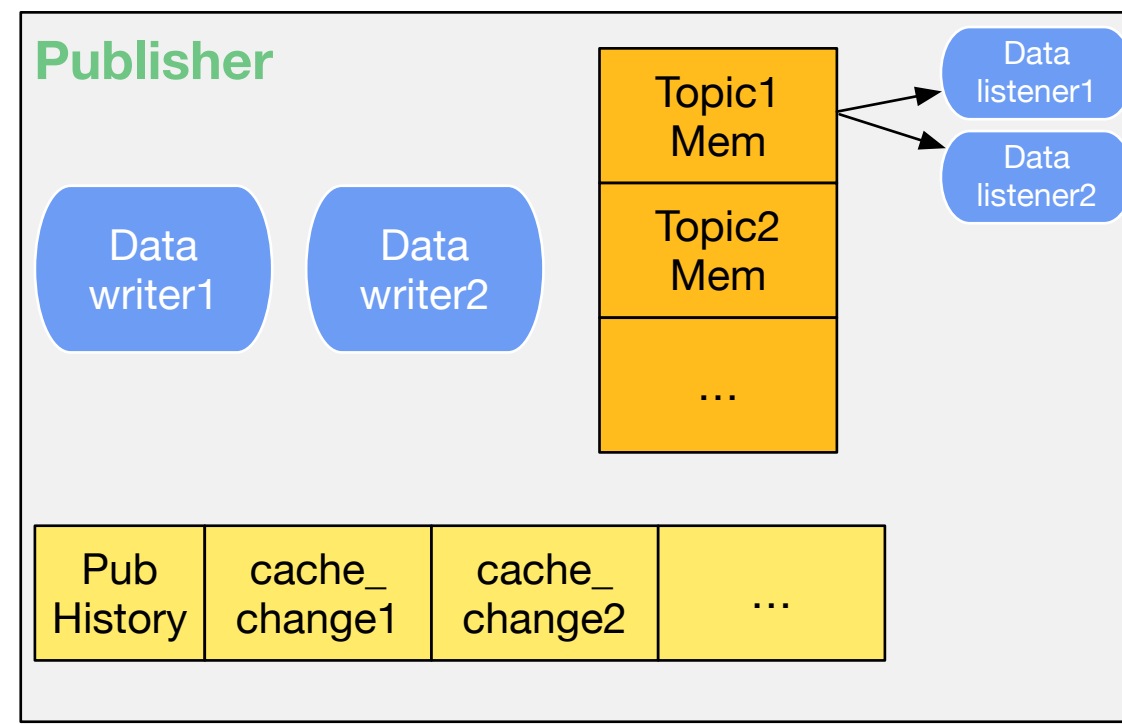
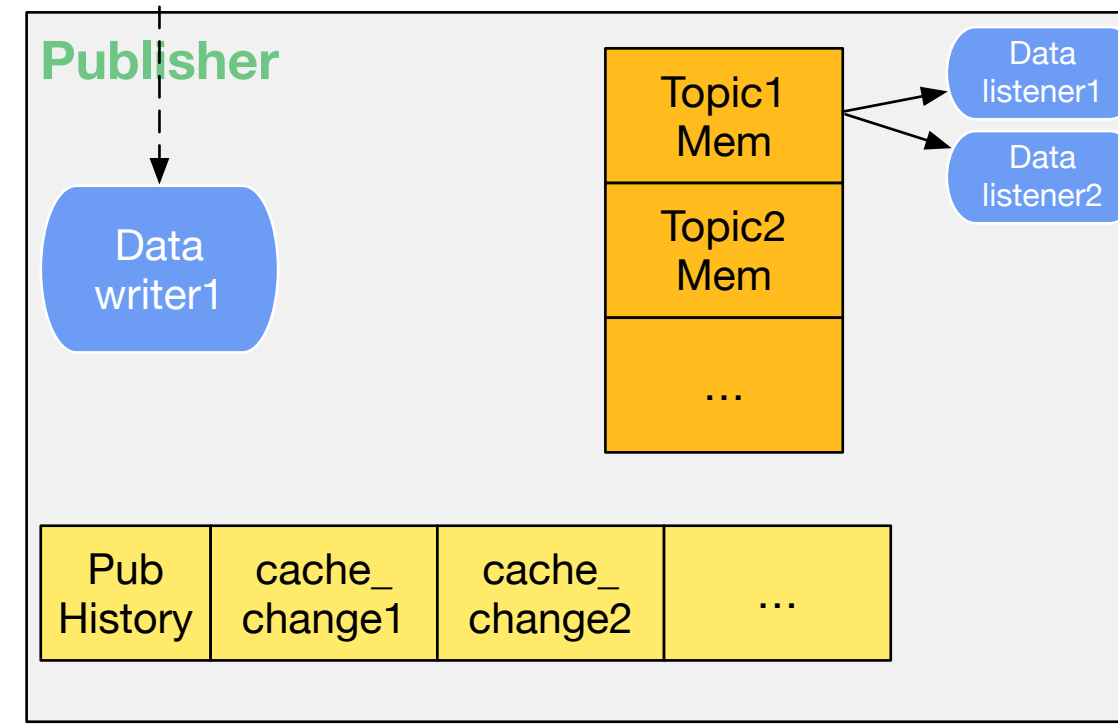


DDS: FAST-RTPS



ROS2-pub



ROS2

在用户执行pub
的callback线程
中执行

ROS2-pub publish `msg`:

1. 先从DDS-Publisher获得`msg`大小的内存空间
2. 数据序列化serialization
3. 数据添加到Pub_history_mem_pool中
4. 根据Topic中的数据，**通知**相应的Data listener (ROS2-sub)

同步通知：(以下操作仍发生在相同线程)

1. 数据**拷贝**至listener (sub) 自己的history_mem_pool中
2. 修改一些标志位，以便executor更新wait_set(..)

异步通知：(default)

1. 触发单独线程完成上述数据拷贝和修改标志位工作

不清晰的点

1. 访问history_mem_pool中间涉及到多个锁，行为如何？
2. 同步通知会否发生线程切换？
3. 异步会否出现pub发送消息占用history_mem_pool, 上一轮sub正在取消息的情况？

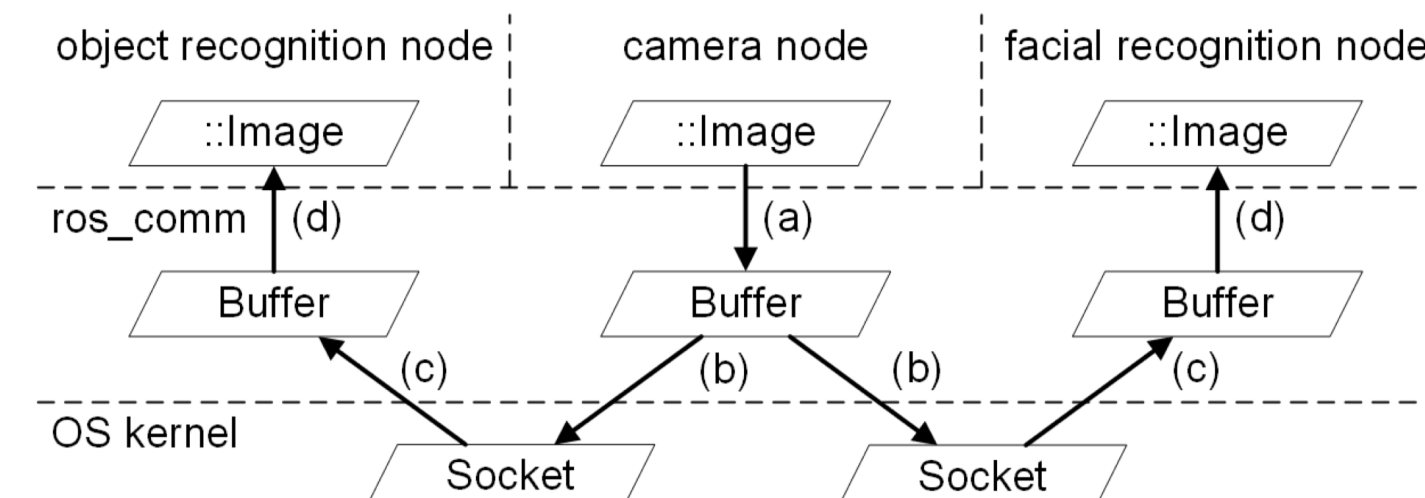
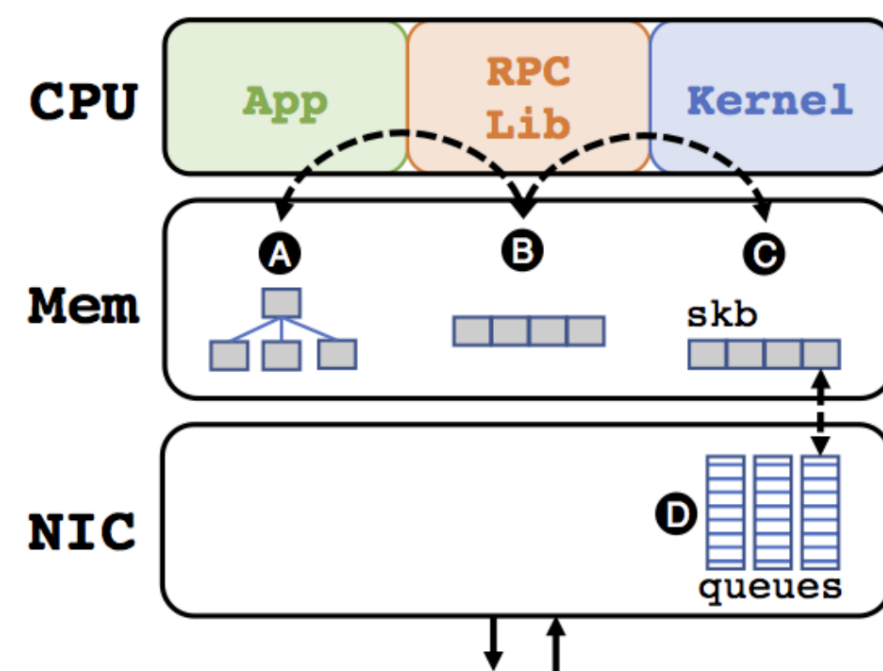


Figure 1: Memory copies in a traditional network stack.

现在的inter-comm, 一般采用socket-based IPC通信, 即

1. 序列化后的`msg`会从DDS的pub_history_pool中拷贝至内核Socket的buffer内存中
2. 如果是同个机器内通信, 不会进入到NIC的buffer, 通过一个idle-loop返回。
3. 如果不在同一个机器内, 则通过NIC线程将数据发送出去