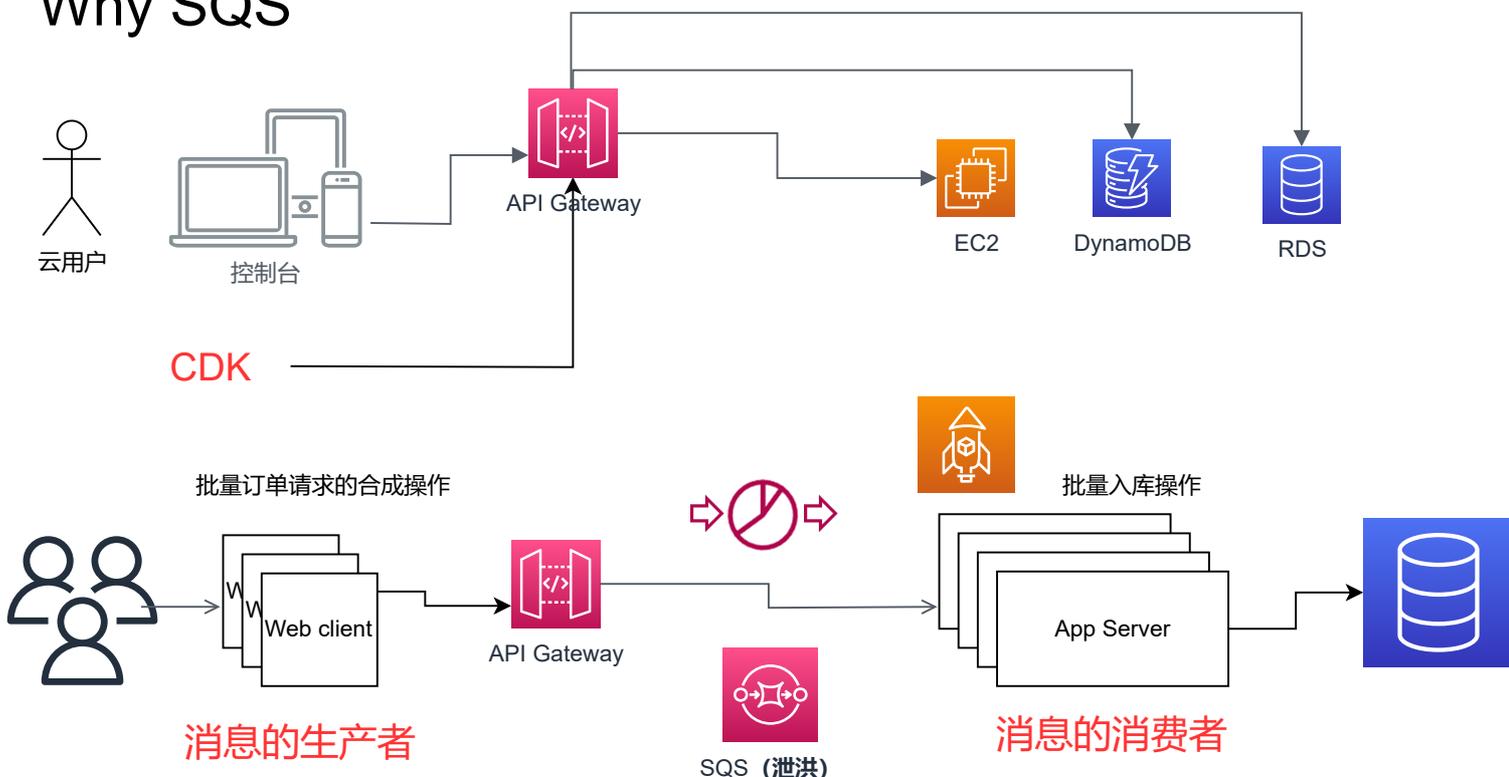


SQS = Simple Queue Service (类比MQ)

产品介绍的思路：Why、What、How、What If

Why SQS



What SQS



队列的处理能力具体包括哪些内容？

- 1、缓冲和分批操作 – 泄洪的场景，类比三峡大坝，有蓄水池的功效。**为您的架构增加可扩展性和可靠性，在不丢失消息或不增加延迟的情况下缓和临时数量峰值。**批处理任务的详细解读：使用SQS可以将处理任务分解成若干批处理任务，支持并行处理，以达到更快的处理速度。**
- 2、请求分流 – 通过将请求排队，将处理缓慢的操作从交互式请求路径中移出。**到一个特定的队列处理。
- 3、弹性伸缩实例 – 使用 SQS 队列可帮助确定应用程序的负载。**与弹性伸缩搭配使用时，您可以根据流量来增加或减少 Amazon EC2 实例的数量。

How SQS

关注的要点操作

标准



一、队列处理方式的配置

FIFO



各有什么优缺点？

- 标准：1、随机异步处理，一个消息会有一到多次处理机会。
2、每秒的 API 调用次数几乎没有限制。类似于CPU对指令的轮询。
3、处理速度比FIFO快。

- FIFO：1、一个消息处理只且处理一次。
2、每秒的API调用数量有限。
3、FIFO 类似于串行处理，相对于标准队列性能低。
4、很可能被一个很大的消息卡死。类似于火车或飞机安检的顺序处理。

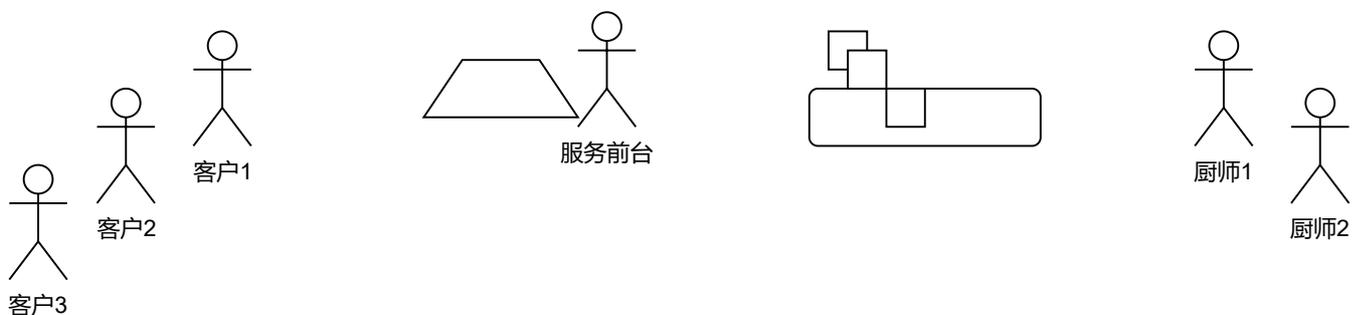
问题：基于应用场景，如何选择用哪种队列？

- 1、如果您的应用程序能够处理不按顺序多次到达的消息,并且对处理性能有更高的要求，则可以使用标准消息队列。比如执行无状态的科学计算或模型训练所关联的消息处理请求。针对客户上传到S3的图片，触发图片渲染操作的消息处理请求，由Lambda去做渲染处理。
- 2、如果您的应用程序适用于操作和事件的顺序至关重要的情况，则使用FIFO消息队列。比如针对有业务处理有紧密先后逻辑的消息处理请求，比如类似于数据库存储过程，按流程步骤先后的数据库写入或修改操作。

二、消息处理的一致性问题的配置

1、关于死信队列死信队列 (Dead-letter Queue)(容错能力)

类比饭店点餐的流程（客户、服务员，大厨），**类比刁钻客户要求特殊配料的汉堡包，但是厨师不知如何处理，往返很多次，还是不知道如何 Offer，只好找专门经理介入解决。**



2、SQS消息的可见性超时，防止重复操作（比如电商的重复扣款）。

消费者收到一条 SQS 消息后,该消息会一直在队列中,直到消费者将其删除。**您可以配置 SQS 队列的可见性超时设置,使该消息在一段时间内对其他消费者不可见。**这有助于防止其他消费者处理相同的消息。**默认可见性超时设为 30 秒。**消费者在处理完消息后,就会将其删除。**如果消费者未能在可见性超时到期之前删除消息,它就会对其他消费者可见,并可以再次被处理。**

三、关于不同业务场景的轮询类型配置

1、轮询的类型

长轮询和短轮询（主要区别在于持续的时间的长短）。

2、长轮询的特点

长轮询持续时间较长，可以达到20秒，它可以最大限度地减少轮询次数，从而提高吞吐量和节省资源。长轮询相对于短轮询，更加经济。

3、短轮询的特点

短轮询的持续时间较短，只有一到几秒钟，它可以快速响应消息，但会消耗更多的资源。

4、缺省的设置

SQS设置界面，关于**接受消息等待时间**为零。如果将接收消息等待时间设置为零，则接收请求将使用短轮询。

5、轮询对应的场景分析

短轮询适用的场景：

成本高：按轮询的次数来收费，成本比长轮询要高。适合于需要立即处理消息的业务模式。比如前端发起支付请求（比如买大额机票），用户应该立刻看到回显界面。验证短信发送请求，应该在秒级立刻有回显，否则就失效了。

长轮询适用的场景：

每次轮询的间隔时长比较长，最多达到20秒。如果有消息，一定会读到，适合于对消息处理的完整性要求高并对时效性要求低的场景。比如大数据相关的计算处理。

What if SQS

一、探讨SQS的两条局限性

(1)、对大型消息的支持：单个消息大小一般为几十到几百个KB，在SQS中建议单个消息不超过1M。

可选的解决方案：

1、如果单个消息大小超过1M，可以存消息在S3上，**S3是Key-Value**的存储方式。

案例场景：

如果前端应用直接产生消息，消息中包括大文件（比如图片文件超过1MB）的处理请求，给后端应用去处理，会造成消息处理的缓慢或卡死。

解决方案：（SQS+S3）

前端客户，针对上传到S3的图片文件需要触发针对此图片的渲染操作，SQS的消息中只需要保留针对这个图片的Key而不是整个图片本身。当消息推送到后端处理时，可以通过后端程序如Lambda执行图片的渲染处理，并把处理的记录写到DynamoDB中。

(2)、对选择**特定消息**的支持：

案例场景1：

在购物网站中，一般需要处理消息的是种类多样的，包括订单生成、订单支付、支付失败和物流更新等消息等。如果特定用户（应用）只关注一种类型的消息，有可能每次轮询都是空转，并且产生不必要的费用。

案例场景2：

针对由SNS推送过来的，基于大数据支持的电子商务产品精准推送的消息，与一般针对普通推送的消息同时进入SQS消息队列。

针对特定人群，业务需要优先推送特定产品消息给特定的客户，需要通过进一步调用程序来进行二次的优先级处理。即SQS本身无法针对特定消息提升其优先级。

话句话来讲，由于**由于SQS不支持多个消息队列中消息处理的优先级设置，必须由指定应用程序来选择先轮询特定的消息。**

解决方案：（SQS+SNS）

建议特定消息建立推送通知机制，通过SNS转给特定人群。