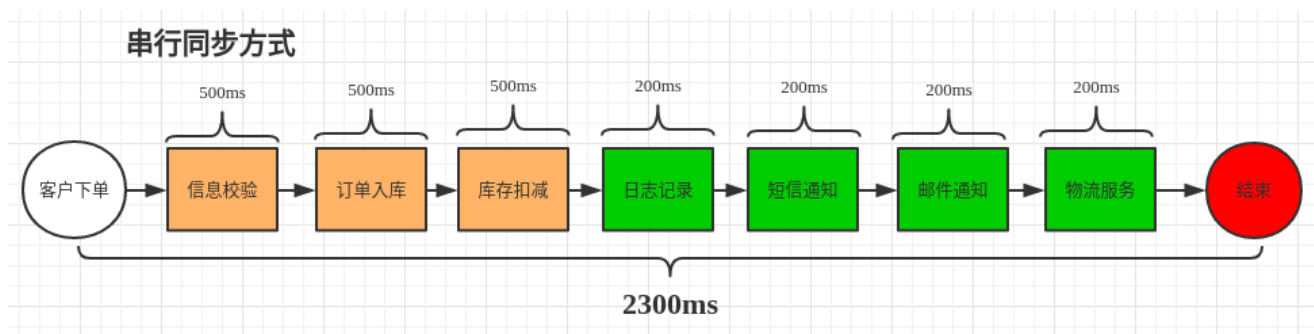


MQ使用技术培训

为什么要使用MQ?

MQ是分布式系统中重要的组件，主要解决应用解耦，异步消息，流量削锋等问题，提供高吞吐量，高效率的通信能力。我们看下传统同步调用方式与MQ并行异步调用方式的对比：

传统的同步调用方式：



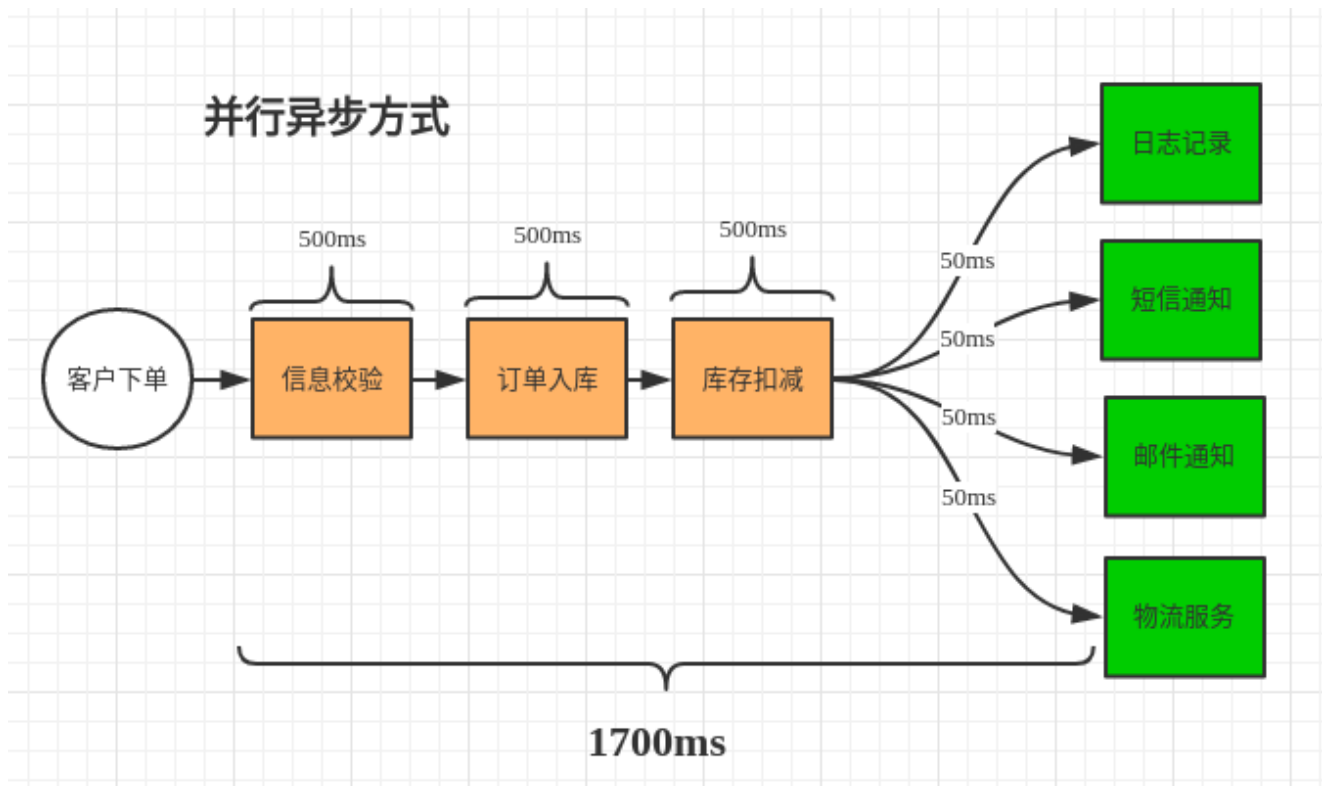
优点：

1. 流程逻辑简单
2. 对于强依赖的业务单元，事务一致性较高，保证主业务的事务一致性

缺点：

1. 整体响应时间过长，容易超时，总体耗时是各业务单元的耗时叠加
2. 弱事务业务单元被迫逻辑强依赖，阻塞整体响应流程
3. 并发量吞吐量较低，应用层线程栈出栈较慢，网络层socket连接不能快速释放

接入MQ后的并行异步方式：



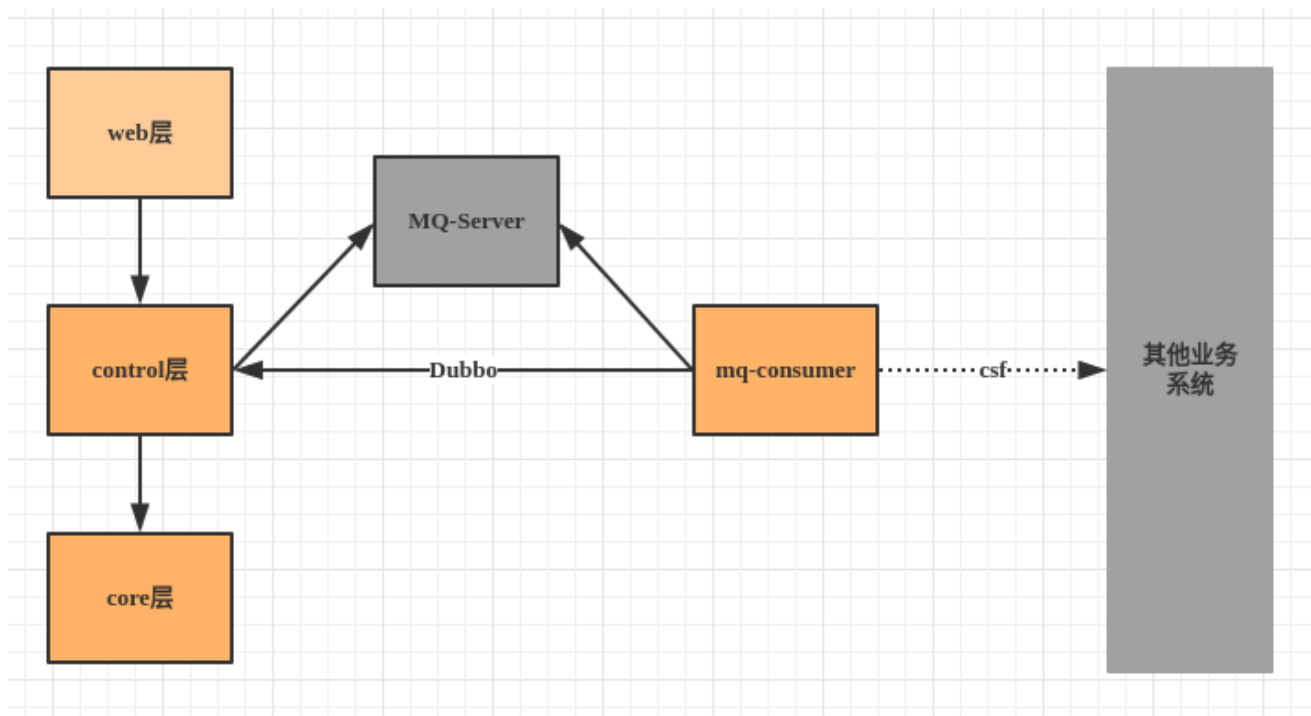
优点：

1. 并发量吞吐量较高，线程出栈快，**socket**连接可以快速释放
2. 应用与MQ服务器长连接，消息传递效率高
3. 解耦弱依赖业务单元处理逻辑，提高响应效率

缺点：

1. 不适用于存在强一致事务的业务单元组合逻辑，如库存及订单服务
2. 存在中间件故障风险，多一层通信介质，就多一层风险，需要业务系统保证业务高可用
3. 业务系统需要通过事务补偿，保证事务的最终一致性
4. 不能作为系统间调用唯一通信介质，容易出现单点问题，由于狭义协议问题导致扩展性不高
5. 不适用大消息传递，如：文件上传及下载

调用关系及部署结构



步骤说明：

1. web层接收用户请求，调用control层dubbo服务
2. control层收到请求后，将请求报文发送至MQ
3. mq-consumer(消费者进程)监听对应Topic，拿到消息后(请求报文)调用control层dubbo服务，如果此功能非本业务系统功能，如短信服务，可以在mq-consumer中调用短厅接口。
4. control层收到MQ消费者进程请求接口后，调用core层服务。

为什么mq-consumer要独立集群部署？

1. 减轻control层资源压力(网络IO，线程)，故障隔离，避免因消费进程问题影响control层主业务。
2. mq-consumer采用多节点集群部署，可提高消息消费吞吐量，放这一起的话，control会被迫扩容
3. 可以不停机发布，方便MQ集群迁移，机房迁移等，便于运维割接

为什么mq-consumer不能调core层服务？

1. 破坏control层服务完整性，core层是原子数据服务，control层是业务逻辑服务，control层可能会调多个core层服务或其他业务功能逻辑处理，如果mq-consumer调core层会破坏业务服务的完整性，导致业务功能及调用关系紊乱
2. 假如以后不用MQ了，会导致mq-consumer的相关功能在control层重新写

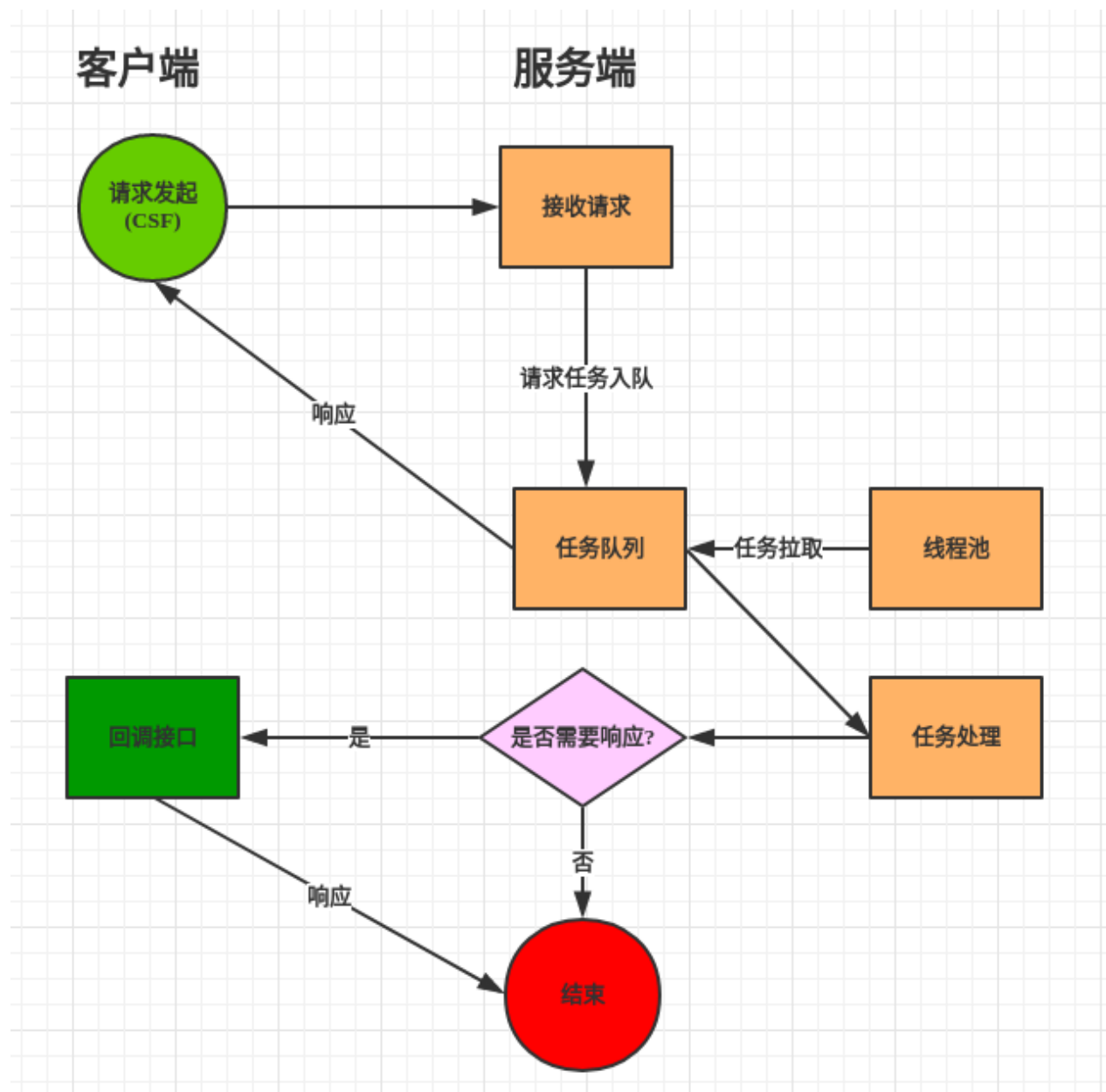
系统间异步调用解决方案

系统间异步调用主要用于处理耗时较高的服务接口，但不能使用MQ作为通信介质，主要有以下 2 个问题：

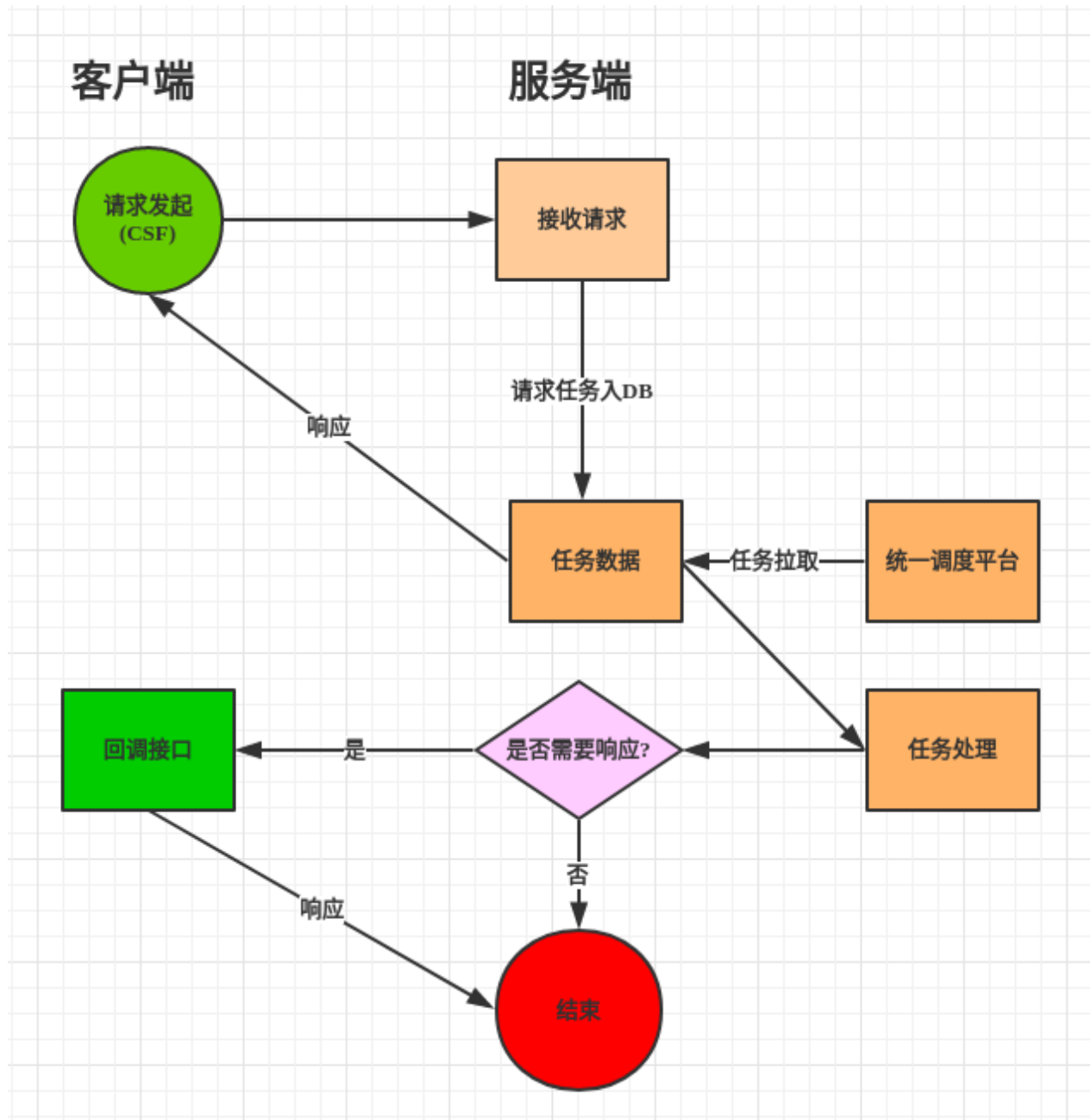
1. 服务提供方无法将服务注册到服务治理平台，无法进行服务统一管理及共享
2. 存在调用单点故障风险

可以采用以下 2 种方案：

1. 异步非阻塞模型,可以使用成熟的Fork/Join并行处理框架



2. 异步回调接口



规范要求

1. 开发规范严格遵守《cmos-msg接入手册-2.0.0-SNAPSHOT.pdf》
2. 部署规范严格遵守《ROCKETMQ4.1.0安装及部署规范.pdf》
3. 禁止MQ作为系统间通信介质，严格遵守CSF调用规范
4. MQ生产者，消费者不能在一个应用中，消费者必须单独拆出独立应用
5. 消费者应用必须调业务层服务接口，不能调数据层服务接口
6. 业务系统需要保证事务的最终一致性
7. 应用中的mq地址必须是域名
8. 不能用于存在强一致事务的业务单元组合逻辑，如库存及订单服务
9. 不适用大消息传递，不能超过4M,如：文件上传及下载功能
10. mq-consumer不能写业务逻辑功能代码，只能转发调用control层或其他业务系统接口。
11. mq-consumer必须集群部署，至少2个节点，可以根据消息量再进行扩容
12. 对于mq-consumer直接调core层部分业务系统，已经上生产的暂不进行改造，新接入的按此规范接入

