



APUSIC  
固若长城  
睿比世界

# 产品白皮书

金蝶Apusic分布式消息队列

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

## 版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

## 商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

# 目录

- 1 背景
- 2 受众与核心能力
  - 2.1 产品定位
  - 2.2 产品受众
  - 2.3 核心能力
- 3 产品优势
  - 3.1 计算与存储分离
  - 3.2 高性能低延迟
  - 3.3 无限制的主题存储
  - 3.4 无缝故障恢复
  - 3.5 队列和流模型的结合
  - 3.6 内置轻量级计算引擎
  - 3.7 统一认证和授权
  - 3.8 高效运维
  - 3.9 多租户隔离
  - 3.10 消息查询
  - 3.11 国密SSL传输
  - 3.12 多协议接入
- 4 产品架构
  - 4.1 概述
  - 4.2 发布订阅
  - 4.3 跨地域复制
  - 4.4 多租户
  - 4.5 权限控制
  - 4.6 事务支持
  - 4.7 消息分块传输
  - 4.8 传输安全
  - 4.9 与其他系统集成
  - 4.10 多副本存储
  - 4.11 多集群传输通道
  - 4.12 黑白名单
  - 4.13 可视化管控台

- 4.14 监控运维
- 5 应用场景
  - 5.1 系统解耦
  - 5.2 客户案例 某计费平台
    - 5.2.1 背景
    - 5.2.2 解决方案
  - 5.3 削峰填谷
  - 5.4 客户案例 某电商网站
    - 5.4.1 背景
    - 5.4.2 解决方案
  - 5.5 物联网
  - 5.6 客户案例 电网智能传感
    - 5.6.1 背景
    - 5.6.2 解决方案

# 1 背景

消息中间件主要用于协助程序之间实现异步通信，在系统架构中起到削峰填谷、异构集成、应用解耦和异步隔离等作用。传统的消息中间件基于JMS规范发展而起，如金蝶天燕于2003年左右推出的金蝶Apusic消息中间件。伴随互联网应用的发展，RocketMQ、Kafka、RabbitMQ等分布式开源消息中间件开始兴起，这些开源消息中间件有的关注吞吐能力，有的关注高性能，不同的产品适用于不同的应用场景，如kafka倾向于数据流处理，主要应用于大数据流处理和日志处理场景。

近年来，随着云计算及微服务架构的流行，分布式消息引擎在物联网、分布式事务、实时计算和大规模缓存同步等场景中的应用日益增多。以银行为例，分布式消息队列可以将交易拆分成多个本地子事务，通过消息队列进行异步处理，提高交易处理的效率和并发性，减少系统同步阻塞和脏读的可能性。分布式消息队列可以用于大规模数据的快速分析和处理，提高风险评估和数据分析的效率和准确性。

2020年，金蝶天燕启动了传统消息中间件面向金融、电信行业的改造升级，将目光投向了新一代金融级消息队列技术，研发新一代消息中间件产品并提供相关的支持服务。

## 2 受众与核心能力

### 2.1 产品定位

金蝶 Apusic 分布式消息队列（Apusic Distributed Message Queue，简称ADMQ）是一款金融级分布式消息中间件，具有多租户隔离、跨集群复制、强一致性、高可靠、高并发、云原生、丰富的消息类型等特性。采用计算存储分离的架构，可灵活扩缩容。支持原生 Java、C++、Python、GO 多种 API。支持以 Kafka、RocketMQ、RabbitMQ 客户端和 MQTT、JMS 等协议接入，从而简化不同业务系统的接入难度。

ADMQ可应用于削峰限流、系统解耦、异步处理、数据分析等场景，性能方面满足海量消息堆积、高吞吐、低延迟需求。提供资源可视化统一管理控制平台和实时监控功能。

### 2.2 产品受众

- 分布式应用架构师和开发人员
- 大数据开发工程师
- 系统监控、运维人员

### 2.3 核心能力

- 消息发布/订阅：一对多消费模式，发布者可以将消息发送到主题，被一个或多个消费者同时消费。同时支持点对点传输模式。
- 跨地域复制：消息的跨地域、远距离、毫秒级实时同步，支持消息的跨地域发布和订阅。
- 多租户支持：集群内置多租户支持，充分利用的基础设施，简化了运维和管理。
- 多语言客户端：提供Java、Go 等多语言客户端。
- 多协议接入：支持原生Kafka、MQTT、RocketMQ、AMQP和JMS协议的接入，无需修改代码即可完成到ADMQ 的迁移。
- 消息解压缩：消息压缩发送，支持lz4格式、zlib格式、zstd格式、snappy格式。
- IP：支持IPV4和IPV6；支持主机双网卡；支持消息传输时指定IP地址。
- 文件传输：支持文件可靠传输，支持断点续传，支持多线程并发发送。
- 消息传输：支持消息可靠传输，支持断点续传。
- 统一的认证和授权管理。
- 优先级队列。
- 消息查询。
- 国密SSL传输。

## 3 产品优势

### 3.1 计算与存储分离

采用计算与存储分离的云原生架构，将消息的存储和服务分开，可实现存储层和服务层的独立扩展。扩容过程无需任何数据再平衡，不会将旧数据从现有存储节点复制到新存储节点，从而降低了网络带宽和I/O的消耗。

### 3.2 高性能低延迟

能够高效支持百万级消息生产和消费，海量消息堆积且消息堆积容量不设上限。单集群 QPS 超过10万，同时在时耗方面有保护机制来保证低延迟，满足业务性能需求。

### 3.3 无限制的主题存储

主题被分割成分片，以分布式方式存储，因此主题的容量不受单一节点容量的限制。

### 3.4 无缝故障恢复

由于服务层是无状态节点，所以某一个节点宕机对整体的生产和消费没有任何影响，集群会根据负载将主题重新分配给可用的节点。

一条消息在被确认之前会被持久化到N个存储节点，数据在多个节点上都可以被访问，并且可以在N-1个节点故障中存活。通过添加新的存储节点来替换失败的节点，不影响整个集群的可用性。

### 3.5 队列和流模型的结合

队列模型主要是采用无序或者共享的方式来消费消息，而流模型要求消息的消费严格排序或独占消息消费。

ADMQ 提供了统一的消息模型和API，做到了队列模型和流模型的统一。在主题级别只需保存一份数据，可被多次消费，为用户提供了更多灵活性，方便用户程序以最匹配模式来使用消息系统。

### 3.6 内置轻量级计算引擎

内置轻量级计算引擎，让开发人员可以使用Java或Python实现函数（处理逻辑），为用户提供了一个部署简单、运维方便的 FaaS（Function as a service）平台。此功能使用户可以享受无服务器计算（serverless）的好处，类似于AWS的Lambda计算。

### 3.7 统一认证和授权

提供统一的可视化界面，管理租户、命名空间、主题的创建和销毁，并分配用户相应的使用权限，保证用户的数据安全。

### 3.8 高效运维

拥有管控一体的可视化平台，提供服务可视化部署，简化部署流程，集群资源实现多维度监控告警功能，为实施运维带来简单快捷的操作方式。

### 3.9 多租户隔离

提供多租户间隔离机制，保证租户之间互不干扰，保障用户隐私，同时对每个租户的资源进行管理监控，实时查看生产消费情况。

### 3.10 消息查询

支持在命名空间层开启消息轨迹记录功能，开启后会自动记录消息的接收、存储、消费和消费确认状态，在管控台可查询消息轨迹。

### 3.11 国密SSL传输

在消息传输过程中支持国密SSL加密传输。

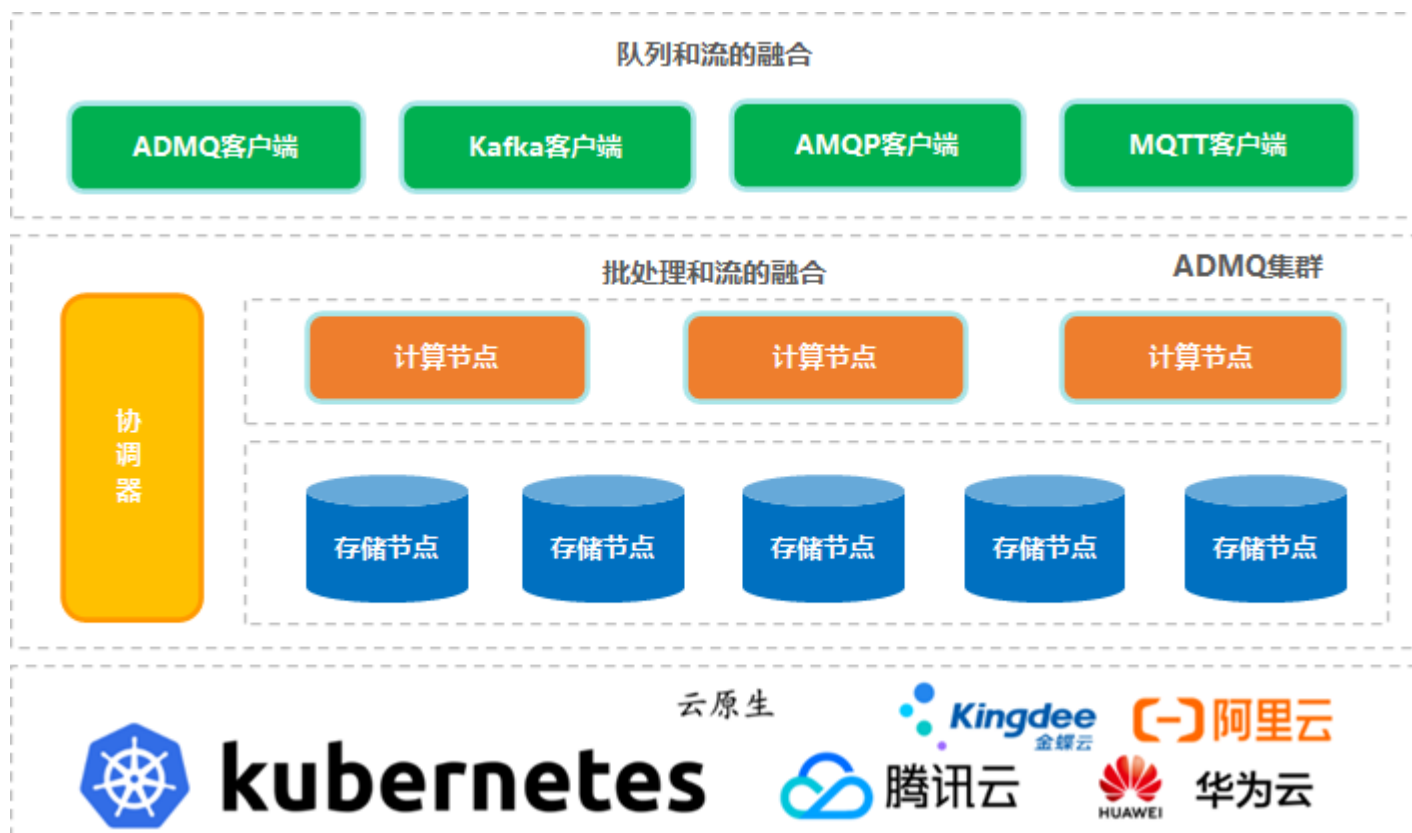
### 3.12 多协议接入

客户端支持 Java、C++、Python、GO 多种语言；另外，还支持 Kafka、RocketMQ、RabbitMQ 客户端的接入，若用户只是利用消息队列的基础功能对消息进行生产和消费，可以不用修改代码就能完成到ADMQ的迁移工作；同时还支持MQTT、JMS 等协议接入。

## 4 产品架构

### 4.1 概述

ADMQ由三部分组成：无状态的服务层，负责处理生产者发出的消息，并将这些消息分派给消费者；存储层负责消息的持久化存储；Zookeeper进行元数据存储、集群配置和协调。



### 4.2 发布订阅

通过“订阅”抽象出了统一的消费模型: 生产者-主题-订阅-消费者。ADMQ的消息模型既支持队列模型，也支持流模型。消息在主题上仅存储一次，但是用户可以有不同的订阅方式来消费这些消息：

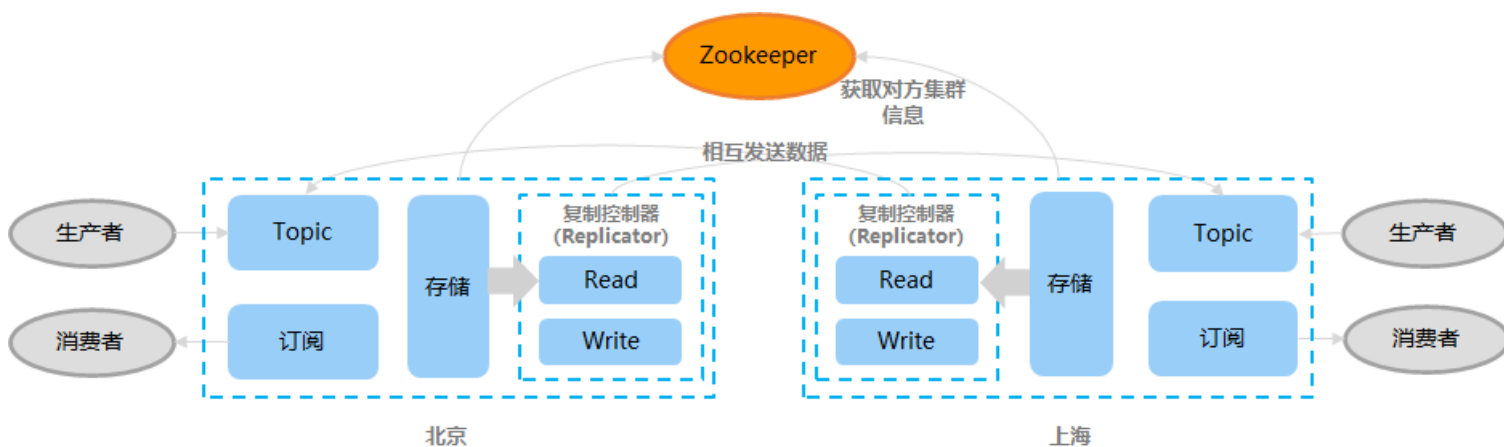
- 消费者被组合在一起消费消息，每个消费组是一个订阅。
- 每个主题可以有不同的消费组。
- 每组消费者都是对主题的一个订阅。
- 每组消费者可以拥有自己不同的消费模型
  - 独占 (Exclusive)：在任何时间，一个消费者组中有且只有一个消费者来消费主题中的消息
  - 故障切换 (Failover)：多个消费者可以附加到同一订阅。但是一个订阅中的所有消费者，只会有一个消费者被选为该订阅的主消费者，其他消费者将被指定为故障转移消费者。

- 共享 (Share)：同一个订阅，用户按照应用的需求挂载任意多的消费者。订阅中的所有消息以循环分发形式发送给多个消费者，并且一个消息仅传递给一个消费者。
- key共享：同一个订阅，用户可根据给定的映射规则，订阅中相同key的消息发送给同一个消费者。

### 4.3 跨地域复制

跨地域复制是将ADMQ中持久化的消息在多个集群间备份。ADMQ在核心服务上构建了跨地域复制功能，用户可以在部署时选择同步或异步配置，并且可以按主题配置复制机制。生产者可以从任何地区写入共享主题，ADMQ负责确保这些信息对各地的消费者均可见。

ADMQ只需进行简单配置，即可在多个方向复制单个主题到任意数量的外部数据中心。ADMQ在后台进行数据同步，由存储层处理，不会影响服务层正在处理的其他工作负载。



北京和上海的集群共用一个全局Zookeeper，用于相互同步集群信息。

生产者发送消息到集群后并存储后，集群会启动复制流程，从存储中读取消息，然后发送到另外的集群。

### 4.4 多租户

每个租户都可以有单独的认证和授权机制。租户也是存储配额、消息 TTL 和隔离策略的管理单元。多租户基础设施可以跨多个用户和组织共享，同时保证它们彼此隔离。一个租户的活动不会影响其他租户的安全或SLA。

多租户可以从两个方面降低成本。首先，简单地共享单个租户没有充分利用的基础设施——将组件的成本分摊到所有用户。第二，通过简化管理——当有几十、几百或几千个租户时，管理一个实例明显简单许多。

### 4.5 权限控制

支持在租户和命名空间上管理和分配用户的权限。当消息生产者和消费者连接到集群之间，需要对生产者和消费者使用的资源进行权限的分配，保障数据的安全性。

支持主-子账户权限模型，子账户拥有的权限是主账户的子集。

## 4.6 事务支持

多个消息发送和接收可包含在一个事务里，ADMQ保证一个事务中的所有消息发送和消息接收的单元操作要么全部成功，要么全部失败。通过对事务的支持，ADMQ可以将多个消息的发送包含进一个事务中，从而轻松地实现诸如关联交易等业务。

## 4.7 消息分块传输

ADMQ对单条消息的长度没有严格限制，在消息传输协议中使用4字节存储消息负载长度，理论上最大支持传输2G的消息内容。

为了保证传输性能，每条消息的长度最好限制在10M以内。对于比较大的消息，ADMQ支持分块传输的方式，在客户端侧把消息切割成多个消息块，当所有的消息块都传输成功后，ADMQ再按顺序把所有消息块组装成一条完整的消息发送给订阅客户端。对于基于其他例如MQTT协议传输的消息，消息最大长度是协议本身限制的，ADMQ不会做任何限制。

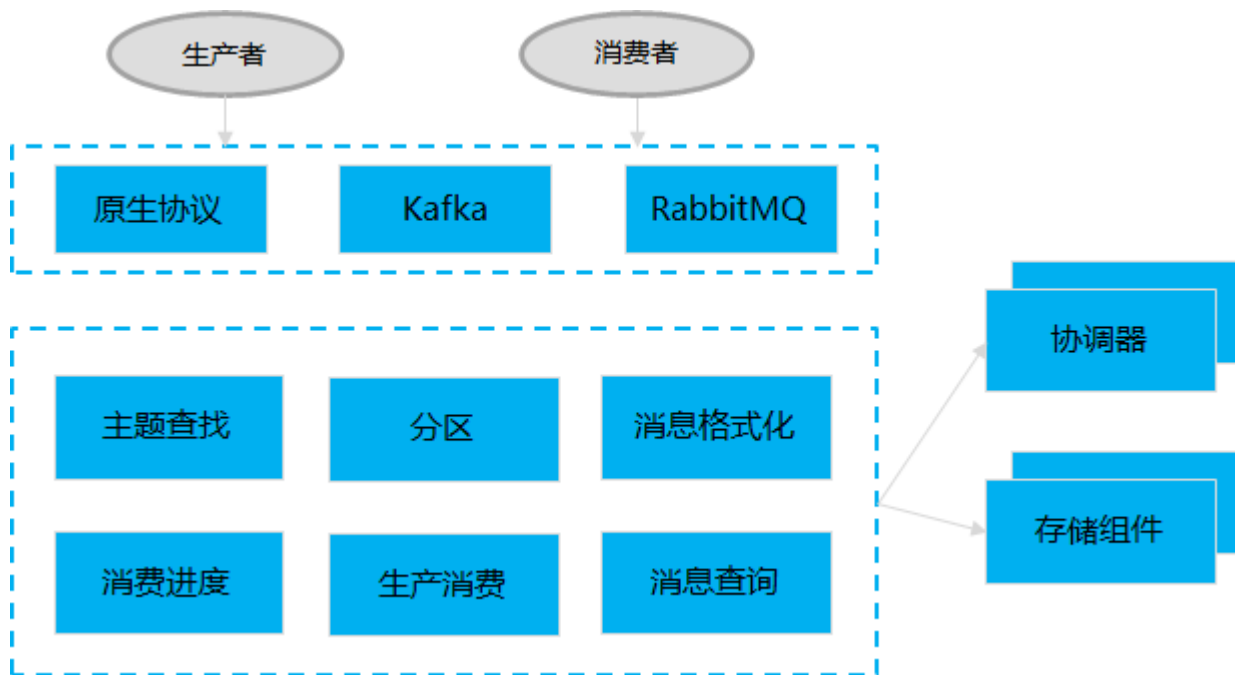
## 4.8 传输安全

支持TLS协议，从而实现客户端与服务器之间、以及服务器与服务器之间的传输通道安全保护；支持国密SSL。

支持传输过程中的消息进行加密。生产者和消费者可以提前约定私钥，生产者发送消息时对消息进行加密，消费者通过私钥对消息进行解密，从而保证消息在传输过程中的安全性，也能避免当其他无密钥的消费者解析消息。

## 4.9 与其他系统集成

ADMQ在消息接收和处理中间添加了中间层 - 协议处理框架，可以处理其他协议的客户端发送的消息。例如用户可以通过Kafka、RabbitMQ客户端发送和接收消息，从而在不改变原来代码的情况下无缝转换到ADMQ。



计算和存储是相互独立的，存储组件的核心功能就是消息的存储和查询，不会涉及到业务相关的逻辑。

计算部分把涉及到消息处理的部分抽象出API接口，包括消息的读写、Topic发布和管理、分区处理、消息内容格式化、消费进度和消息查询等功能API的抽象。当客户端使用其他协议接入时，就可以很方便的把对应的操作转换成抽象API中的方法。

这种方式能很容易的适配其他协议的客户端，目前支持Kafka协议、AMQP协议、JMS协议和MQTT协议。

## 4.10 多副本存储

ADMQ接收到消息后可以把消息副本存储到多个节点上，同时可基于地域或者机房的策略存储，保证副本平分到不同的地域和机房，从而提高数据的可用性。

## 4.11 多集群传输通道

ADMQ内置了跨集群消息传输功能，当其中一个集群收到消息后自动转发到其他集群，从而实现多集群、跨网络的消息传输功能。

支持租户、命名空间的维度配置消息的转发目的地。

## 4.12 黑白名单

ADMQ支持IP、网段的黑白名单过滤，严格控制客户端访问的IP或网段。

## 4.13 可视化管控台

集群管理：部署管理ADMQ、RocketMQ集群；证书管理、依赖管理、插件管理，节点管理；配置集群信息、配置节点参数；自动扩容；节点资源监控等。

用户管理：新增用户信息、编辑用户信息、删除用户信息、修改用户密码、查看令牌、查看权限。

通道管理：跨集群间的消息复制可视化界面，包括：新增通道信息、编辑通道信息、删除通道信息。

资源管理：管理监控租户、命名空间、主题。

消息查询：监控查询消息的轨迹。

系统运维：监控运维、系统告警、功能验证、节点日志。

系统配置：服务器管理、license管理、软件包管理、默认配置管理、操作日志、事件管理。

插件管理：RocketMQ管理、Kafka管理、RabbitMQ管理。

## 4.14 监控运维

管控台自带的监控运维模块可用于监控管理集群信息并以图表的形式直观显示出来。

集成auc登录平台和amp监控平台，集群节点可通过管控台监控自动跳转到amp监控图标界面，对节点资源情况进行监控。

对于ROP插件、KOP插件，管控台分别从租户、主题、订阅组三个维度对资源生产消费情况进行监控；对于AOP插件，管控台从租户、交换机、队列三个维度对资源生产消费情况进行监控。

提供HTTP监控接口。

## 5 应用场景

### 5.1 系统解耦

各个业务系统仅需要处理自己的业务逻辑，发送事件消息到消息队列。下游业务系统直接订阅消息队列的队列或主题获取事件。消息队列可用于单体应用被拆解为微服务后不同微服务间的通信。系统解耦的好处是不同系统的迭代不再相互依赖，能有效缩短数据链路长度，提高数据处理效率。

### 5.2 客户案例 某计费平台

#### 5.2.1 背景

某计费平台汇集国内外主流支付渠道，提供账户管理、精准营销、安全风控、稽核分账、计费分析等多维度服务。平台承载了每天数亿收入大盘，为 180+ 个国家（地区）、万级业务代码、100W+ 结算商户提供服务，托管账户总量 300 多亿，是一个全方位的一站式计费平台。

交易引擎作为计费平台最核心的系统，每笔交易订单数据需要被十几个下游业务系统关注，包括物品批价、道具发货、积分、流计算分析等等，多个系统对消息的处理逻辑不一致，单个系统不可能去适配每一个关联业务。此时，ADMQ 可实现高效的异步通信和应用解耦，确保业务的连续性。

#### 5.2.2 解决方案

某计费平台覆盖 80+ 特点各异的渠道，300+ 不同业务逻辑，单个支付逻辑常横跨众多不同的内外部系统，调用链路比较长，异常出现的概率相对也会比较大，特别是网络超时（比如海外支付业务）。

通过消息队列、消息到期重发，从断点开始继续执行整个交易事务，保证每日亿级交易请求的一致性。

借助 ADMQ 的实时管理能力以及流式计算框架对计费流水进行实时对账和监控，共同保证整个交易的时效性和一致性。

### 5.3 削峰填谷

诸如秒杀、抢红包、企业开门红等大型活动时皆会带来较高的流量脉冲，或因没做相应的保护而导致系统超负荷甚至崩溃，或因限制太过导致请求大量失败而影响用户体验，削峰填谷是解决该问题的有效方式。

### 5.4 客户案例 某电商网站

#### 5.4.1 背景

某电商网站新手机发布在即，拥有预约码的用户可优先购买手机。预约方式为：注册账户即可获得预约码，预计预约用户超过1000万。

像双11秒杀、手机预约抢购等对 IO 时延敏感业务环境下，当外部请求超过系统处理能力时，如果系统没有做相应保护，可能由于历史累计的超时请求负荷过多而导致系统处理的每个请求都因超时而无效，系统对外呈现的服务能力为 0，且这种情况下服务不能自动恢复。

## 5.4.2 解决方案

引入消息中间件，将非即时处理的业务逻辑进行异步化。例如服务接收请求、处理请求和返回请求三个不同的业务逻辑。

当预约活动时，海量并发访问汹涌袭来：

- 所有客户的预约申请，页面均立即返回成功。客户便可关闭网页进行其他活动。预约码稍后推送到客户的邮箱/手机；
- 超过千万级别的注册、预约申请，先暂存在消息队列集群；
- 后端服务进行处理，按照数据库实际的 select、insert、update 能力处理注册、预约申请；
- 处理成功后返回结果给用户。预约结束后，用户大约在5 - 30min内，都收到了预约码。

## 5.5 物联网

## 5.6 客户案例 电网智能传感

### 5.6.1 背景

电网智能传感场景主要基于与电网公司合作的输电线路智能多参数传感器集成研究项目。该项目的传感器来自不同的厂家，分布在输电线路的各个位置，传感器类型因此也不尽相同，包括杆塔、杆塔上、输电线路侧等十多种。整个系统目前接入总长度约六百公里，包含六百多个杆塔的输电线路传感器。这一场景主要负责对各种传感器的数据进行在线监测和告警，同时，我们也单独针对电压传感器做了暂态电压分析。

这个应用场景有两个难点：一是来自不同厂商的传感器没有统一的通信协议，有的使用电力相关的 IEC104 规约，有的使用 protobuf 或其他厂商自定义协议；二是项目数据量比较大，有些传感器可能会单次产生 20 MB 甚至更大的消息，有些传感器则每秒上传一次数据。

### 5.6.2 解决方案

借助 ADMQ，我们选择在 producer 端不做任何数据处理，直接将数据转发到 ADMQ 中，再通过 ADMQ Functions 做进一步的数据预处理和其他业务操作。以电压传感器为例，电压传感器会产生三类数据，分别是心跳数据、稳态波形数据和暂态波形数据。其中心跳数据和稳态波形数据通过 protobuf 协议传输，暂态数据则通过 zip 压缩文件的形式传输。接收到 protobuf 的数据后，借助 ADMQ functions 进行一系列的数据处理，包括通过解密 function 完成数据解密和 protobuf 的反序列化，再对数据进行路由，通过对应的 ETL function 做数据解析和解析，最后通过 Schema Mapping 将数据入库。

全国统一服务热线  
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。

**Apusic**  
金蝶天燕

云计算国家标准制定企业  
金蝶集团旗下基础软件企业  
信息技术应用创新核心企业  
官网: [www.apusic.com](http://www.apusic.com)

