KAFKA CLUSTER SETUP

Đơn vị: Công ty CP Giáo dục và Công nghệ QNET

Quality Network for Education and Technology

QNET JOINT STOCK COMPANY

Address: 14th Floor, VTC Online Tower
18 Tam Trinh Street. Hoang Mai District
Hanoi, Vietnam

Agenda

- Environment Setup
- Installing a Kafka Broker
- Configuring the Broker
- Selecting Hardware
- Configuring Kafka Cluster

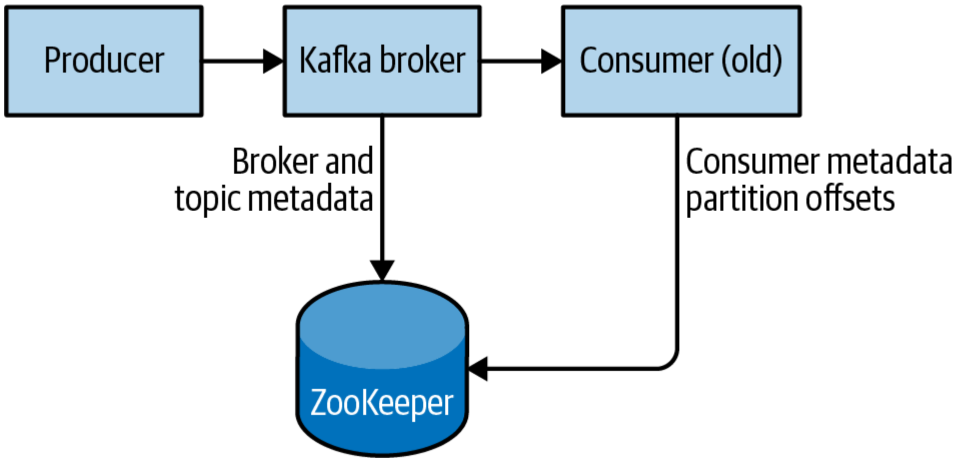
Kafka Cluster Setup

Environment Setup

- Choosing an Operating System: Apache Kafka is a Java application and can run on many operating systems. While Kafka is capable of being run on many OSs, including Windows, macOS, Linux, and others, Linux is the recommended OS for the general use case
- Installing Java: Prior to installing either Zookeeper or Kafka, you will need a Java environment set up and functioning. Kafka and Zookeeper work well with all OpenJDK-based Java implementations, including Oracle JDK. The latest versions of Kafka support both Java 8 and 11

Kafka Cluster Setup

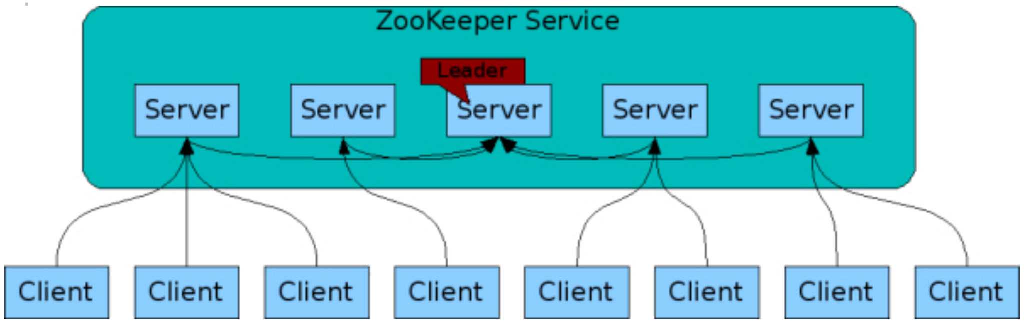
Installing Zookeeper


Producer
Kafka broker
Consumer (old)
Broker and topic metadata
Consumer metadata partition offsets
ZooKeeper

Apache Kafka uses Apache ZooKeeper to store metadata about the Kafka cluster, as well as consumer client details. ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services

Kafka Cluster Setup

Installing Zookeeper: Deployment mode



- Zookeeper Standalone: using for test
- Zookeeper ensemble:
  - using for production
  - Recommend to set up cluster with 5 nodes

Kafka Cluster Setup

Installing Kafka Brokers

- Deployment process:
    - Download the extract binary
    - Configure Kafka brokers
    - Start Kafka brokers
    - Create and verify topic
    - Produce messages to a test topic
    - Consume messages from test topic

Kafka Cluster Setup

Configuring Broker Parameters

- Broker parameters:
    -
    - Listeners
    - zookeeper.connect
    - Log.dirs
    - Num.recovery.threads.per.data.dir
    - Auto.create.topic.enable
    - Auto.leader.rebalance.enable
    - Delete.topic.enable
- Topics:
    - num.partitions
    - default.replication.factor
    -
    - Log.retention.bytes

Kafka Cluster Setup

Selecting Hardware: Disk throughput

- The performance of producer clients will be most directly influenced by the throughout of the broker disk that is used for storing log segments.
    - Faster disk writes will equal lower producer latency
    - Use SSD for very large number of client connections or read-write intensive
    - Use HDD for clusters with very high storage needs but aren’t access open.
    - Could use more disks for better performance

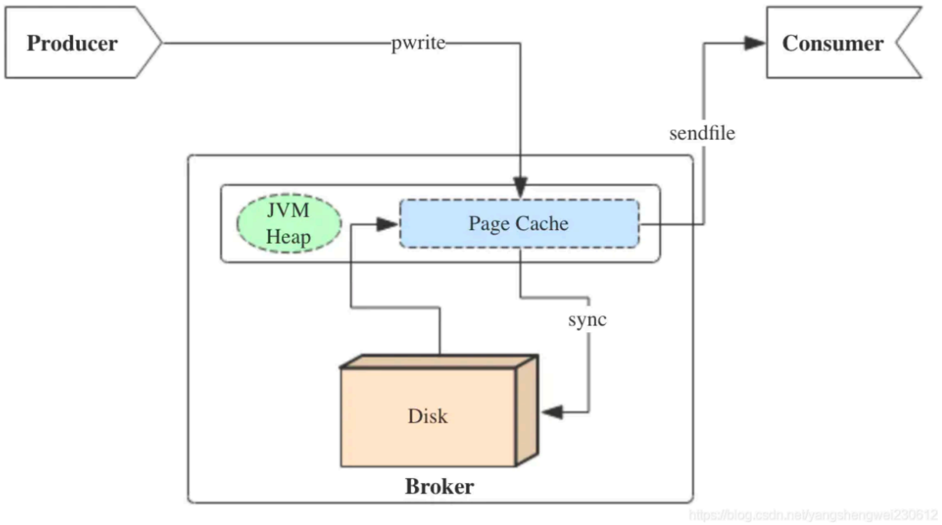
Kafka Cluster Setup

Selecting Hardware: Disk Capacity

- The amount of disk capacity that is needed is determined by how many messages need to be retained at any time.
- Disk capacity was impacted by replication-strategy and the increase of traffic

Kafka Cluster Setup

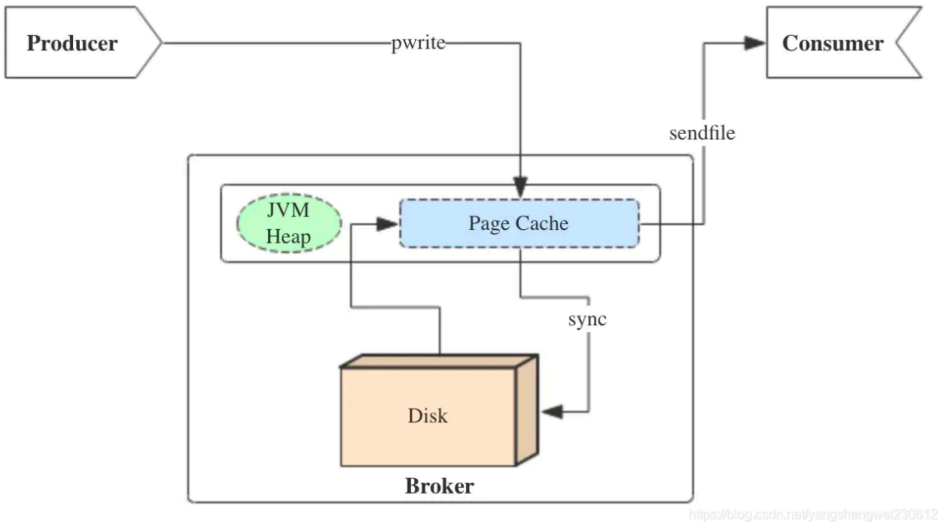
Memory



- When a producer writes messages to the broker, the messages will not immediately persist to disk. The messages will be stored in system pagecache before pushing to disk by OS
- If lagging behind the producers very little, the messages the consumer is reading are optimally stored in the system's page cache
- Having more memory available to the system for page cache will improve the performance of consumer clients

Kafka Cluster Setup

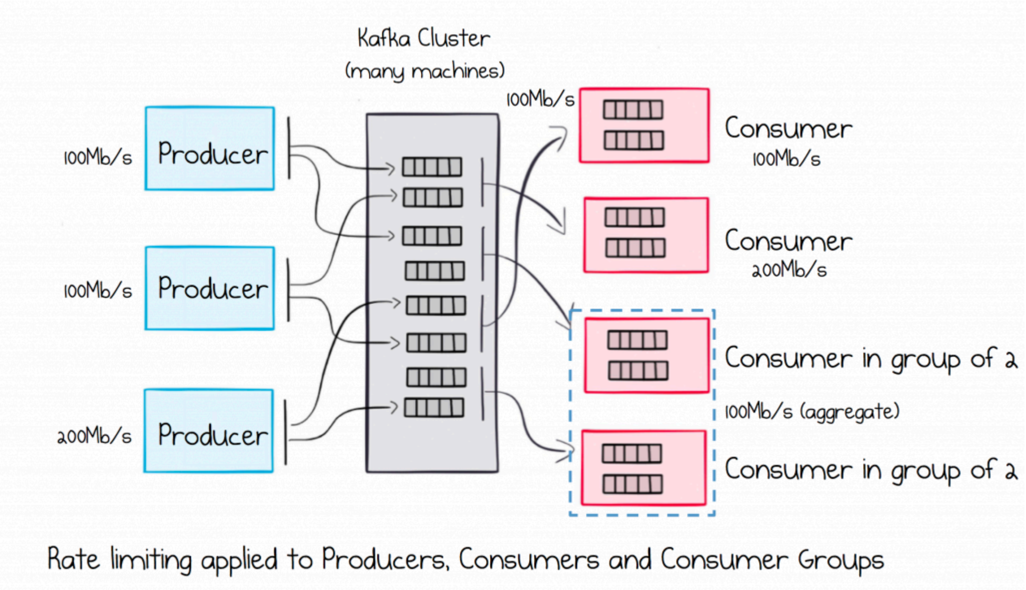
Memory



- Kafka itself does not need much heap memory configured for the Java Virtual Machine. (A broker is handling 150000 messages and data rate of 200 megabits per second can run with a 5GB heap)
- The rest of memory will be used by the page cache.
- => It is not recommend to have Kafka colocated on a system with any other significant application, as it will have to share use of the page cache˜

Kafka Cluster Setup

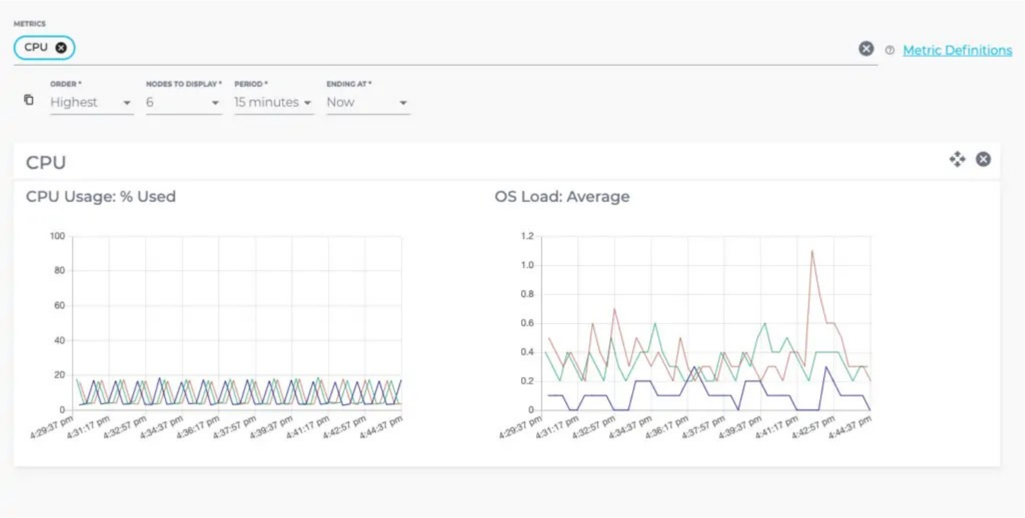
Networking



- The available network throughput will specify the maximum amount of traffic that Kafka can handle
- Network throughput was impacted by the number of producers and consumers and other operations likes: cluster replications and mirroring
- It is recommended to run with at least 10GB NICs. Older machines with 1GB NICs is not recommended or could use Network bonding for better bandwidth

Kafka Cluster Setup

CPU



METRICS: CPU

ORDER *: Highest

NODES TO DISPLAY *: 6

PERIOD *: 15 minutes

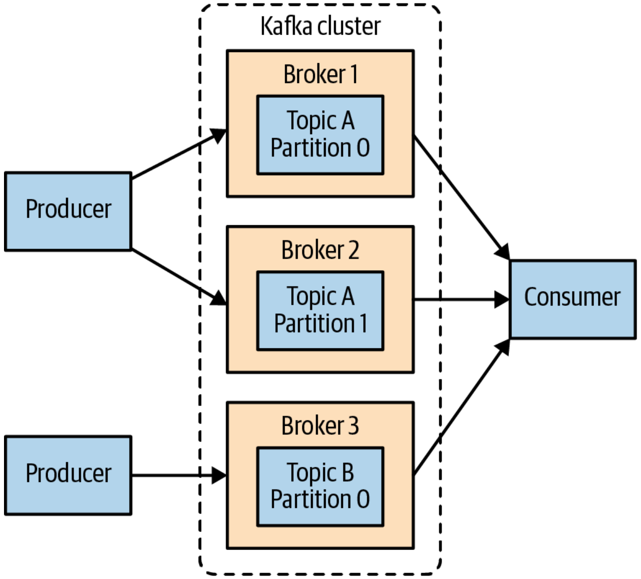
ENDING AT *: Now

Processing power is not as important as disk until Kafka cluster is very large but it will affect overall performance of the broker to some extent.

The most of Kafka's requirement for processing power is for compressing and decompressing message

Kafka Cluster Setup

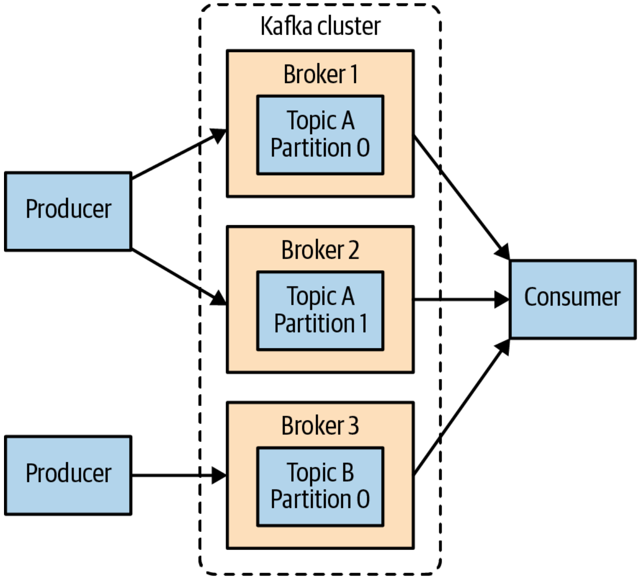
Configuring Kafka Clusters



- A single Kafka broker works well for local development work, or for a proof-of-concept system, but there are significant benefits to having multiple brokers configured as a cluster.
- The biggest benefit is the ability to scale the load across multiple servers.
- Kafka replication will allow for performing maintenance work on Kafka or the underlying systems while still maintaining availability for clients

Kafka Cluster Setup

OS Tuning



- While most Linux distributions have an out-of-the-box configuration for the kernel- tuning parameters that will work fairly well for most applications, there are a few changes that can be made for a Kafka broker that will improve performance.These parameters are typically configured in the */etc/sysctl.conf* file,
- Virtual Memory:
    - vm.swappiness=1
- Disk: The most common choices are:
    - EXT4 (Fourth extended filesystem)
    - XFS (Extends File System)
- Networking:
    - Buffer size:
        - Net.core.wmem_default=131072
        - net.core.rmem_default=131072
        - net.core.wmem_max=2097152
        - net.core.rmem_max=2097152
- Garbage Collector Options:
    - Use G1GC (Garbage-First garbage collector) as the default garbage connector

Kafka Cluster Setup

Garbage Collector

Use G1GC as the default Garbage collector. There are two configuration options for G1GC used to adjust its performance:

`MaxGCPauseMillis`

This option specifies the preferred pause time for each garbage-collection cycle. It is not a fixed maximum—G1GC can and will exceed this time if required. This value defaults to 200 milliseconds. This means that G1GC will attempt to schedule the frequency of garbage collector cycles, as well as the number of zones that are collected in each cycle, such that each cycle will take approximately 200 ms.

`InitiatingHeapOccupancyPercent`

This option specifies the percentage of the total heap that may be in use before G1GC will start a collection cycle. The default value is 45. This means that G1GC will not start a collection cycle until after 45% of the heap is in use. This includes both the new (Eden) and old zone usage, in total.

Kafka Cluster Setup

Demo: Setup a Multiple Node Single Broker

Conclusion

Quality Network for Education and Technology

XIN CHÂN THÀNH CẢM ƠN!