# Kafka测试题和模块作业

## 任务一：Kafka架构与实战

### 1、Kafka的优势有哪些？AB

A、高吞吐量

B、高性能

C、根据Tag进行消息过滤（RocketMQ）

D、遵循AMQP协议（RabbitMQ）

### 2、Kafka应用场景有哪些？ABD

1. 日志收集
2. 消息队列
3. 请求响应模式
4. 流式处理

### 3、下面几项属于Kafka基本架构的是？ABCD

1. 生产者
2. 消费者
3. 主题
4. 分区
5. Router

### 4、属于Kafka概念的是：ACDE

1. AR
2. AI
3. ISR
4. OSR
5. HW

## 任务二：Kafka高级特性解析

### 1、生产者发送数据，客户端需要经过的步骤包括：ACDE

1. 过滤器
2. In-Flight-Request缓存区
3. 分区器
4. 序列化器
5. RecordAccumulator

### 2、Kafka与传统消息系统之间关键区别BC

A、Kafka持久化日志，这些日志默认可以被重复读取和无限期保留

B、Kafka是一个分布式系统：它以集群的方式运行，可以灵活伸缩，在内部通过复制数据提升容错能力和高可用性

C、Kafka支持实时的流式处理

D、支持消息的推送

### 3、kafka为什么那么快ABCD

A、PageCache缓存

B、顺序写；由于现代的操作系统提供了预读和写技术，磁盘的顺序写大多数情况下比随机写内存还要快。

C、零拷技术减少拷贝次数

D、批量处理。合并小的请求，然后以流的方式进行交互，直顶网络上限。

E、Kafka从设计上，延时较小。

## 任务三：Kafka集群与运维

### 1、什么时候发生rebalance？ABCD

A、组订阅topic数变更

B、topic partition数变更

C、consumer成员变更

D、consumer 加入群组或者离开群组的时候

E、生产者停止发送消息的时候

### 2、Kafka的设计是什么样的？ACD

A、Kafka将消息以topic为单位进行归纳

B、将向Kafka topic发布消息的程序称为producer，producer没有横向扩展

C、将预订topics并消费消息的程序成为consumer，consumer可以横向扩展

D、Kafka以集群的方式运行，可以由一个或多个服务组成，每个服务叫做一个 broker.

### 3、Kafka producer如何优化加快速度 ABCDE

A、增加线程

B、提高batch.size

C、增加更多Producer实例

D、增加partition数

E、设置acks=-1时，如果延迟增大：可以增大num.replica.fetchers（follower 同步数据的线程数）来调解；

F、跨数据中心的传输：增加socket缓冲区设置以及OS tcp缓冲区设置。

## 任务四：Kafka源码剖析

### 1、Kafka中Zookeeper的作用是？ABCE

1. meta信息存储
2. consumer的消费状态
3. group的管理
4. offset管理。
5. 选举controller和检测broker是否存活。

### 2、request.required.acks属性值的解释正确的是？C

A、0表示当Leader接收成功时确认；

B、1表示不进行消息接收是否成功的确认；

1. -1表示Leader和Follower都接收成功时确认；
2. All表示当Leader接收成功时确认；

### 3、数据传输的事务定义有哪三种？ACD

A、最多一次: 消息不会被重复发送，最多被传输一次，但也有可能一次不传输

B、不能保证精确一次

C、最少一次: 消息不会被漏发送，最少被传输一次，但也有可能被重复传输.

D、精确的一次（Exactly once）

## 模块作业：

使用Kafka做日志收集。

需要收集的信息：

1. 用户ID（user\_id）
2. 时间（act\_time）
3. 操作（action，可以是：点击：click，收藏：job\_collect，投简历：cv\_send，上传简历：cv\_upload）
4. 对方企业编码（job\_code）
5. HTML可以理解为拉勾的职位浏览页面
6. Nginx用于收集用户的点击数据流，记录日志access.log
7. 将Nginx收集的日志数据发送到Kafka主题：tp\_individual

架构：

HTML+Nginx+[ngx\_kafka\_module](https://github.com/brg-liuwei/ngx_kafka_module)+Kafka

实战步骤：

1. 安装Kafka
2. 安装Nginx
3. 配置ngx\_kafka\_module
4. 开发HTML页面