消息组件使用场景介绍：

1、应用场景

MQ是一种非常常见的上下游“逻辑解耦+物理解耦”的消息通信服务。使用了MQ之后，消息发送上游只需要依赖MQ，逻辑上和物理上都不用依赖其他服务。

2、不能使用mq场景

上游和下游是调用与被调用的关系，即上游依赖下游调用的结果

3、是么时候使用mq

场景一:数据驱动的任务依赖 任务间有相互依赖

Task1执行，后

Task2执行，后

Task3执行

常见方案：cron人工排执行时间表

1）task1，0:00执行，经验执行时间为50分钟

2）task2，1:00执行（为task1预留10分钟buffer），经验执行时间也是50分钟

3）task3，2:00执行（为task2预留10分钟buffer）

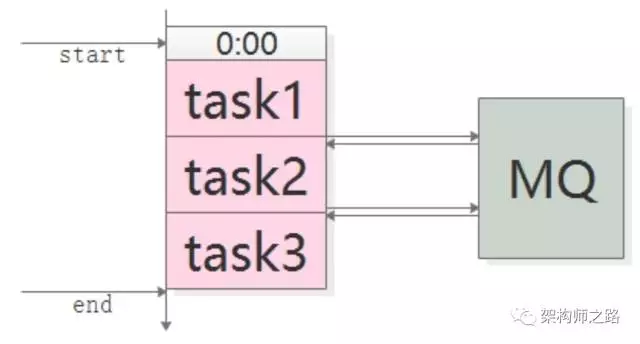
这种方法的**坏处**是：

1）如果有一个任务执行时间超过了预留buffer的时间，将会得到错误的结果，因为后置任务不清楚前置任务是否执行成功，此时要手动重跑任务，还有可能要调整排班表

2）总任务的执行时间很长，总是要预留很多buffer，如果前置任务提前完成，后置任务不会提前开始

3）如果一个任务被多个任务依赖，这个任务将会称为关键路径，排班表很难体现依赖关系，容易出错

4）如果有一个任务的执行时间要调整，将会有多个任务的执行时间要调整



优化方案；采用mq解耦

1）task1准时开始，结束后发一个“task1 done”的消息

2）task2订阅“task1 done”的消息，收到消息后第一时间启动执行，结束后发一个“task2 done”的消息

3）task3同理

采用MQ的**优点**是：

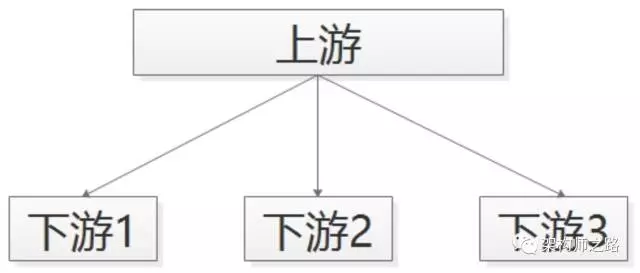
1）不需要预留buffer，上游任务执行完，下游任务总会在第一时间被执行

2）依赖多个任务，被多个任务依赖都很好处理，只需要订阅相关消息即可

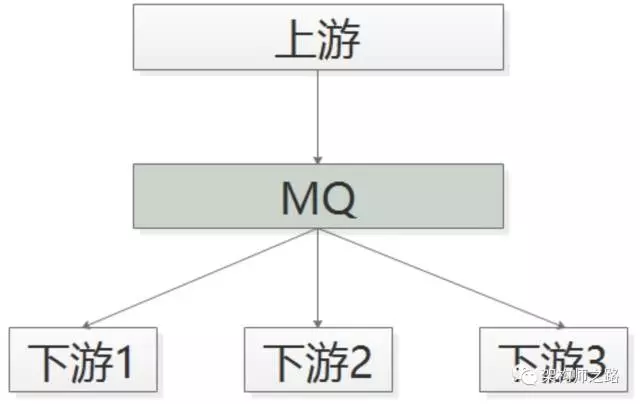
3）有任务执行时间变化，下游任务都不需要调整执行时间

场景二：上游不关心执行结果

58同城的很多下游需要关注“用户发布帖子”这个事件，比如招聘用户发布帖子后，招聘业务要奖励58豆，房产用户发布帖子后，房产业务要送2个置顶，二手用户发布帖子后，二手业务要修改用户统计数据。



MQ解决：



1）帖子发布成功后，向MQ发一个消息

2）哪个下游关注“帖子发布成功”的消息，主动去MQ订阅

采用MQ的**优点**是：

1）上游执行时间短

2）上下游逻辑+物理解耦，除了与MQ有物理连接，模块之间都不相互依赖

3）新增一个下游消息关注方，上游不需要修改任何代码

场景三：**上游关注执行结果，但执行时间很长**

**异步回调**

**总结：**

**MQ是一个互联网架构中常见的解耦利器。**

**什么时候不使用MQ？**

**上游实时关注执行结果**

**什么时候使用MQ？**

**1）数据驱动的任务依赖**

**2）上游不关心多下游执行结果**

**3）异步返回执行时间长 ，回调返回结果**