocols such as Java, C, C++, C#, Ruby, Perl, Python, PHP
- Examples of some use cases
  - Transactional messaging
  - High performance market data distribution
  - Clustering and general purpose async messaging model
  - Web Streaming of data
  - RESTful API to messaging using HTTP

Apache ActiveMQ - Why we use it?

- Allows applications built with different languages and on different operating systems to integrate with each other
- Location transparency
- Reliable communication
- Scaling
- Asynchronous communication
- Reduced coupling

Apache Kafka

- Apache Kafka is a software where topics can be defined
  - A topic is a category name to which messages are stored and published.
- Applications may connect to this system and transfer a message onto the topic.(Publishers)
- Examples of Messages:
  - Information about an event that has happened on your website
  - A simple text message that is supposed to trigger an event
- Another application may connect to the system and process messages from a topic.(Subscribers/Consumers)
- There is a TCP connection between the application and the Kafka Broker.

Apache Kafka

- Distributed messaging system with integrated message queuing system,
- Used for collecting and delivering high volumes of log data with minimal delay,
- Just like a messaging system where you can access data at your own rate whenever,
- Kafka has higher throughput than conventional messaging systems

Apache Kafka

- Uses a partitioning system for every topic (data within the partition is ordered).
- Provides integrated distributed support and can scale out by using consumer groups.
- Multiple consumer groups can read from the same set of topics, and at different times.

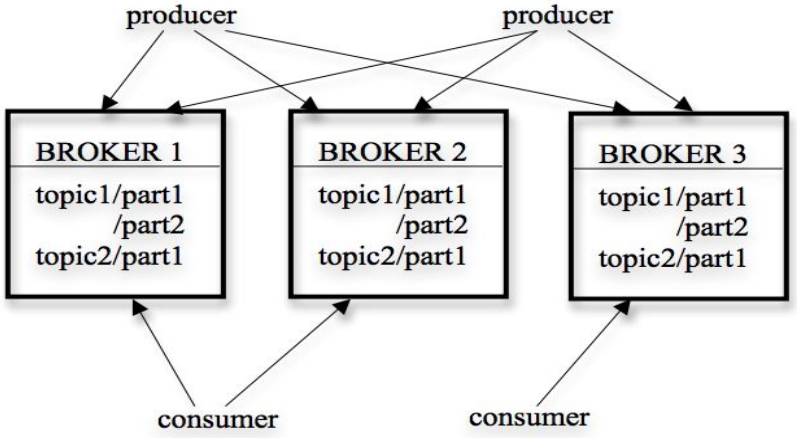
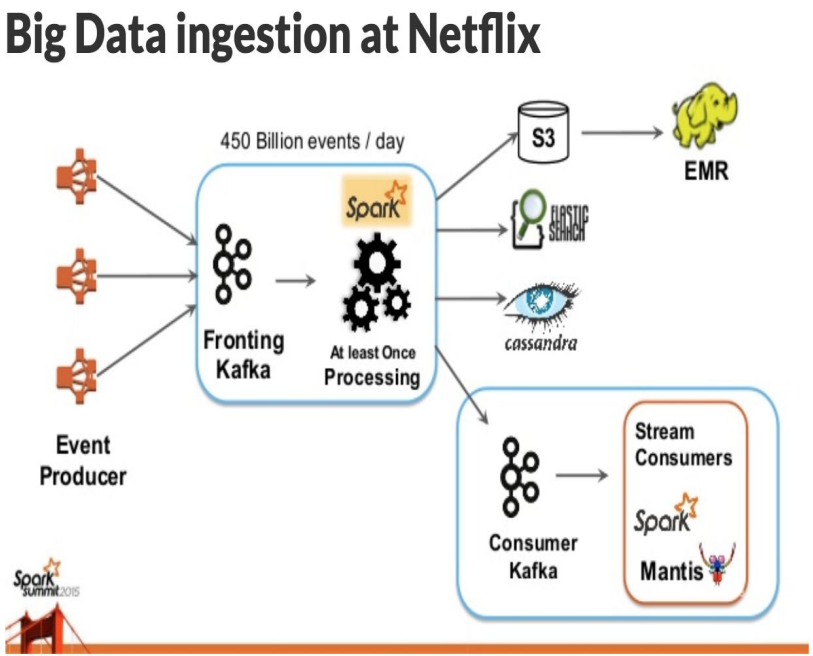


Figure 1. Kafka Architecture

- Used for processing huge volume of log data streams.
- Kafka is very fast and guarantees zero downtime and zero data loss as it has a system of sending acknowledgements and creating replicas of a partition.

Uses of Apache Kafka



- Messaging
  - In comparison to most messaging systems Kafka has better throughput, built-in partitioning, ordering within the partition,replication, and fault-tolerance
- Website Activity Tracking
  - Kafka can track site activity (page views, searches, or other actions users may take) as a set of real time publish-subscribe feeds
- Was originally designed by LinkedIn, used by platforms like LinkedIn, Netflix, etc.

Sources

- Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: a Distributed Messaging System for Log Processing. *ACM SIGMOD Workshop on Networking Meets Databases*, 6. Retrieved from http://pages.cs.wisc.edu/~akella/CS744/F17/838-CloudPapers/Kafka.pdf
- Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. 2003. The many faces of publish/subscribe. ACM Comput. Surv.35, 2 (June 2003), 114-131. DOI=http://dx.doi.org/10.1145/857076.857078
- “Benefits of Pub/Sub Messaging” https://aws.amazon.com/pub-sub-messaging/benefits/

Code Implementation Sources:

Source code to obtain logs
https://social.msdn.microsoft.com/Forums/en-US/3e8486d3-8fee-4477-945f-44d9576a8cb8/c-code-to-read-windows-event-viewer-system-log-in-real-time?forum=csharpgeneral

ActiveMQ
http://activemq.apache.org/getting-started.html
https://tech.lalitbhatt.net/2014/08/activemq-introduction.html
http://activemq.apache.org/
http://blog.christianposta.com/activemq/what-is-activemq/

Dll file
https://www.solvusoft.com/en/files/missing-not-found-error/dll/windows/activemq-apache-org/apache-nms-for-activemq-classhttps://www.solvusoft.com/en/files/missing-not-found-error/dll/windows/activemq-apache-org/apache-nms-for-activemq-class-library/page/1/-library/page/1/
https://remark.wordpress.com/articles/messaging-with-net-and-activemq/