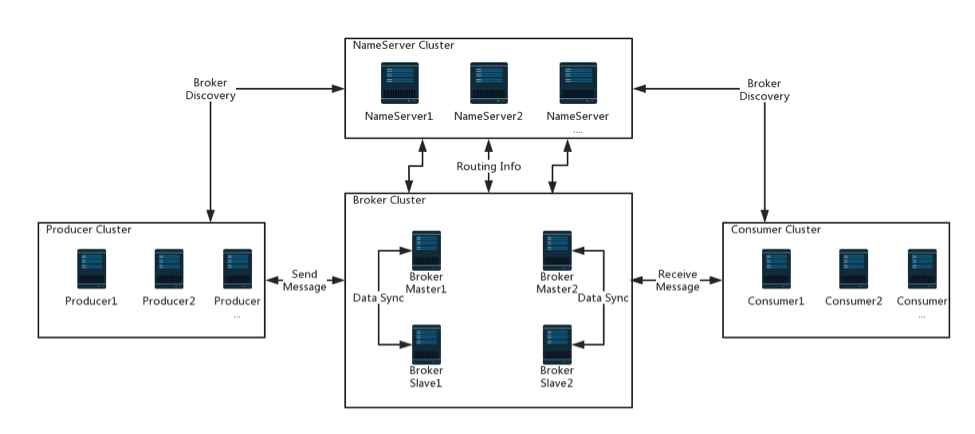
## **Apache RocketMQ架构及核心概念**

### 概述

RocketMQ是一个消息与流处理平台，具有低延时、高性能、高可靠、万亿级消息储存以及可扩展性灵活等特性，系统架构图如下： RocketMQ主要由4个核心部分组成：name servers, brokers, producers and consumers，每一部分都能集群部署避免单点故障。

### 核心概念

#### Producer

消息生产者，通过业务应用系统发送消息给brokers，RocketMQ提供了多种消息发送范例：synchronous（同步）, asynchronous（异步） and one-way（一站式）。  
Producer与Name Server集群中的其中一个节点（随机选择）建立长连接，定期从Name Server取Topic路由信息，并向提供Topic服务的Master建立长连接，且定时向Master发送心跳。Producer完全无状态，可集群部署。

#### Producer Group

标识发送同一类消息的Producer，通常发送逻辑一致。

#### Consumer

消息消费者，从brokers中获取消息进行处理，RocketMQ提供了两种消费者实现：PullConsumer和PushConsumer。  
Consumer与Name Server集群中的其中一个节点（随机选择）建立长连接，定期从Name Server取Topic路由信息，并向提供Topic服务的Master、Slave建立长连接，且定时向Master、Slave发送心跳。Consumer既可以从Master订阅消息，也可以从Slave订阅消息，订阅规则由Broker配置决定。

#### PullConsumer

拉取模式的消费者。应用主动控制拉取的时机，怎么拉取，怎么消费等，主动权更高，但要自己处理各种场景。

#### PushConsumer

推送模式的消费者。RocketMQ使用的是长轮询，此模式本质是在内部封装了消息拉取、消费处理以及其他维护逻辑，预留回调接口给用户以便处理消息到达后的逻辑。

#### Consumer Group

标识处理同一类消息的Consumer,通常处理逻辑一致。对于同一消费组的消费者实例，其订阅的topic要完全一致。

#### Topic

可以理解为消息分组的粒度，消息的逻辑管理单位。无论消息生产还是消费，都需要指定Topic。

#### Message

被发送额信息。一个Message必须包含有一个topic，也有可选的属性tag和自定义的键值对信息，比如你可以把业务主键放入到Message中，方便后续消费消息时的业务处理。

#### Message Queue

消息队列，消息的物理管理单位，理解成Topic的子集分区。

#### Tag

消息标签。RocketMQ支持给在发送的时候给topic打tag，因此消费时刻根据不同tag进行不同逻辑处理。

#### Broker

Broker是RocketMQ的主要部件，它从生产者接收消息、储存消息并处理消费者拉取消息的请求，它也保存消息元数据，消费者组、消费进程指标以及topic/queue等信息。

#### Name Server

Name server作为路由信息提供者，Producer/Consumer客户端查找主题得到相应broker列表信息。

#### Message Model

* Clustering（集群，消息被一个Customer消费）
* Broadcasting（广播，消息被所有Customer消费）

#### Message Order

消息有序指的是一类消息消费时，能按照发送的顺序来消费。