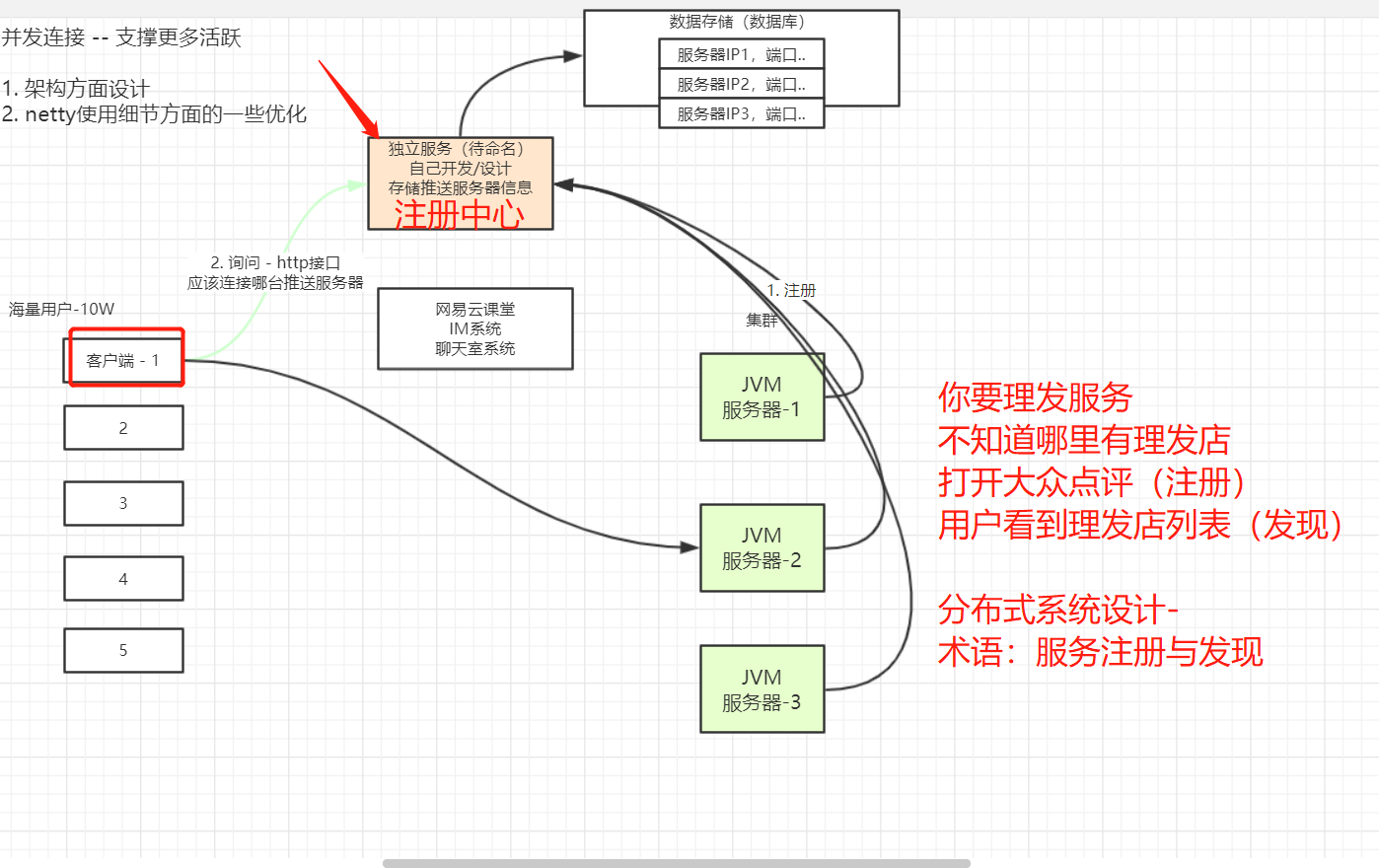
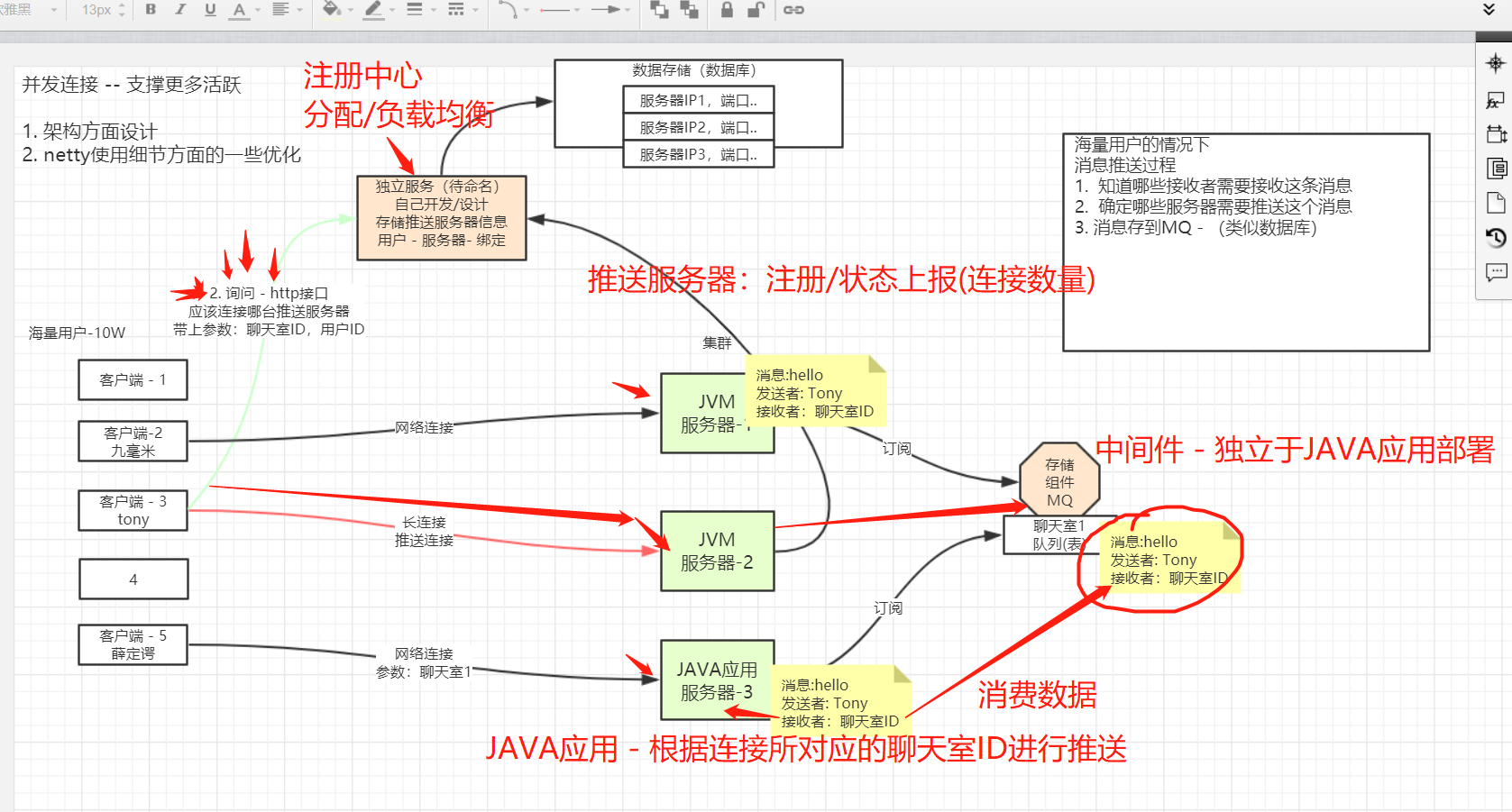
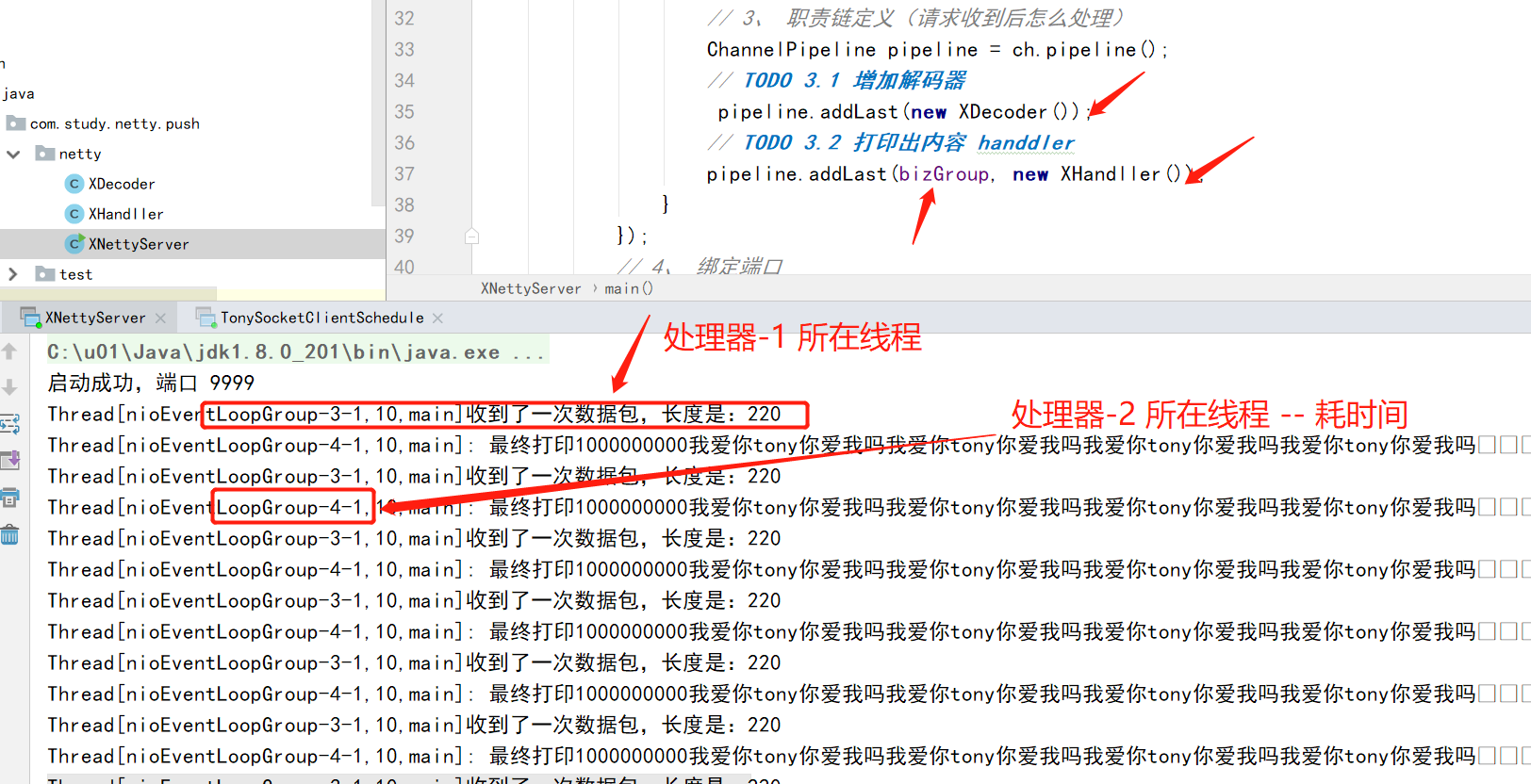
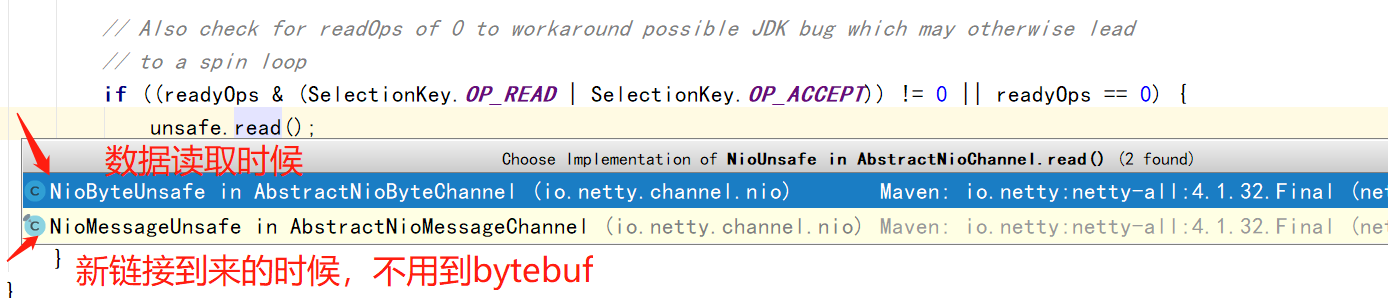
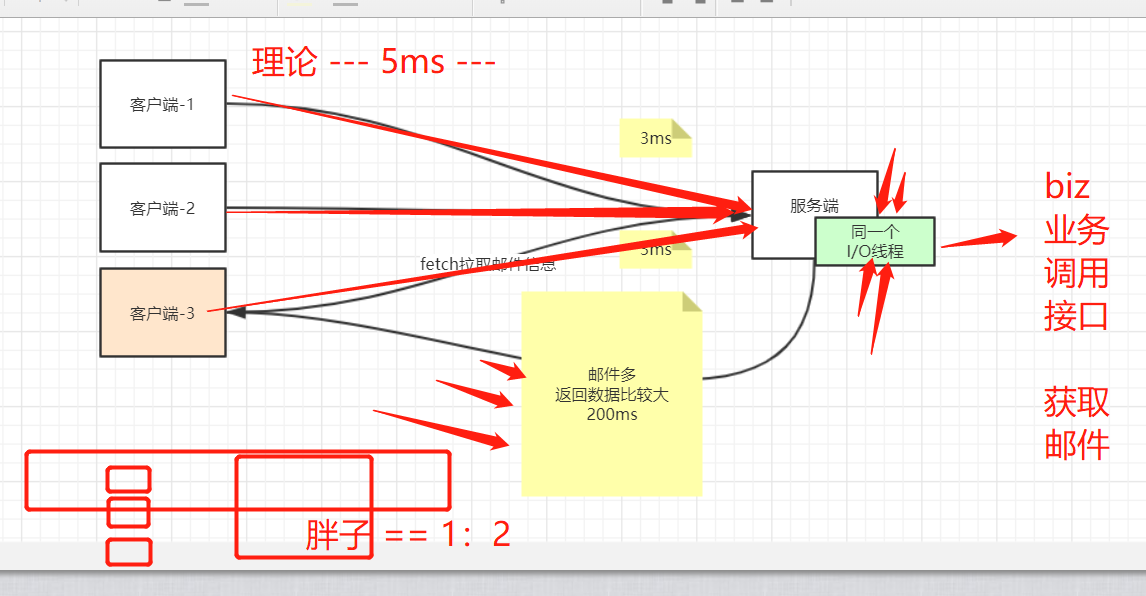
从架构层面规避掉 – 单体应用 容量问题





# 具体netty相关优化

核心三个要点：线程模型、职责链、ByteBuf（复用、零拷贝）

1. 业务上的耗时操作，通过单独创建线程池进行处理，不要占用宝贵I/O线程  
     
     
   
2. Pipeline机制，有一项需要优化：handler复用（单例）  
   注意: 共享之后，多个线程操作同一个对象的属性，谨慎使用共享  
   
3. Netty本身就是有很多优化 – ByteBuf （零拷贝、对象复用、内存复用）  
     
   注意： 调试阶段，你就去防止。Debug一下  
   ---- 类名@hash ----   
     
   记得release
4. 大数据数据写入， 分成多批次，小数据包的方式。  
   

后续同学们碰到 还是可以直接发群里，联系Tony老师….  
技术支撑  
  
积累 --- 需要有特定的业务场景，特定问题，特定的解决方案 --