

Flume's Massive Log Aggregation & Kafka's Distributed Messaging System

Prepared by : Hagar Muhamed

Lecture 7





مجموعة جامعة
الجميع الذكية
للتعليم القابضة



Flume's Massive Log Aggregation & Kafka's Distributed Messaging System

Lecture 7

Prepared by
Dr. Hagar Muhamed

www.midocean.ae

Flume: Massive Log Aggregation

- Overview and Architecture
- ▢ Key Features

Kafka: Distributed Messaging System



Chapter 07

Content

What Is Flume?

Flume is a streaming log collection tool that roughly processes data and writes the data to data receivers. It collects data from local files , real-time logs , Thrift, Avro, Syslog, Kafka, and other data sources.

What Can Flume Do?

Collects log information from a fixed directory to a destination (such as HDFS, HBase, or Kafka).

Collects log information to a destination in real time.

Supports cascading (connecting multiple Flumes) and data conflation.

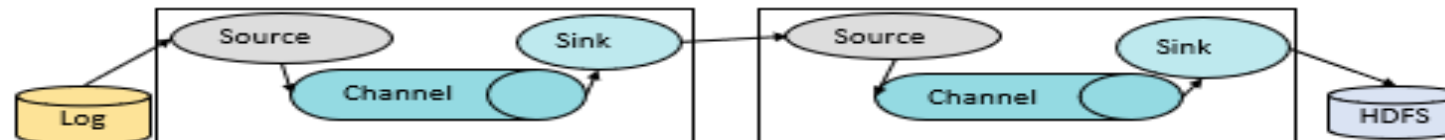
Supports custom data collection tasks based on users.

Flume Architecture

- **Infrastructure:** Flume can directly collect data with an agent, which is mainly for data collection in a cluster.



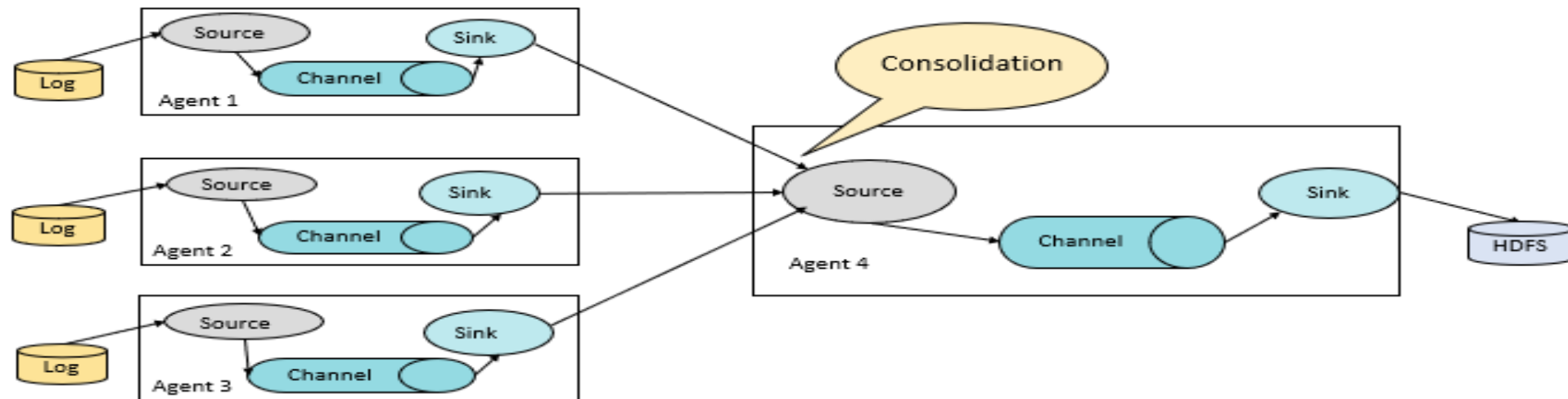
- **Multi-agent architecture:** Flume can connect multiple agents to collect raw data and store it in the final storage system. This architecture is used to import data into the cluster.



Chapter

07

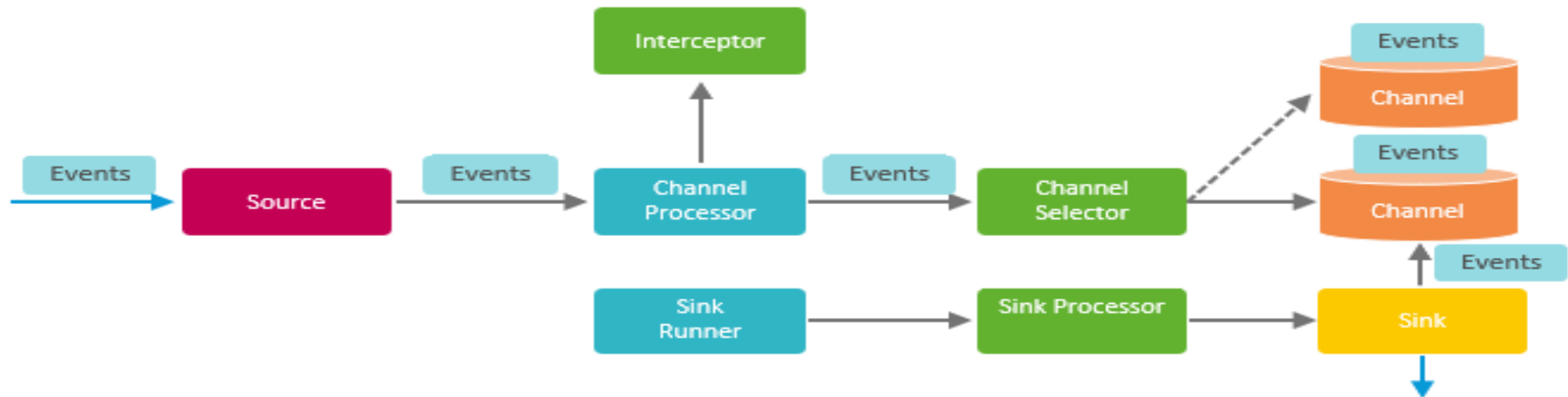
Flume Multi-Agent Consolidation



Chapter

07

Flume Architecture Diagram



Chapter

07

Basic Concepts — Source (1)

A source receives or generates events using a special mechanism and places them to one or more channels in batches.

There are two types of sources: driver-based and polling.

Driver-based sources: External systems proactively send data to Flume, driving Flume to receive data.

Polling sources: Flume periodically obtains data.

A source must be associated with at least one channel.

Basic Concepts — Source

Source Type	Description
Exec source	Executes a command or script and uses the output of the execution result as the data source.
Avro source	Provides an Avro-based server, which is bound to a port, and waits for the data sent from the Avro client.
Thrift source	Same as Avro, but the transmission protocol is Thrift.
HTTP source	Sends data via the POST request of HTTP.
Syslog source	Collects syslogs.
Kafka source	Obtains data from Kafka.

Basic Concepts — Channel

Channels are located between a source and sink. They function as queues and are used to cache temporary events. When a sink successfully sends events to the channel of the next hop or the final destination, the events are removed from the channel.

Channels are fully transactional, provide weak ordering guarantees, and can work with any number of sources and sinks.

Different channels offer different levels of persistence:

- Memory channel: volatile

- File channel: backed by write-ahead log (WAL) implementation

- JDBC channel: implemented based on embedded databases

Basic Concepts — Channel

- Memory channel: Messages are stored in the memory, providing high throughput but not ensuring reliability. This means data may be lost.
- File channel: Data is persisted; however, the configuration is complex. You need to configure the data directory (for each file channel) and checkpoint directory.
- JDBC channel: A built-in database makes events persistent with high reliability. This channel can replace the file channel that also supports data persistence.

Basic Concepts — sink

- A sink sends events to the next hop or the final destination, and then removes the events from the channel.
- A sink must work with a specific channel.

Sink Type	Description
HDFS sink	Writes data to HDFS.
Avro sink	Sends data to Flume of the next hop using the Avro protocol.
Thrift sink	Same as Avro, but the transmission protocol is Thrift.
File roll sink	Saves data to the local file system.
HBase sink	Writes data to HBase.
Kafka sink	Writes data to Kafka.

Flume: Massive Log Aggregation

- Overview and Architecture
- ▣ Key Features

Kafka: Distributed Messaging System

Chapter **07**

Content

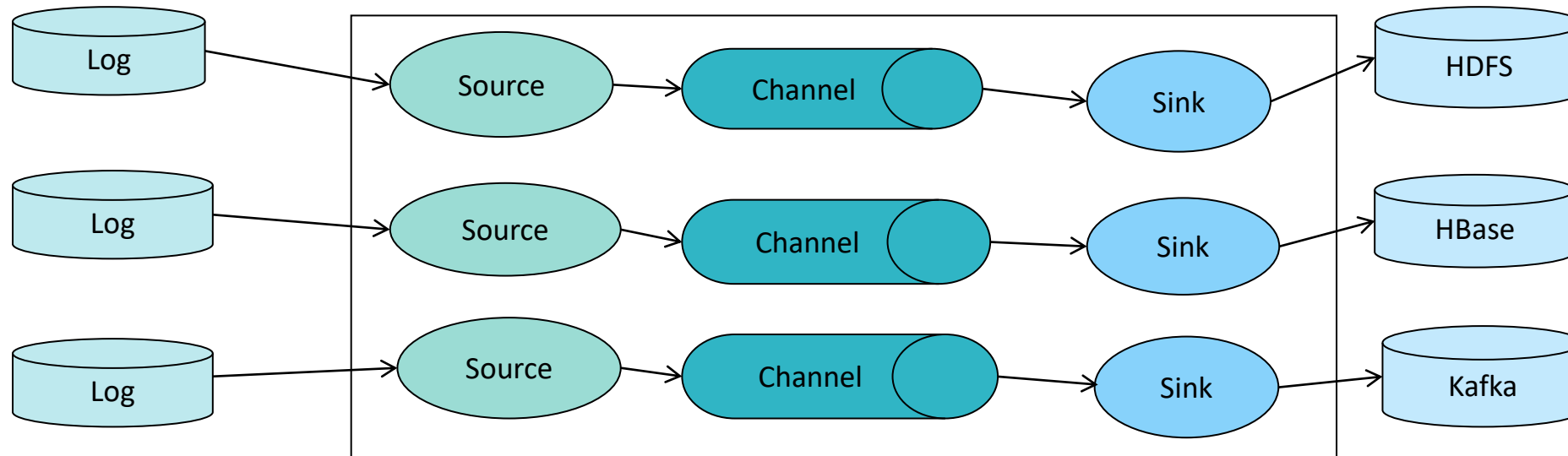


Chapter

07

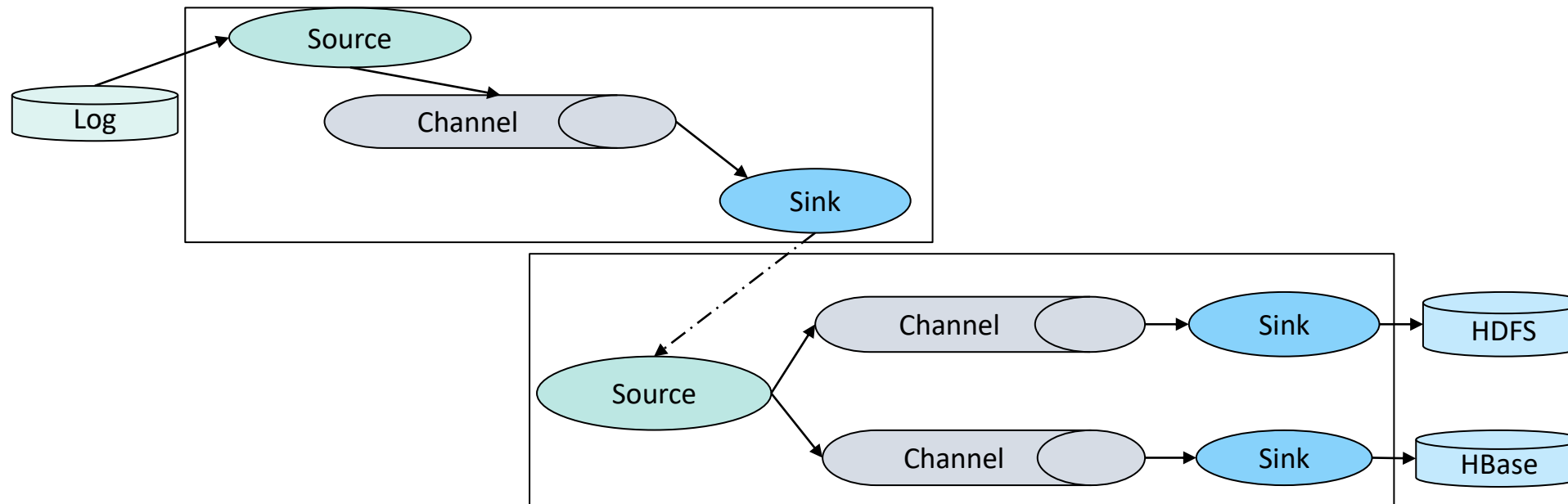
Log File Collection

- Flume collects log files in a cluster and archives them in HDFS, HBase, or Kafka used for data analysis and cleansing for upper-layer applications.



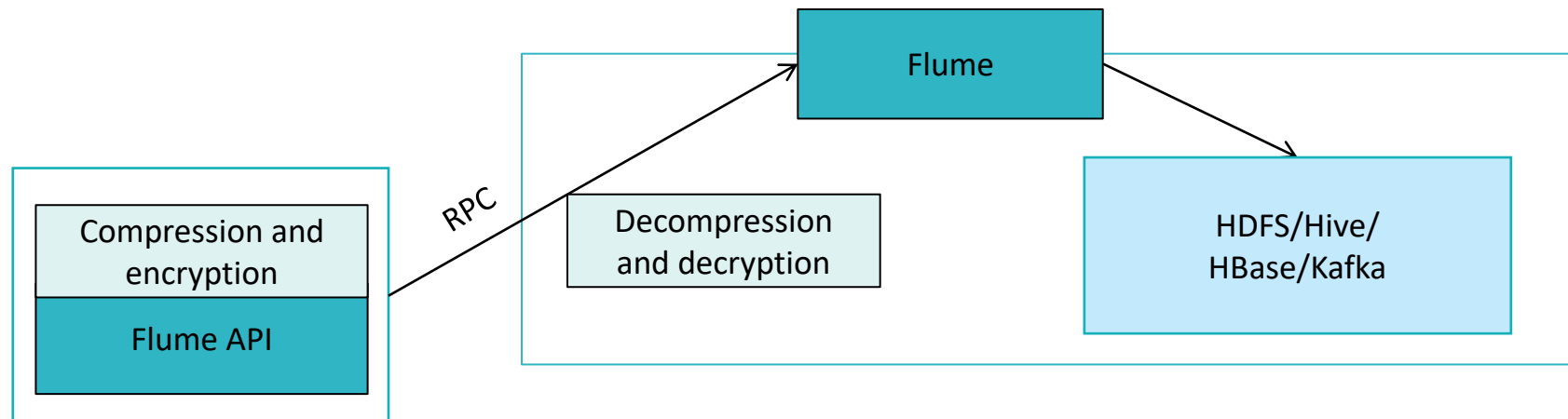
Multi-level Cascading and Multi-channel Replication

- Flume supports cascading of multiple Flume agents and data replication within the cascaded agents.



Cascading Message Compression and Encryption

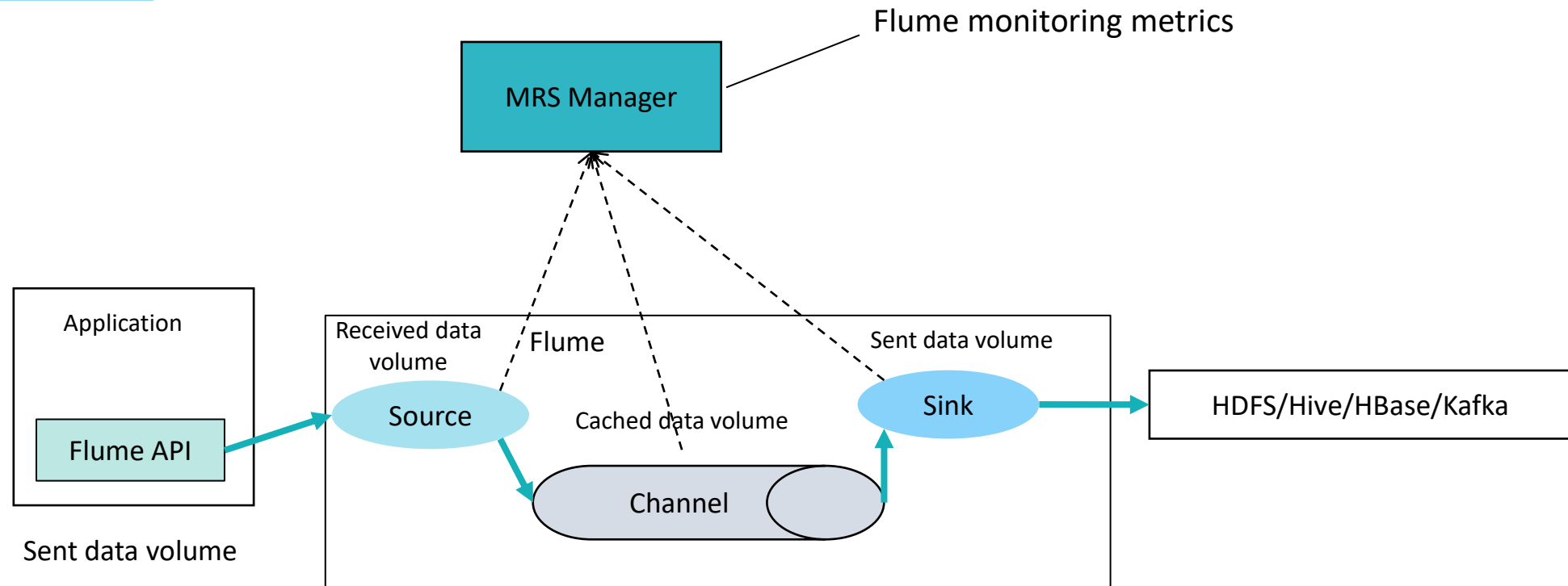
- Data transmission between cascaded Flume agents can be compressed and encrypted, improving data transmission efficiency and security.



Chapter

07

Data Monitoring

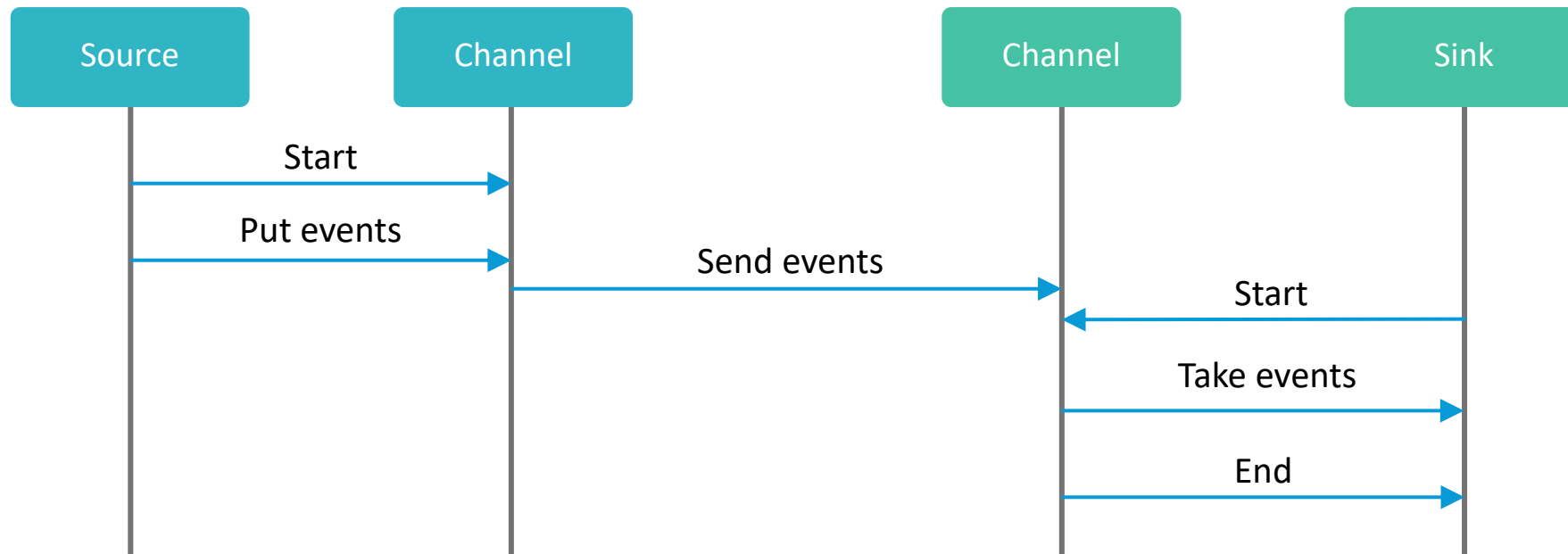


Chapter

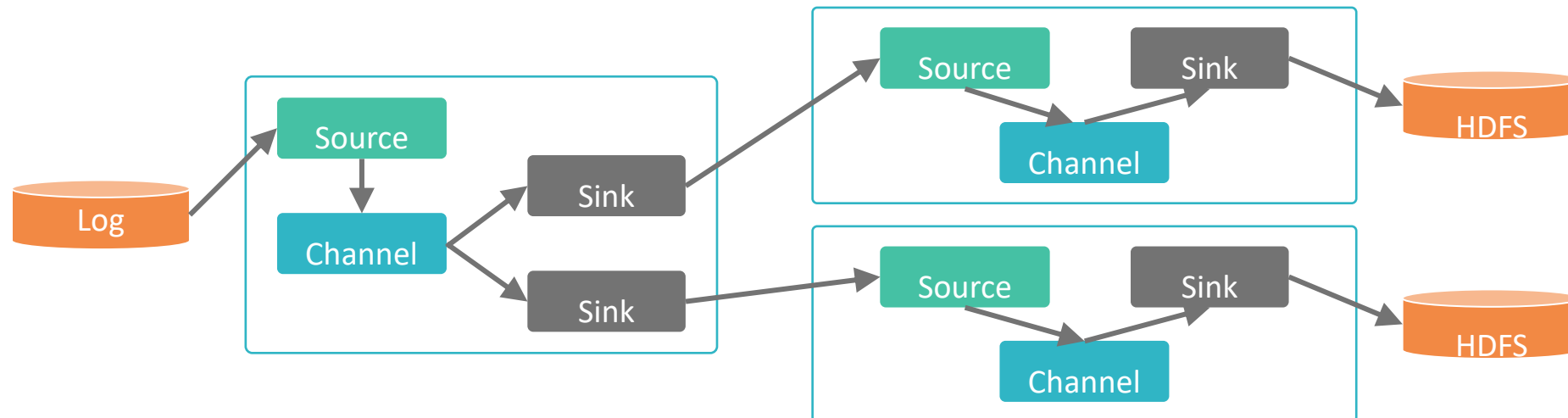
07

Transmission Reliability

- During data transmission, Flume uses the transaction management mechanism to ensure data completeness and keeps transmission reliable. In addition, if the data is cached in the file channel, data will not be lost when the process or agent is restarted.



- Data can be automatically switched to another channel for transmission when the next-hop Flume agent is faulty or data receiving is abnormal during Flume data transmission.

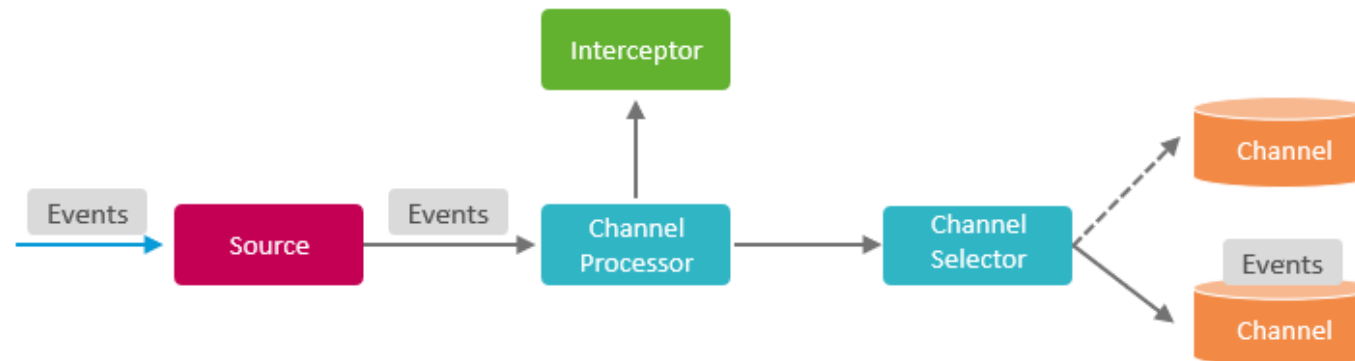


Chapter

07

Data Filtering During Transmission

- During data transmission, Flume roughly filters and cleans data and deletes unnecessary data. If the data to be filtered is complex, users need to develop filter plugins based on their data characteristics.



Flume: Massive Log Aggregation

Kafka: Distributed Messaging System

- Overview
- ▢ Architecture and Functions
- ▢ Data Management

Chapter

07

Content



Chapter

07

Kafka overview:



- Definition of Kafka: Kafka is a high-throughput, distributed, and publishing-subscription messaging system. A large messaging system can be established on low-cost servers with Kafka technology.

Chapter

07

- Kafka can persist messages in a time complexity ,and can maintain data access performance in constant time even in the face of terabytes of data.
- Kafka provides high throughput. Even on cheap commercial machines, a single-node system can transmit 100,000 messages per second.
- Kafka supports message partitioning and distributed consumption, and ensures that messages are transmitted in sequence in each partition.
- Kafka supports offline and real-time data processing.
- Kafka supports online horizontal scale-out.

Flume: Massive Log Aggregation

Kafka: Distributed Messaging System

- Overview
- ▣ Architecture and Functions
- ▣ Data Management

Chapter

07

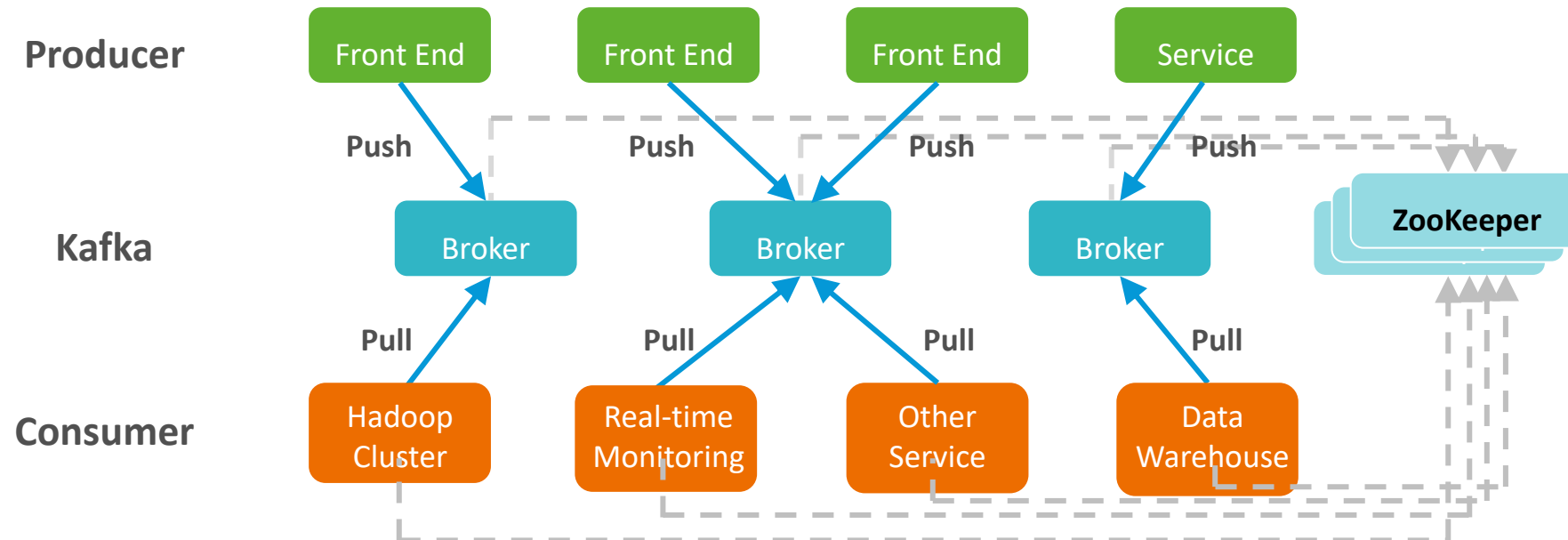
Content



Chapter

07

Kafka topology



Chapter

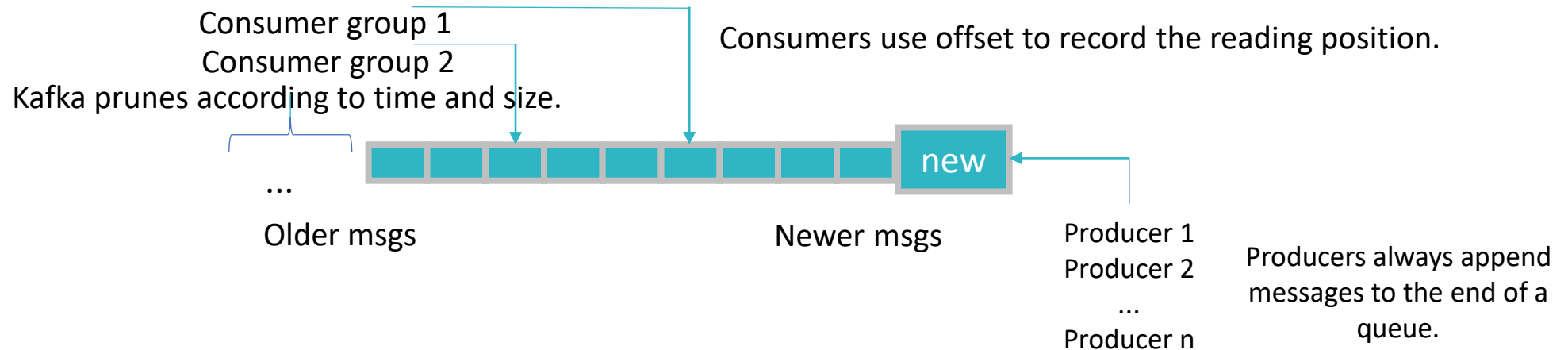
07

- 1 **Broker:** A Kafka cluster contains one or more service instances, which are called brokers.
- 2 **Topic:** Each message published to the Kafka cluster has a category, which is called a topic.
- 3 **Partition:** Kafka divides a topic into one or more partitions. Each partition physically corresponds to a directory for storing all messages of the partition.
- 4 **Producer:** sends messages to Kafka Broker.
- 5 **Consumer:** consumes messages and functions as a Kafka client to read messages from Kafka brokers.
- 6 **Consumer Group:** Each consumer belongs to a given consumer group. You can specify a group name for each consumer.

Chapter

07

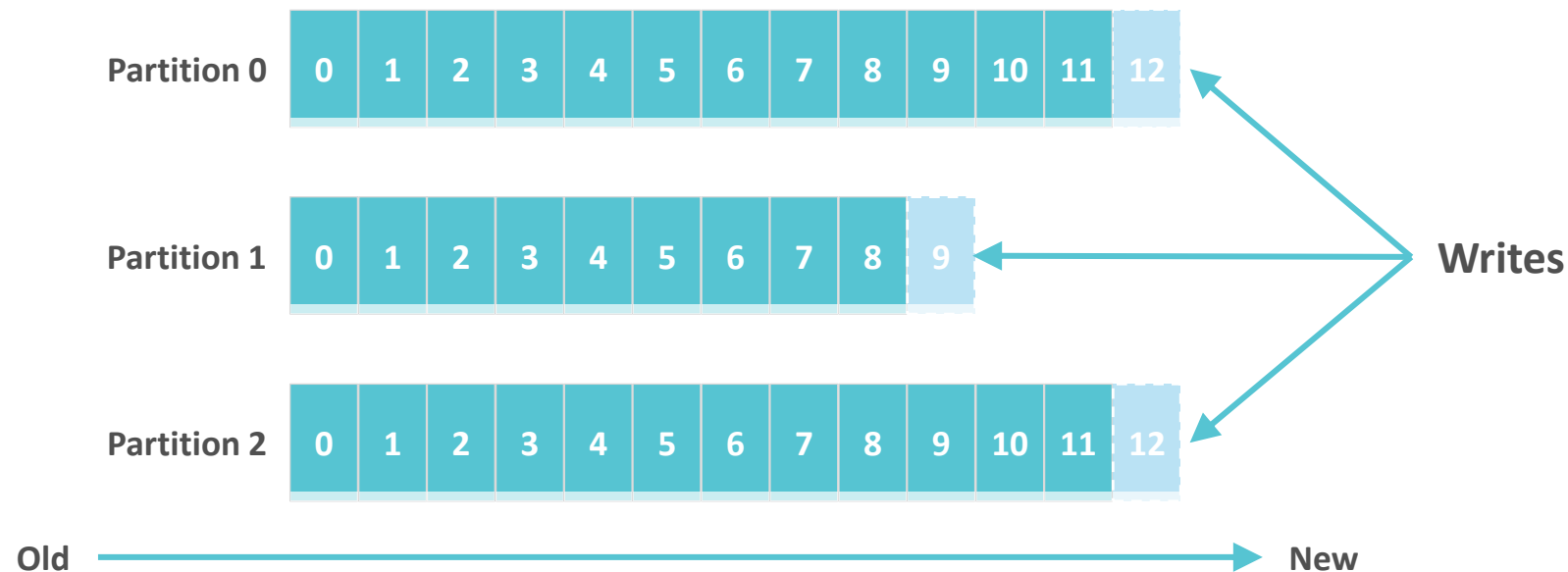
- Each message published to Kafka belongs to a category, which is called a topic. A topic can also be interpreted as a message queue. For example, weather can be regarded as a topic (queue) that stores daily temperature information.



Chapter

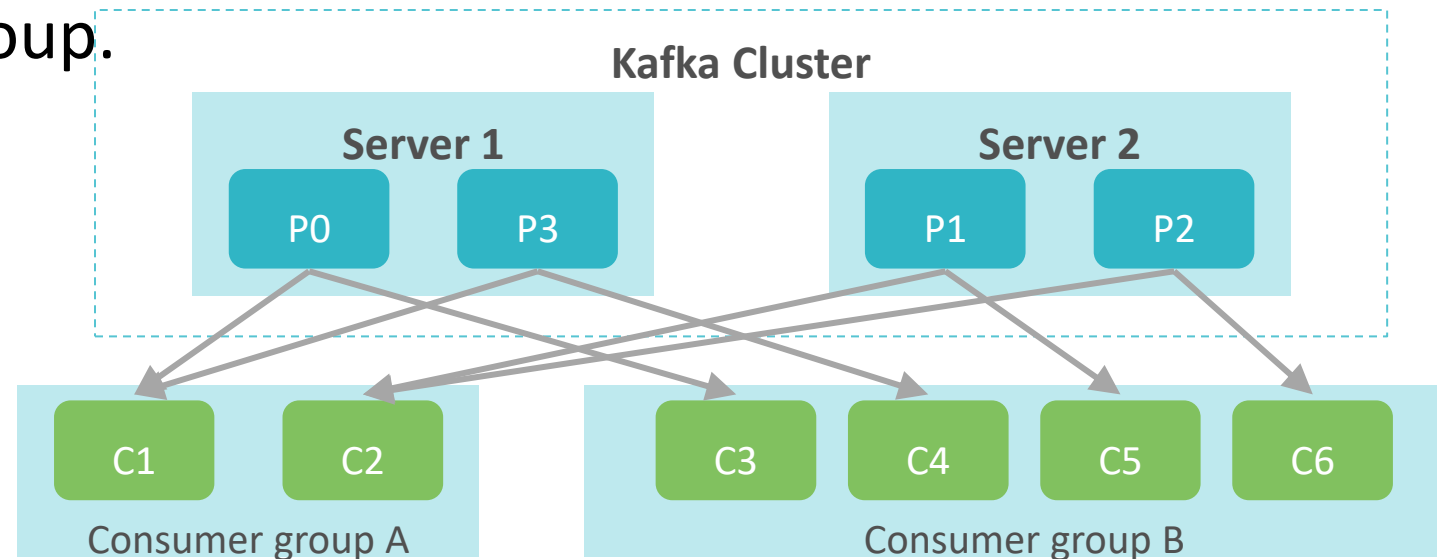
07

To improve the throughput of Kafka, each topic is physically divided into one or more partitions. Each partition is an ordered and immutable sequence of messages. Each partition physically corresponds to a directory for storing all messages and index files of the partition.



Consumer Groups

- Each consumer belongs to a consumer group. Each message can be consumed by multiple consumer groups but only one consumer in a consumer group. That is, data is shared between groups, but exclusive within a group.



Other Important Concepts

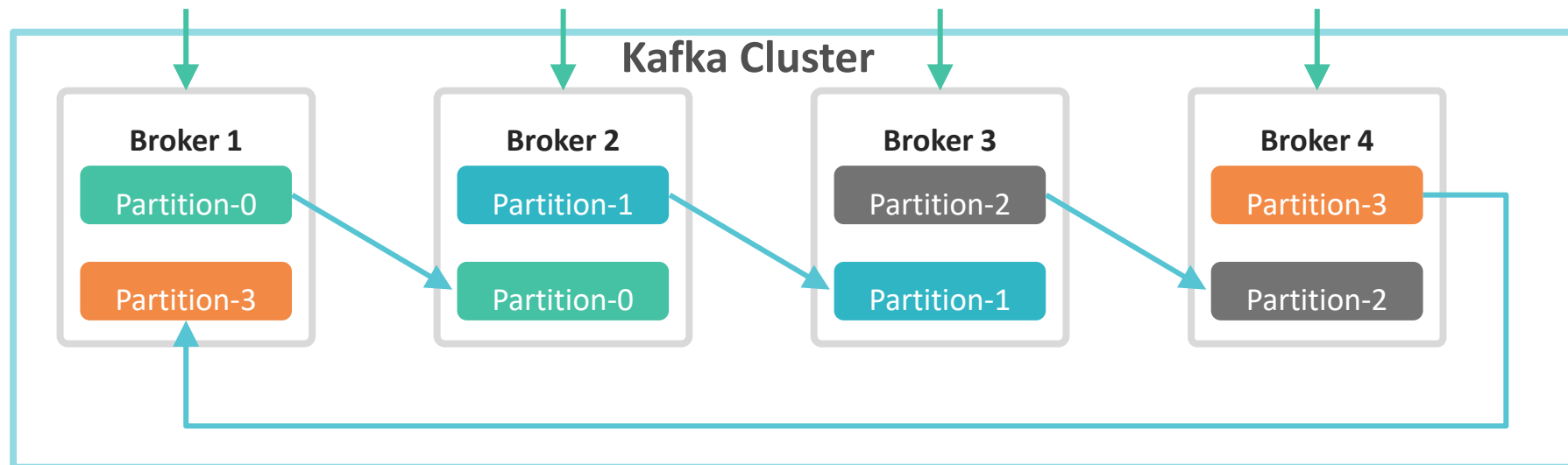
- **Replica:**
 - Refers to a replica of a partition, which guarantees high availability of partitions.
- **Leader:**
 - A role in a replica. Producers and consumers only interact with the leader.
- **Follower:**
 - A role in a replica, which replicates data from the leader.
- **Controller:**
 - A server in a Kafka cluster, which is used for leader election and failovers.

Chapter

07

Partition Replica

- Each partition has one or more replications, which are distributed on different brokers of the cluster to improve availability

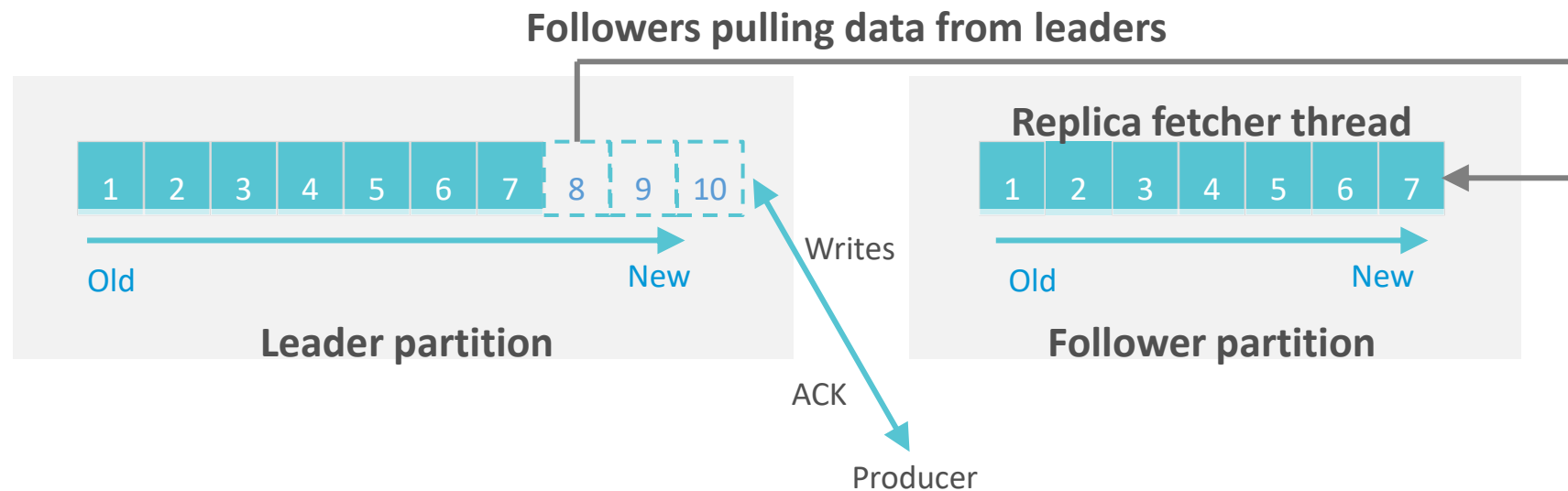


Chapter

07

Partition Replica

- The Kafka partition replica (follower) pulls stored messages below the high watermark from the leader to the local logs.



Chapter

07

In Kafka, each partition of a topic is sub-divided into segments, making it easier for periodical clearing or deletion of consumed files to free up space.

```
-rw----- 1 omm wheel 10485760 Jun 13 13:44 00000000000000000000.index  
-rw----- 1 omm wheel 1081187 Jun 13 13:45 00000000000000000000.log
```

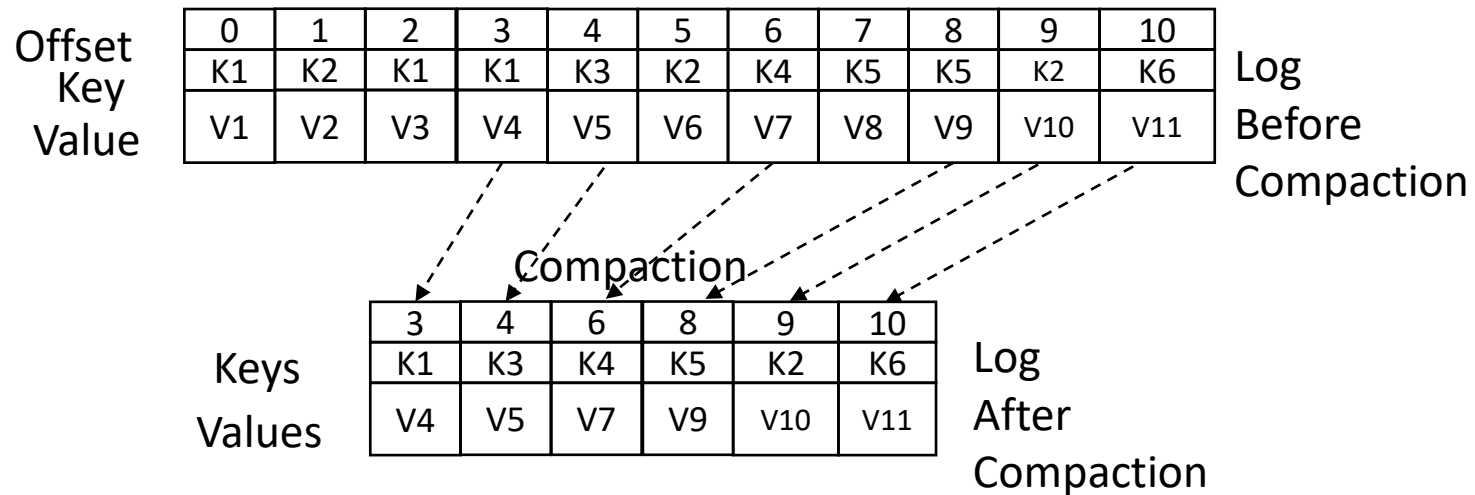
For traditional message queues, messages that have been consumed are deleted. However, the Kafka cluster retains all messages regardless of whether they have been consumed. Due to disk restrictions, it is impossible to permanently retain all data (actually unnecessary). Therefore, Kafka needs to process old data.

Chapter

07

Log cleanup policies: **delete** and **compact**

Threshold for deleting logs: retention time limit and size of all logs in a partition



Flume: Massive Log Aggregation

Kafka: Distributed Messaging System

- Overview
- ▢ Architecture and Functions
- ▢ Data Management

Chapter

07

Content



Write data by producer



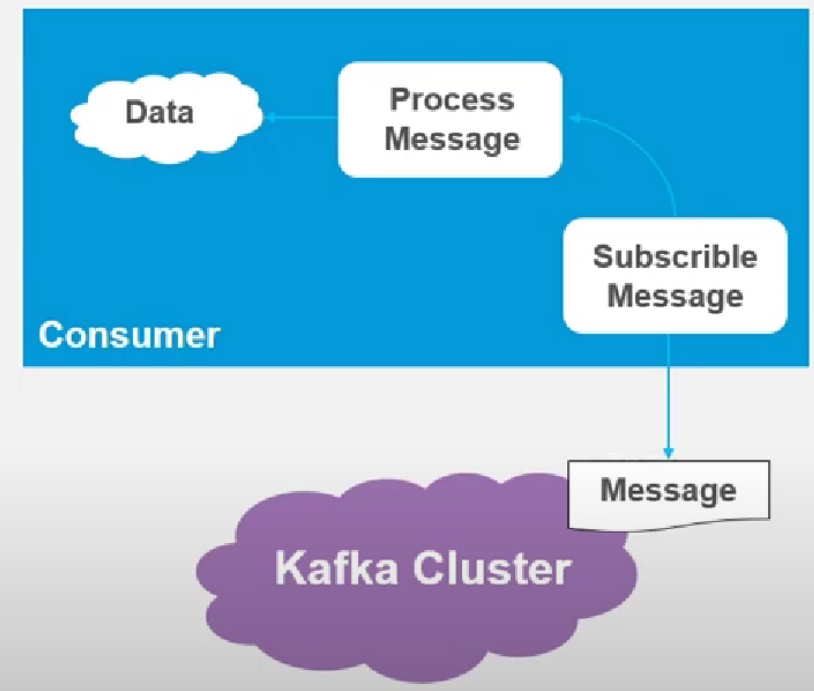
Chapter

07

**Read data by the
consumer**

Overall process:

- A consumer connects to the leader broker where the specified topic partition is located and pulls messages from Kafka logs.



Summary:

Flume is a streaming log collection tool that roughly processes data and writes the data to data receivers. It collects data from local files , real-time logs , Thrift, Avro, Syslog, Kafka, and other data sources.

Kafka, is a distributed streaming platform designed for high-throughput, fault-tolerant, and real-time data streaming. It acts as a distributed publish-subscribe messaging system.



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY
BY EVERYONE'S SMART UNIVERSITY



Thank You

