

Kafka技术原理

www.huawei.com





目标

- 学完本课程后，您将能够：
 - 掌握消息系统的基本概念和**Kafka**的应用场景
 - 掌握**Kafka**系统架构
 - 掌握**Kafka**关键流程
 - 掌握**Kafka**在**ZooKeeper**上的目录结构



目录

1. **Kafka**简介
2. **Kafka**架构与功能
3. **Kafka**关键流程
4. **Kafka**在ZooKeeper上的目录结构
5. **Kafka**高级专题

Kafka简介

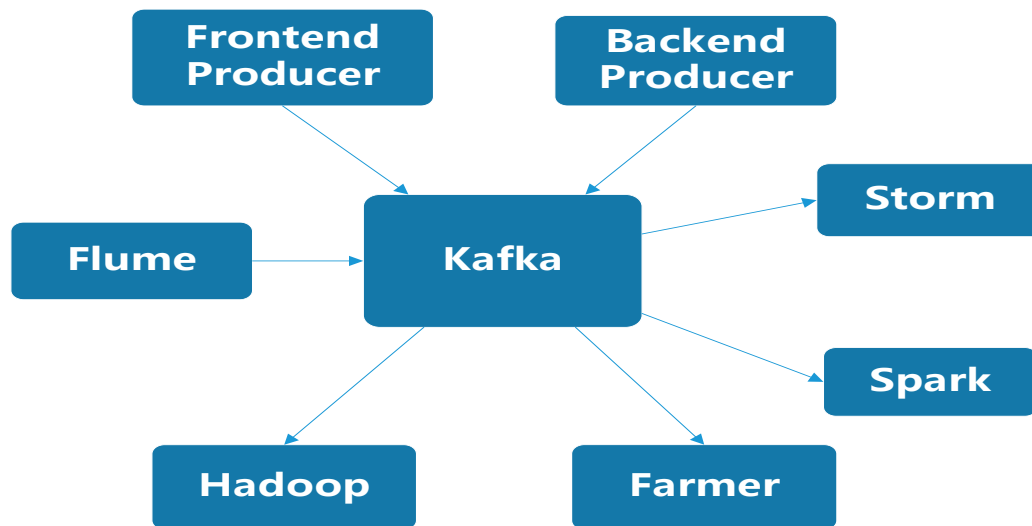
- **Kafka**定义

- **Kafka** -是一个高吞吐、分布式、基于发布订阅的消息系统，利用**Kafka**技术可在廉价**PC Server**上搭建起大规模消息系统。

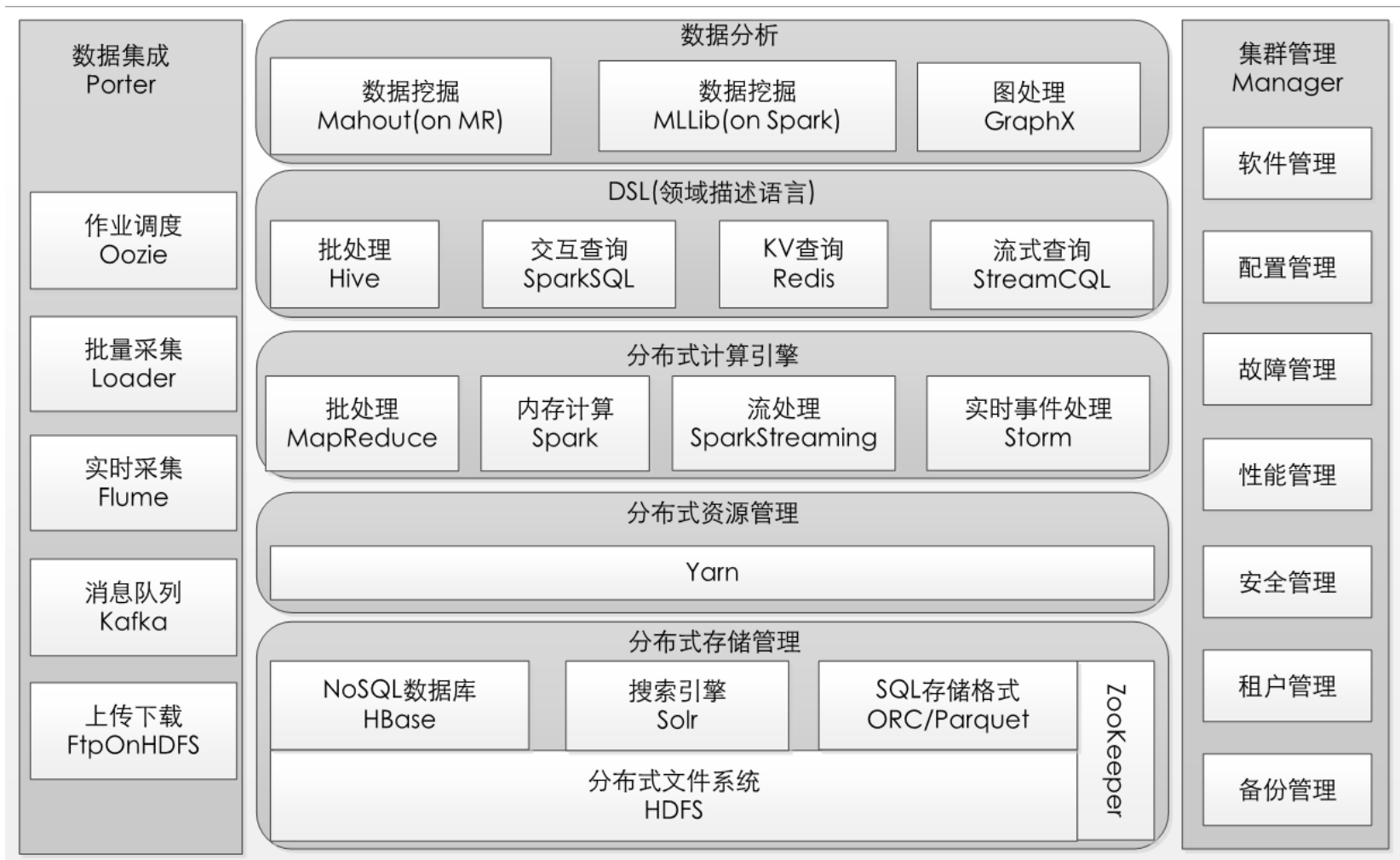
Kafka简介

- **Kafka**应用场景简介

- **Kafka**和其他组件比较，具有消息持久化、高吞吐、分布式、多客户端支持、实时等特性，适用于离线和在线的消息消费，如常规的消息收集、网站活性跟踪、聚合统计系统运营数据（监控数据）、日志收集等大量数据的互联网服务的数据收集场景。



Kafka简介






目录

1. Kafka简介
2. Kafka架构与功能
3. Kafka关键流程
4. Kafka在ZooKeeper上的目录结构
5. Kafka高级专题

Kafka基本概念



Broker: Kafka集群包含一个或多个服务实例，这些服务实例被称为**Broker**。

Topic: 每条发布到Kafka集群的消息都有一个类别，这个类别被称为**Topic**。

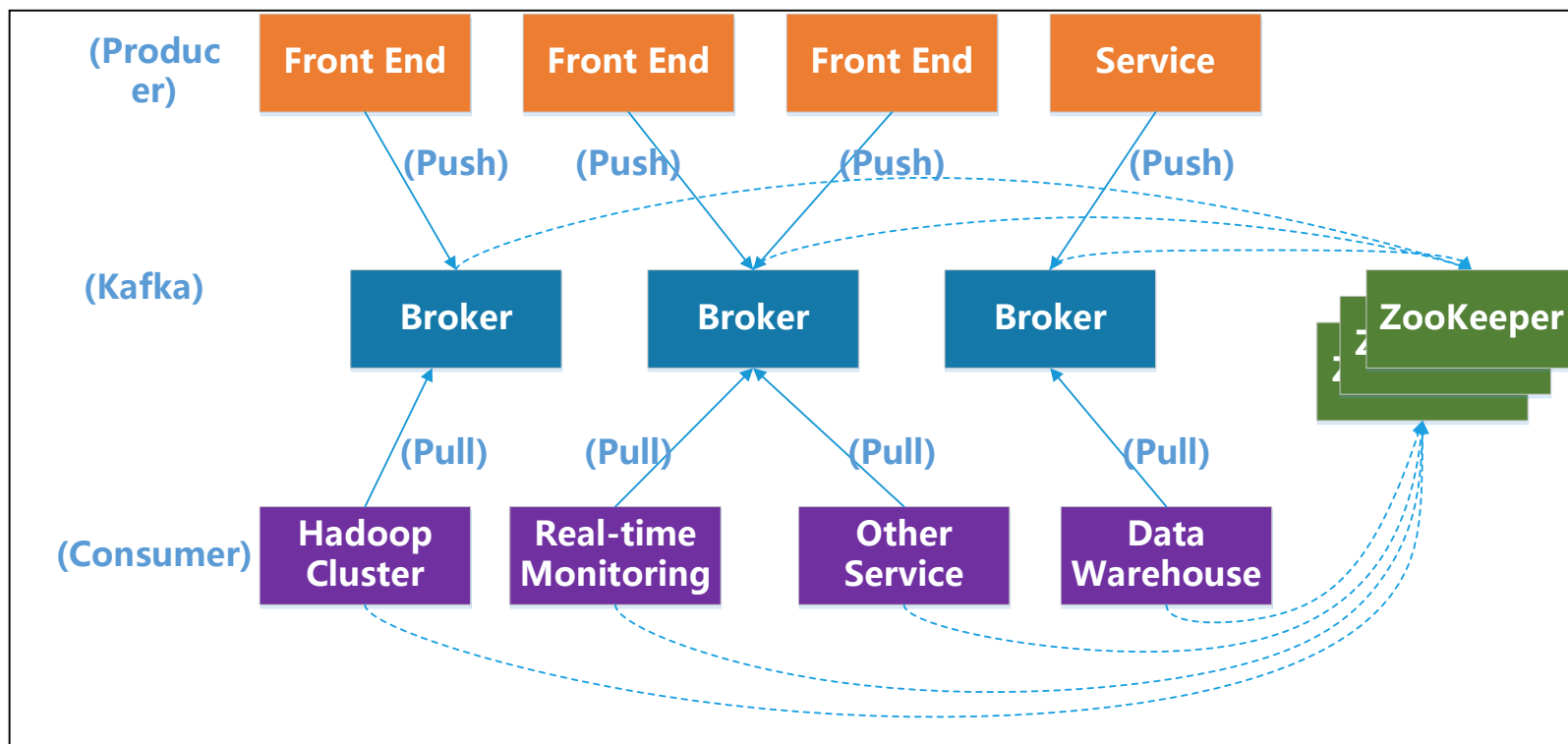
Partition: Kafka将Topic分成一个或者多个**Partition**，每个**Partition**在物理上对应一个文件夹，该文件夹下存储这个**Partition**的所有消息。

Producer: 负责发布消息到Kafka Broker。

Consumer: 消息消费者，从Kafka Broker读取消息的客户端。

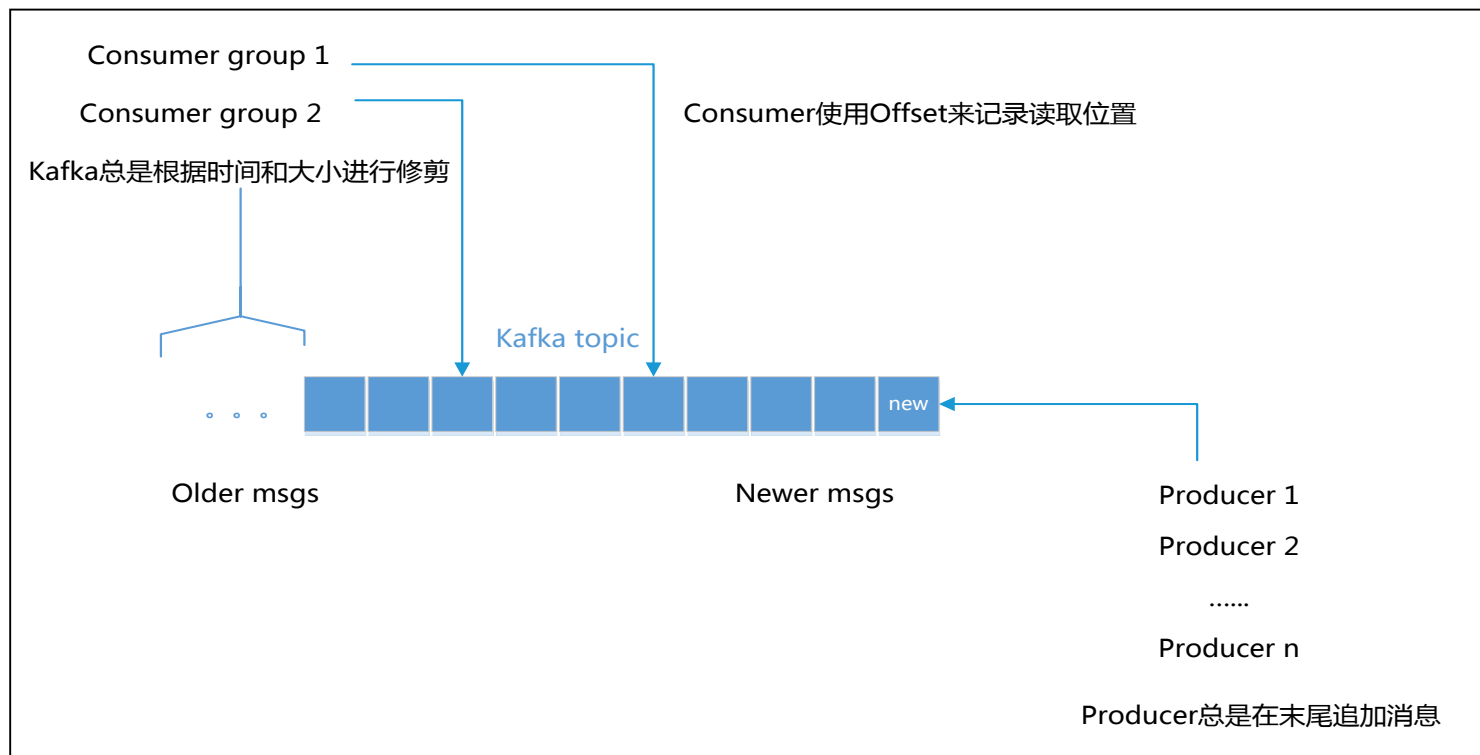
Consumer Group: 每个Consumer属于一个特定的Consumer Group（可为每个Consumer指定group name）。

Kafka拓扑结构图



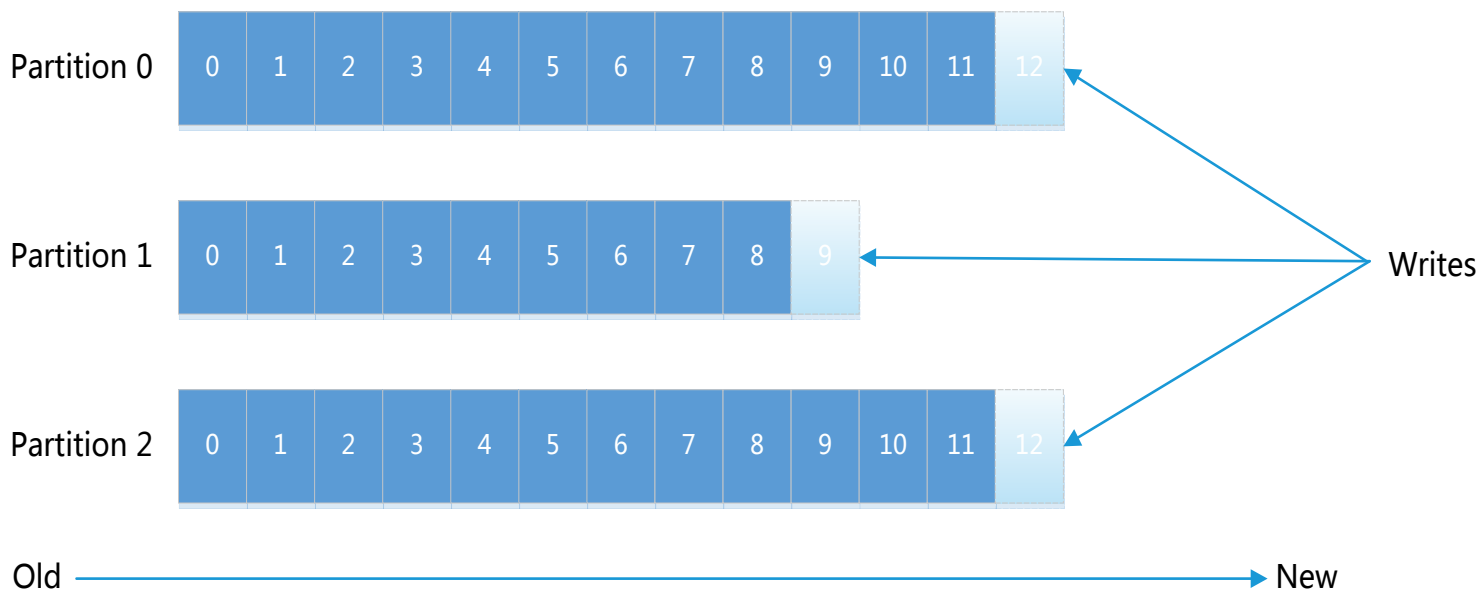
Kafka Topics

- 每条发布到**Kafka**的消息都有一个类别，这个类别被称为**Topic**，也可以理解为一个存储消息的队列。例如：天气作为一个**Topic**，每天的温度消息就可以存储在“天气”这个队列里。



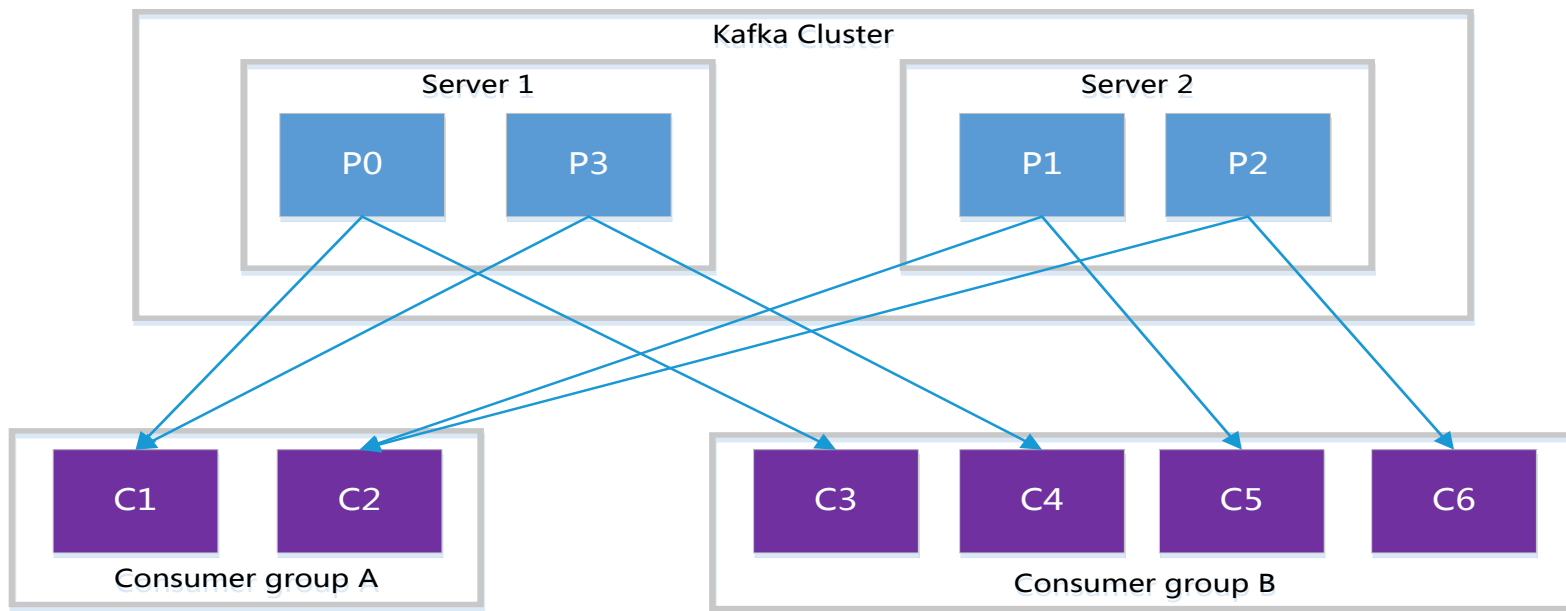
Kafka Partition

- 每个**Topic** 都有一个或者多个**Partitions**构成。每个**Partition**都是有顺序且不可变的消息队列。引入**Partition**机制，保证了**Kafka**的高吞吐能力。



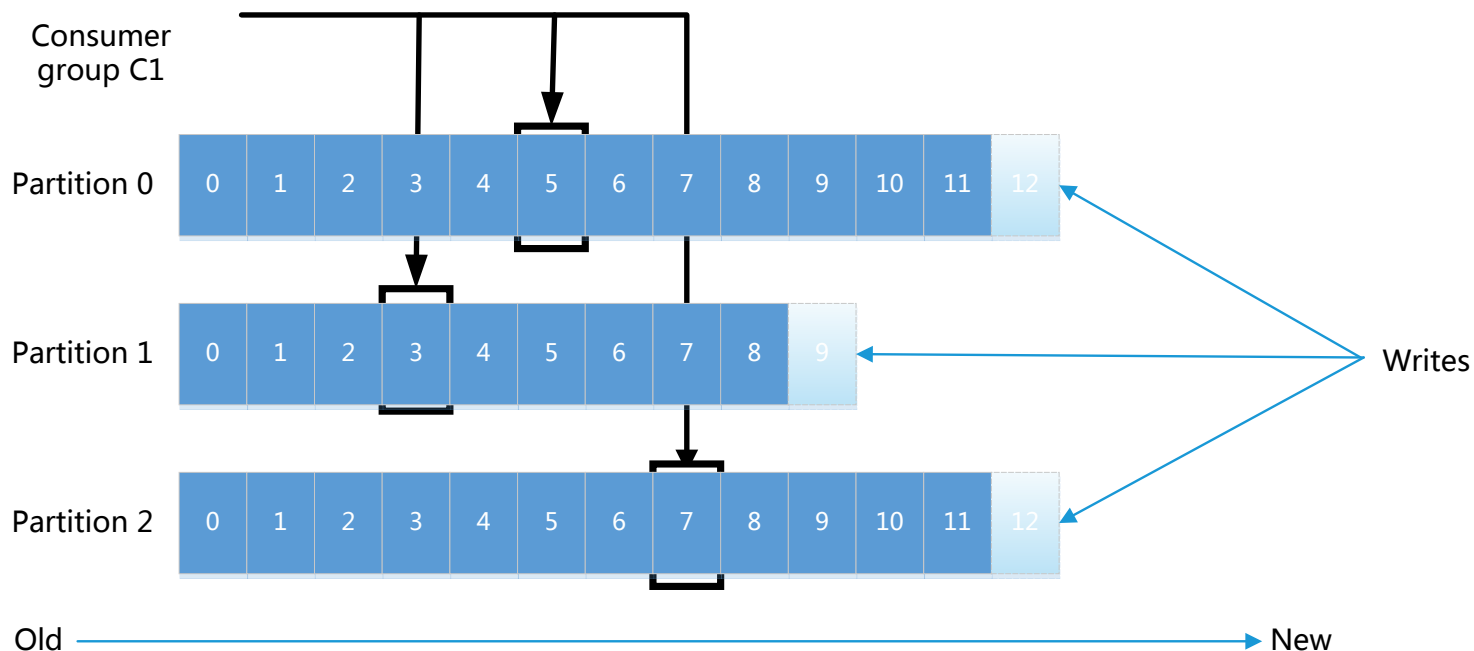
Kafka Partition

- **Topic**的**Partition**数量可以在创建时配置。
- **Partition**数量决定了每个**Consumer group**中并发消费者的最大数量。
- **Consumer group A**有两个消费者来读取**4**个**Partition**中数据；**Consumer group B**有四个消费者来读取**4**个**partition**中数据。



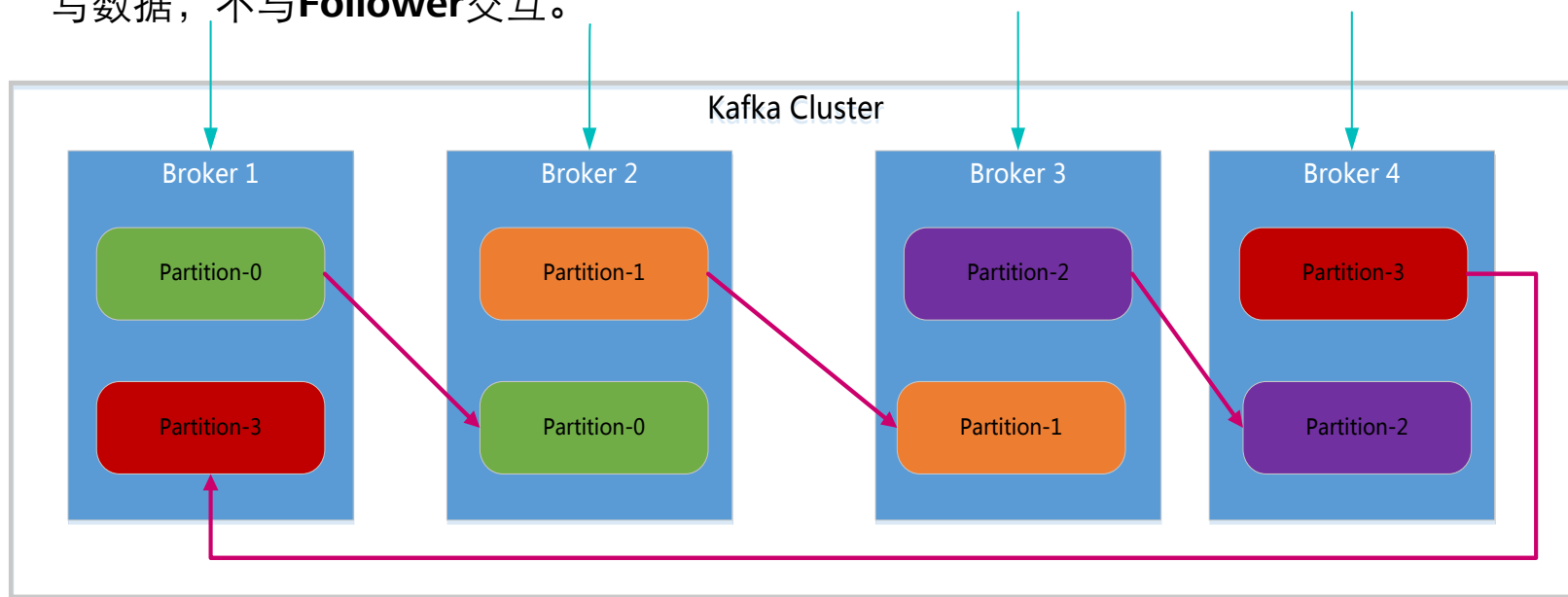
Kafka Partition offset

- 任何发布到此**Partition**的消息都会被直接追加到**log**文件的尾部。
- 每条消息在文件中的位置称为**offset**（偏移量），**offset**是一个**long**型数字，它唯一标记一条消息。消费者通过（**offset**、**partition**、**topic**）跟踪记录。

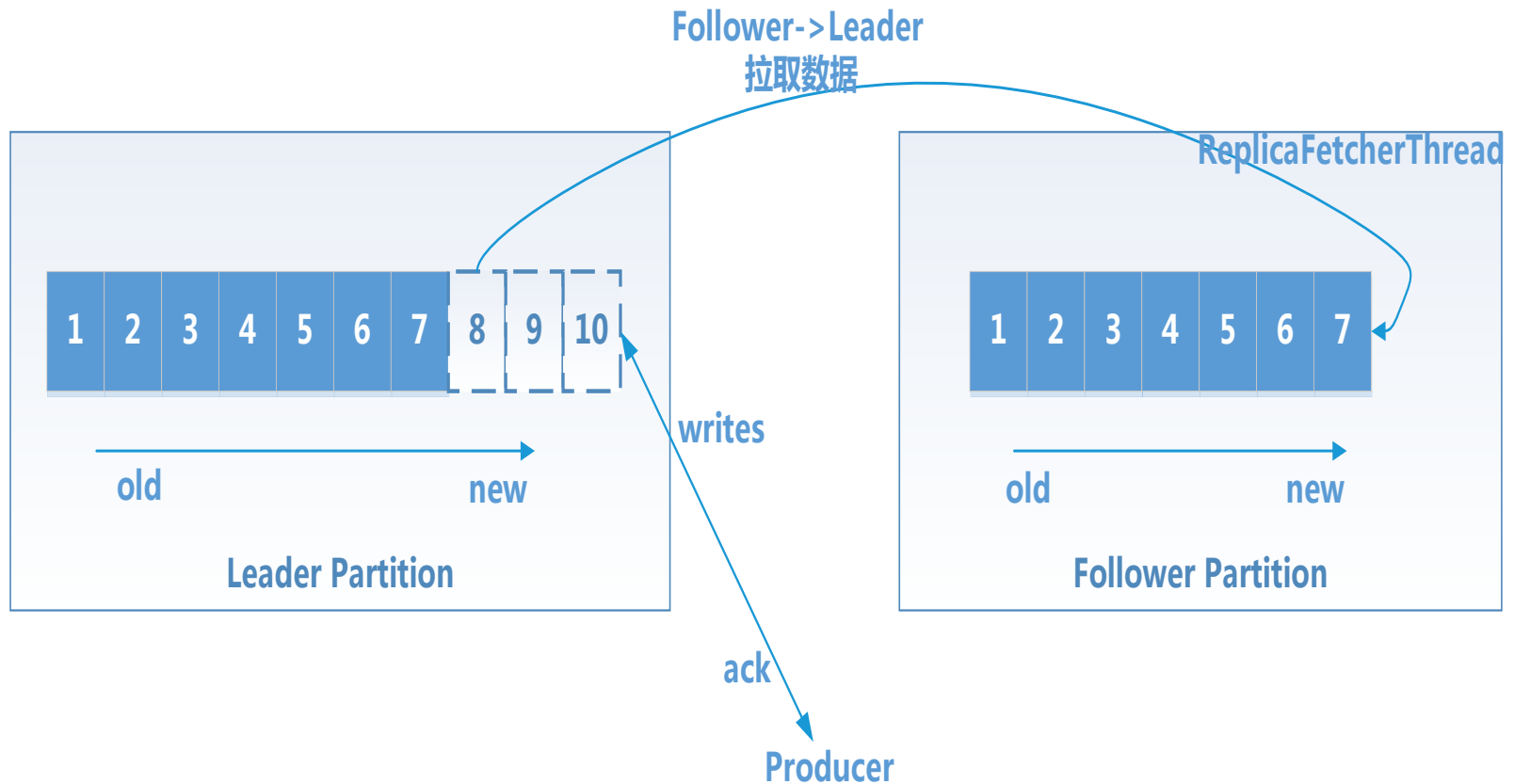


Kafka Partition Replicas

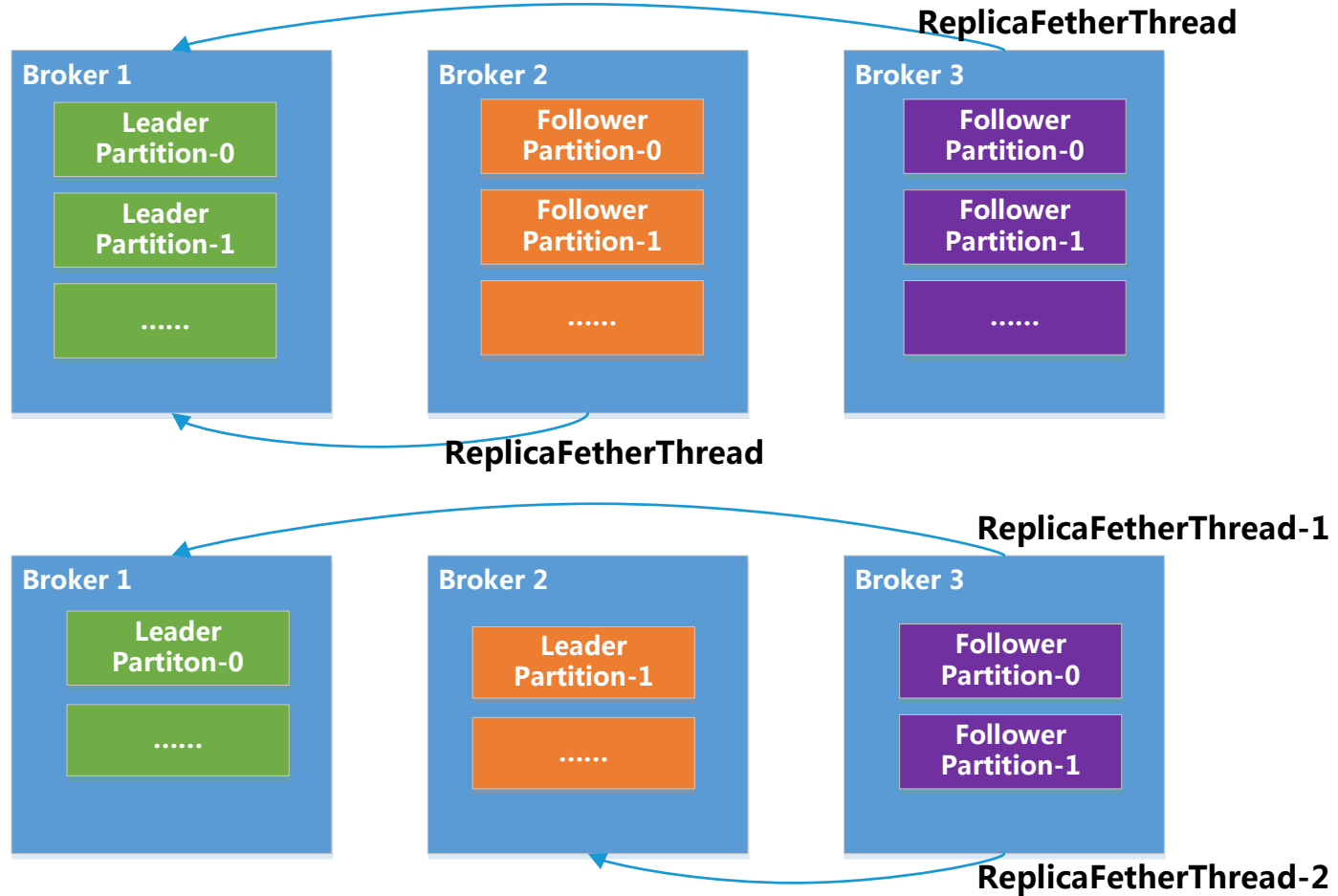
- 副本以分区为单位。每个分区都有各自的主副本和从副本。
- 主副本叫做**Leader**，从副本叫做**Follower**，处于同步状态的副本叫做**In-Sync Replicas (ISR)**。
- **Follower**通过拉取的方式从**Leader**中同步数据。消费者和生产者都是从**Leader**中读写数据，不与**Follower**交互。



Kafka Partition Replicas



Kafka Partition Replicas



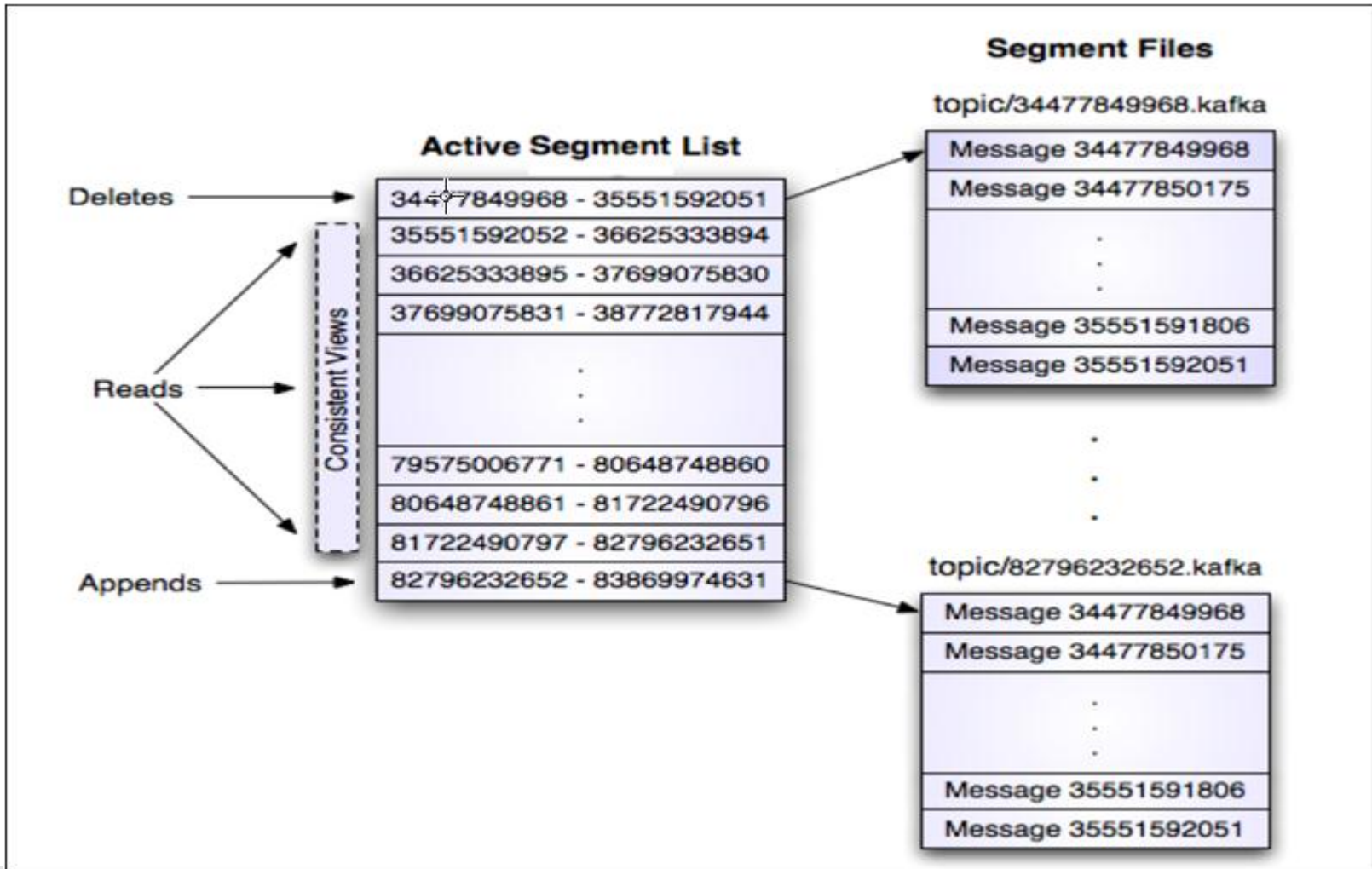
Kafka Logs

- 为了使得**Kafka**的吞吐率可以线性提高，物理上把**Topic**分成一个或多个**Partition**，每个**Partition**在物理上对应一个文件夹，该文件夹下存储这个**Partition**的所有消息和索引文件。**Kafka**把**Topic**中一个**Partition**大文件分成多个小文件段，通过多个小文件段，就容易定期清除或删除已经消费完文件，减少磁盘占用。

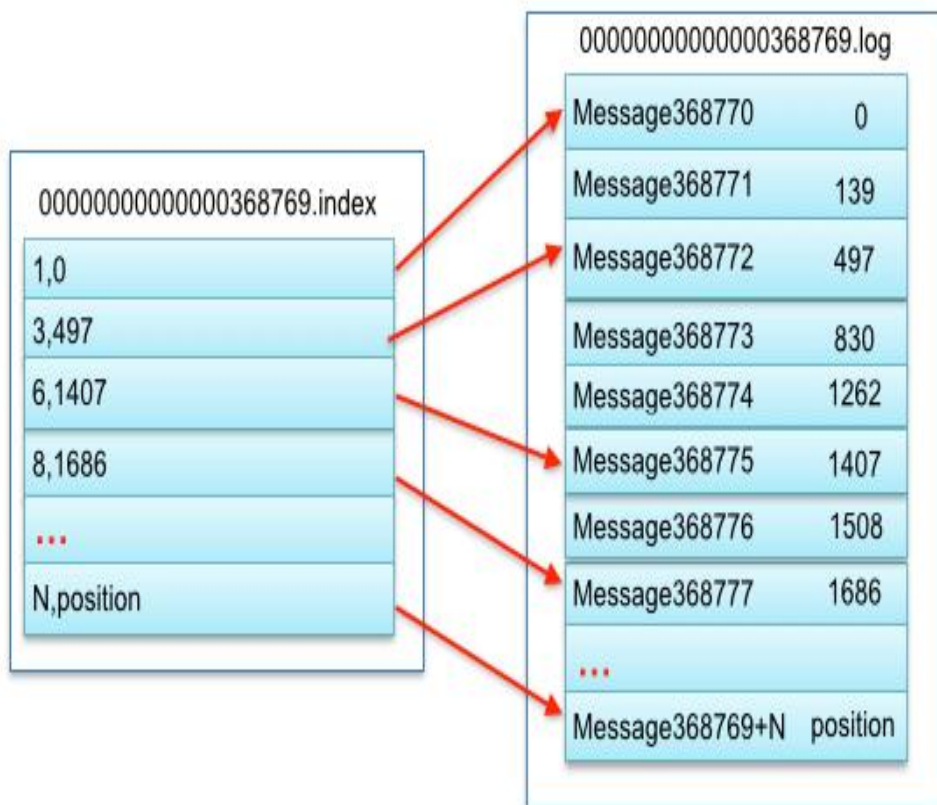
```
-rw----- 1 omm wheel    0 Jun 10 11:58 .lock
drwx----- 2 omm wheel 4096 Jun 12 14:59 example-metric1-0
-rw----- 1 omm wheel  102 Jun 13 17:57 recovery-point-offset-checkpoint
-rw----- 1 omm wheel  107 Jun 13 17:58 replication-offset-checkpoint
drwx----- 2 omm wheel 4096 Jun 12 10:52 test-0
drwx----- 2 omm wheel 4096 Jun 12 10:52 test-1
drwx----- 2 omm wheel 4096 Jun 12 11:08 test2-0
drwx----- 2 omm wheel 4096 Jun 12 11:08 test2-2
drwx----- 2 omm wheel 4096 Jun 12 14:59 test3-0
drwx----- 2 omm wheel 4096 Jun 12 14:59 test3-1
drwx----- 2 omm wheel 4096 Jun 12 14:59 test4-0
drwx----- 2 omm wheel 4096 Jun 12 14:59 test4-1
```

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

Kafka Logs



Kafka Logs



- 通过索引信息可以快速定位 **message**。
- 通过将**index**元数据全部映射到**memory**，可以避免 **segment file**的**index**数据IO 磁盘操作。
- 通过索引文件稀疏存储，可以大幅降低**index**文件元数据占用空间大小。

Kafka Message

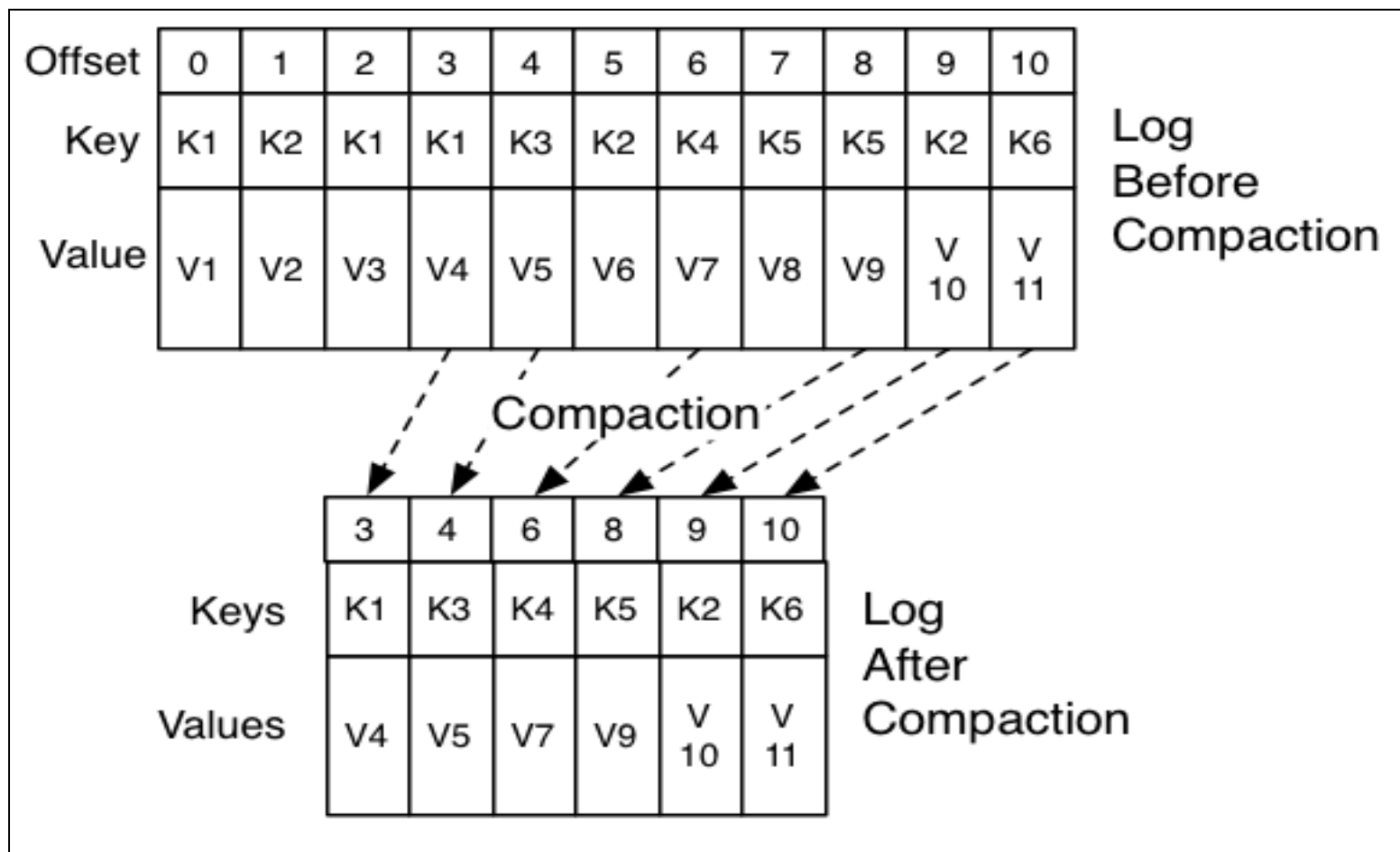
Message结构	关键字	解释说明
8 byte offset	8 byte offset	在 partition （分区）内的每条消息都有一个有序的 id 号，这个 id 号被称为偏移（ offset ），它可以唯一确定每条消息在 partition （分区）内的位置，即 offset 表示 partition 的第几个 message
4 byte message size	4 byte message size	Message 大小
4 byte CRC32	4 byte CRC32	用 CRC32 校验 message
1 byte "magic"	1 byte "magic"	表示本次发布 Kafka 服务程序协议版本号
1 byte "attributes"	1 byte "attributes"	表示为独立版本、或标识压缩类型、或编码类型
4 byte key length	4 byte key length	表示 key 的长度，当 key 为-1时， K byte key 字段不填
K byte key	K byte key	可选
4 byte payload length	Value bytes payload	表示实际消息数据
value bytes payload		

Kafka Log Cleanup

- 日志的清理方式有两种：**delete** 和 **compact**。
- 删除的阈值有两种：过期的时间和分区内总日志大小。

配置参数	默认值	参数解释	取值范围
log.cleanup.policy	delete	当日志过期时（超过了要保存的时间），采用的清除策略，可以取值为删除或者压缩。	delete或compact
log.retention.hours	168	日志数据文件保留的最长时间。单位：小时。	1 ~ 2147483647
log.retention.bytes	-1	指定每个Partition上的日志数据所能达到的最大字节。默认情况下无限制。单位：字节。	-1 ~ 9223372036854775807

Kafka Log Cleanup



Kafka数据可靠性

- **Kafka**所有消息都会被持久化到硬盘中，同时**Kafka**通过对**Topic Partition**设置**Replication**来保障数据可靠。
- 那么，在消息传输过程中有没有可靠性保证呢？

Message Delivery Semantics

消息传输保障通常有以下三种：

- 最多一次 (**At Most Once**)
 - 消息可能丢失。
 - 消息不会重复发送和处理。
- 最少一次 (**At Least Once**)
 - 消息不会丢失。
 - 消息可能会重复发送和处理。
- 仅有一次 (**Exactly Once**)
 - 消息不会丢失。
 - 消息仅被处理一次。

Kafka消息传输

- **Kafka**消息传输保障机制，通过配置不同的消息发送模式来保障消息传输，进而满足不同的可靠性要求应用场景。

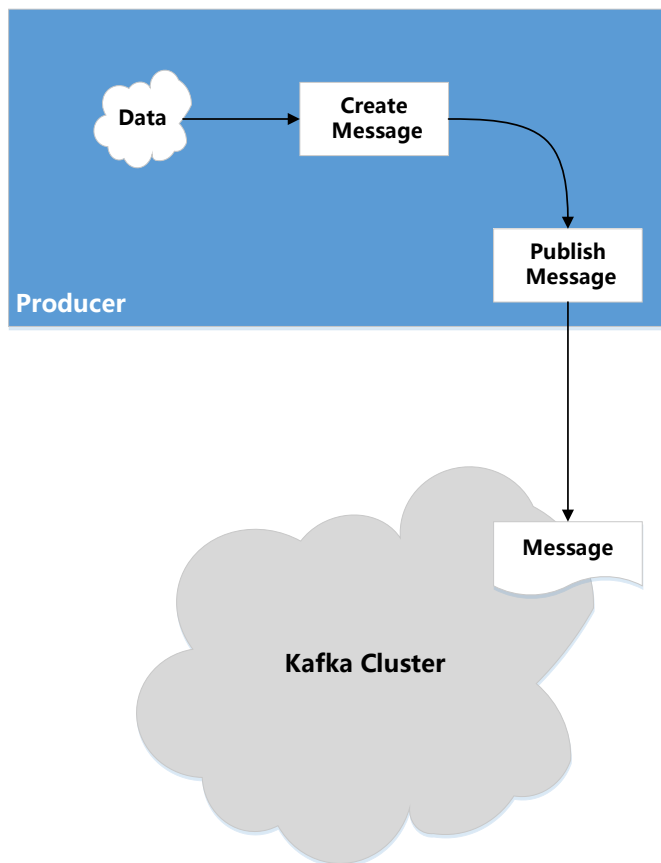
	同步发送不带确认	同步发送带确认	异步发送不带确认	异步发送带确认不重试	异步发送带确认有重试
无副本	最多一次	至少一次	最多一次	至少一次	至少一次
同步复制 (leader fellow)	最多一次	至少一次	最多一次	至少一次	至少一次
异步复制 (leader)	最多一次	消息可能丢失或重复	最多一次	消息可能丢失或重复	消息可能丢失或重复



目录

1. Kafka简介
2. Kafka架构与功能
3. Kafka关键流程
 - ▣ Kafka写流程
 - ▣ Kafka读流程
4. Kafka在ZooKeeper上的目录结构
5. Kafka高级专题

Producer写数据



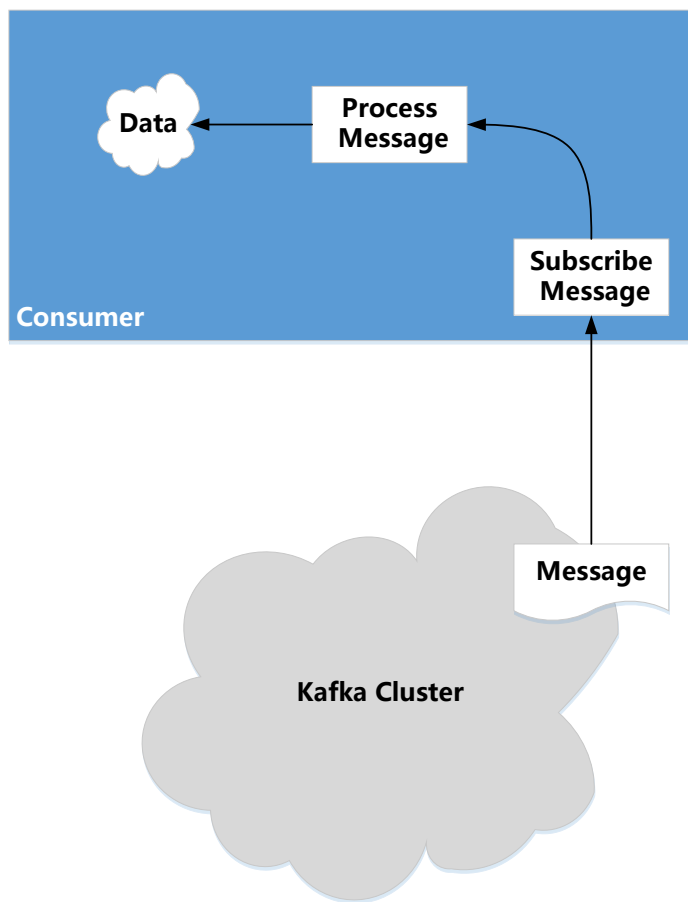
- 总体流程：
Producer连接任意存活的**Broker**，请求制定**Topic**、**Partition**的**Leader**元数据信息，然后直接与对应的**Broker**直接连接，发布数据。
- 开放分区接口：
用户可以制定分区函数，使得消息可以根据**Key**，发送到特定**Partition**。



目录

1. Kafka简介
2. Kafka架构与功能
3. Kafka关键流程
 - ▣ Kafka写流程
 - ▣ Kafka读流程
4. Kafka在ZooKeeper上的目录结构
5. Kafka高级专题

Consumer读数据



- 总体流程：
Consumer连接指定**TopicPartition**所在的**LeaderBroker**，用主动获取方式从**Kafka**中获取消息。



目录

1. Kafka简介
2. Kafka架构与功能
3. Kafka关键流程
4. Kafka在ZooKeeper上的目录结构
5. Kafka高级专题

ZooKeeper Shell

- 通过zkCli来连接正在运行的ZooKeeper Shell客户端，可以通过ls 和 get命令来获取Kafka相关信息

用法:

```
# bin/zkCli.sh -server zk_host:port/chroot
```

```
[zk: 192.168.0.90:24002/kafka(CONNECTED) 1] ls /  
[admin, brokers, config, consumers, controller, controller_epoch]
```

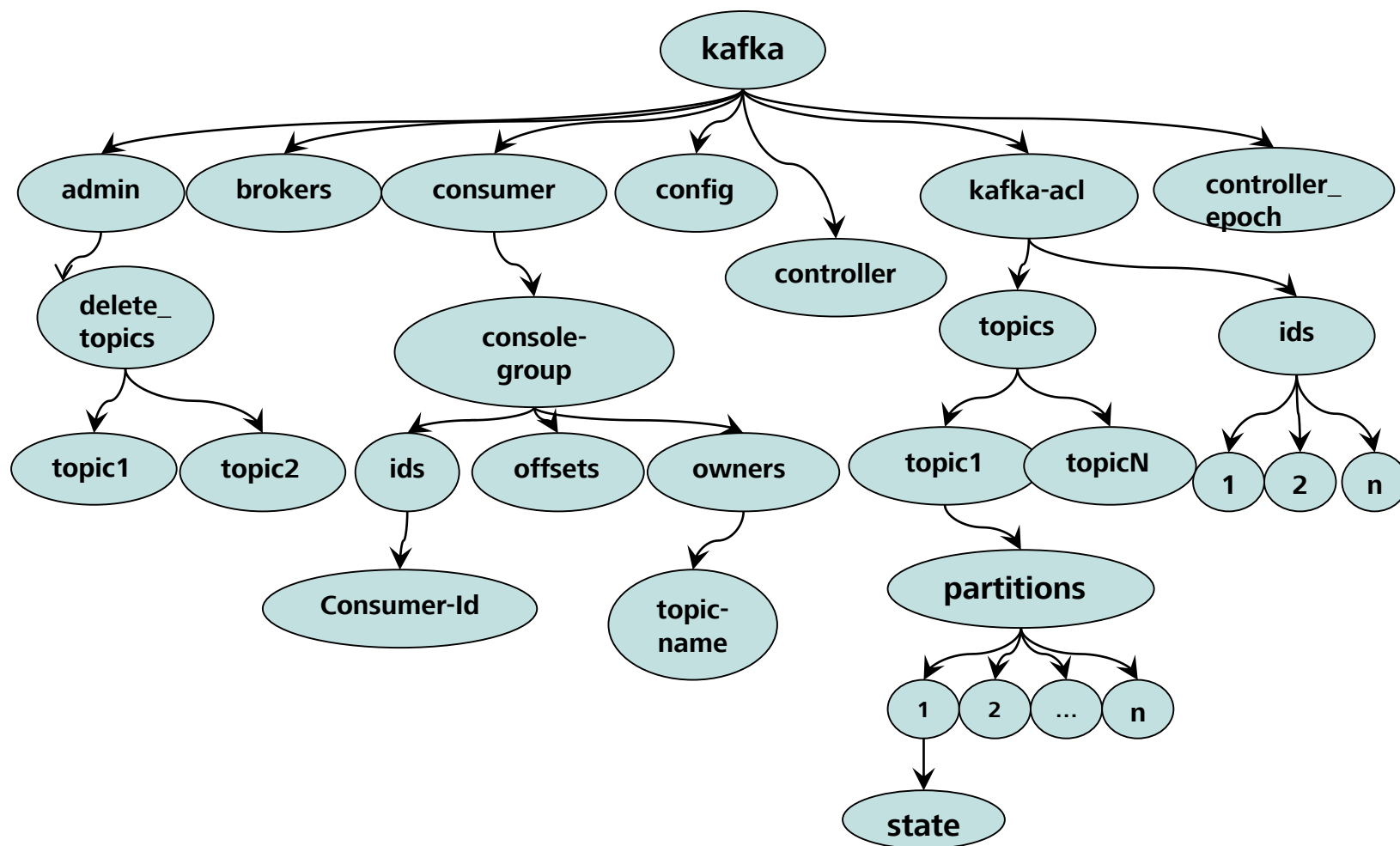
```
[zk: 192.168.0.90:24002/kafka(CONNECTED) 2] ls /brokers/ids  
[26, 27, 28]
```

```
[zk: 192.168.0.90:24002/kafka(CONNECTED) 4] get /brokers/ids/26  
{"jmx_port":21006,"timestamp":"1434266063915","host":"streaming-90","version":1,  
"port":21005}
```

```
[zk: 192.168.0.90:24002/kafka(CONNECTED) 5] ls /brokers/topics  
[test, test1, test2]
```

```
[zk: 192.168.0.90:24002/kafka(CONNECTED) 6] get /brokers/topics/test  
{"version":1,"partitions":{"1":[26,27],"0":[28,26]}}
```

Kafka in ZooKeeper

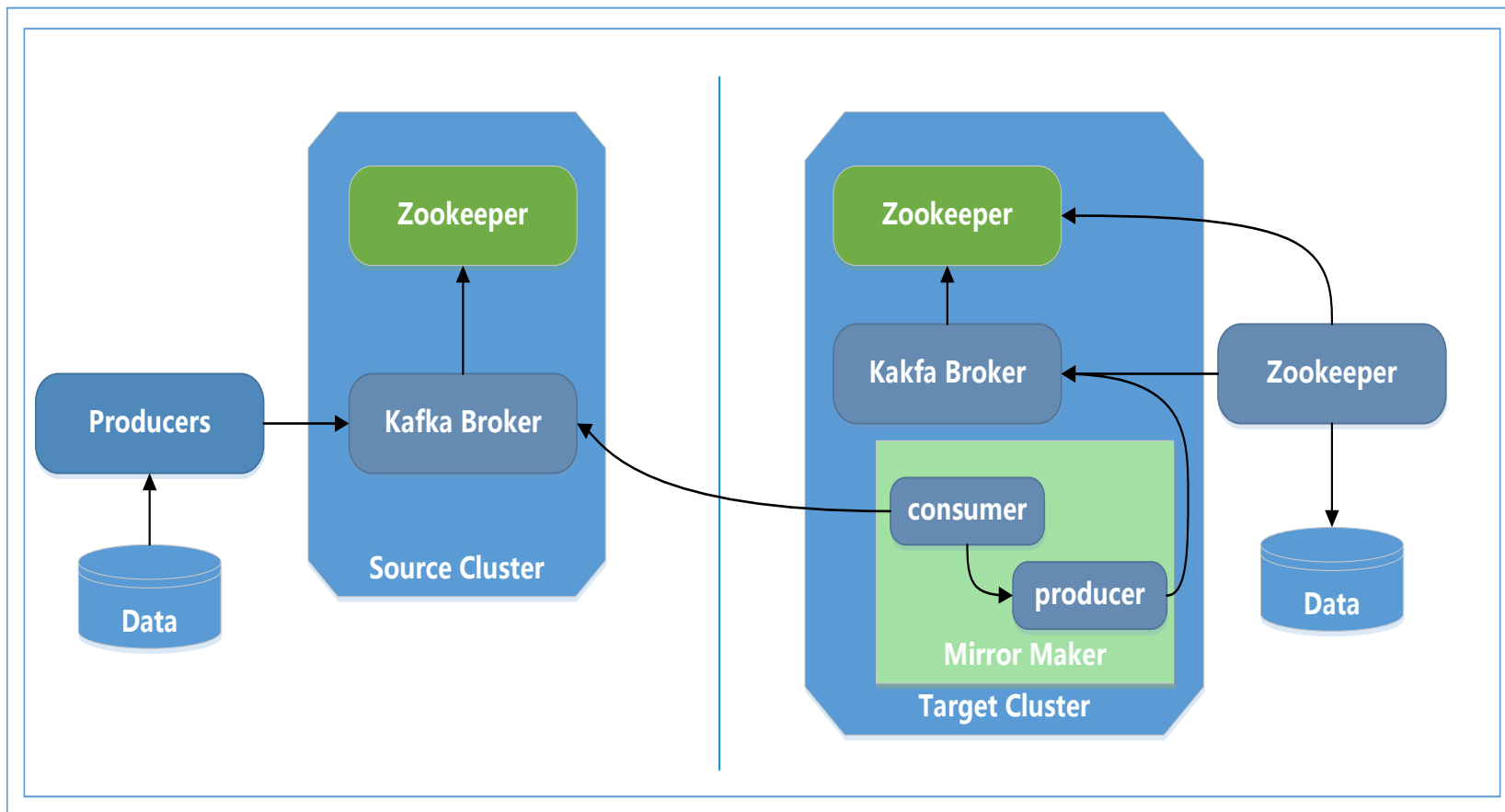




目录

1. Kafka简介
2. Kafka架构与功能
3. Kafka关键流程
4. Kafka在ZooKeeper上的目录结构
5. Kafka高级专题

Kafka Cluster Mirroring





本章总结

本章主要介绍了消息系统的基本概念和**Kafka**的应用场景，以及**Kafka**的系统架构和关键流程，以及**Kafka**在**Zookeeper**中的目录结构，是如何存储的。

习题

1. [多选] 下面哪些关键词是**Kafka**的特点? ()
A. 高吞吐 B. 分布式
C. 消息持久化 D. 支持消息随机读取
2. [单选] **Kafka**集群在运行期间, 直接依赖于下面那些组件?
()
A. HDFS B. ZooKeeper
C. Hbase D. Spark



习题

3. [多选] **Topic Partition**在**Kafka**中是并发单元，通过设置**Partition**数量，**Kafka**提供高吞吐量，以下描述正确的是：
()
- A. **Partition**越多，吞吐量越高。
 - B. **Partition**越多，打开的文件句柄越多。
 - C. **Partition**越多，不可用性增加。
 - D. **Partition**越多，端到端时延可能增加。
 - E. **Partition**越多，客户端内存需要越多。



思考题

4. 通过**Kafka**客户端提供的**Shell**命令可以对**Topic**进行那些操作? ()

5.**Kafka**是如何保障数据可靠的?

Thank you

www.huawei.com