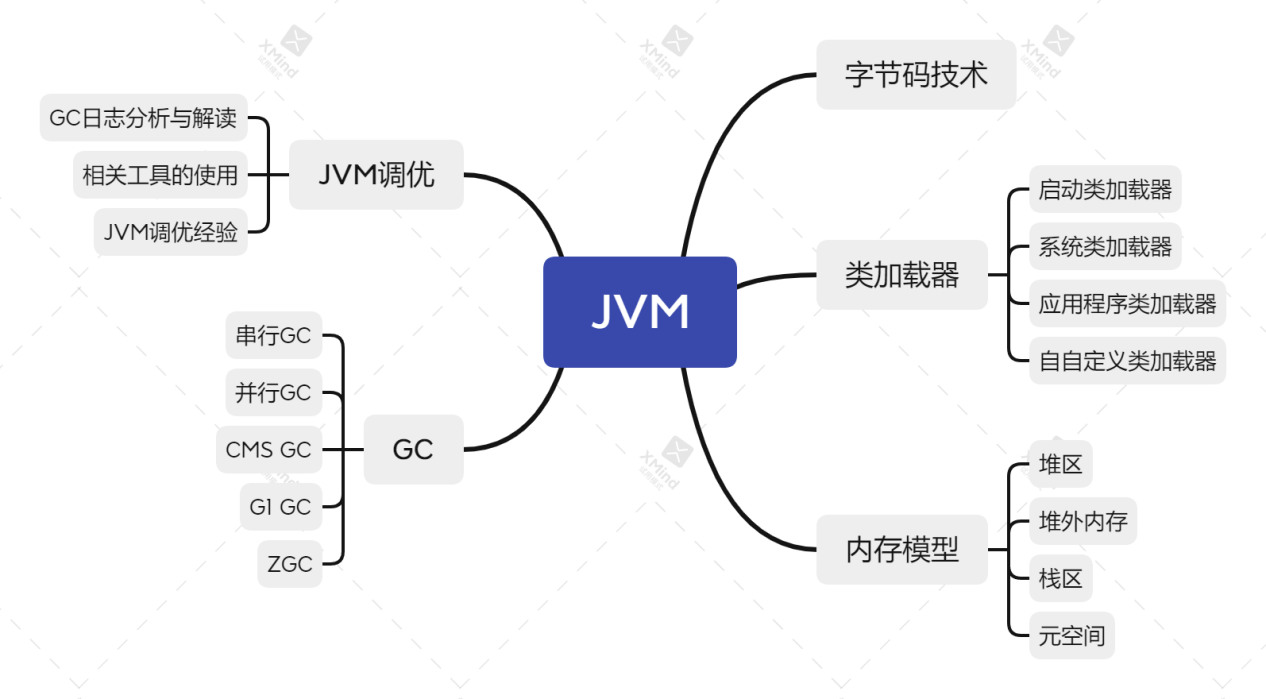
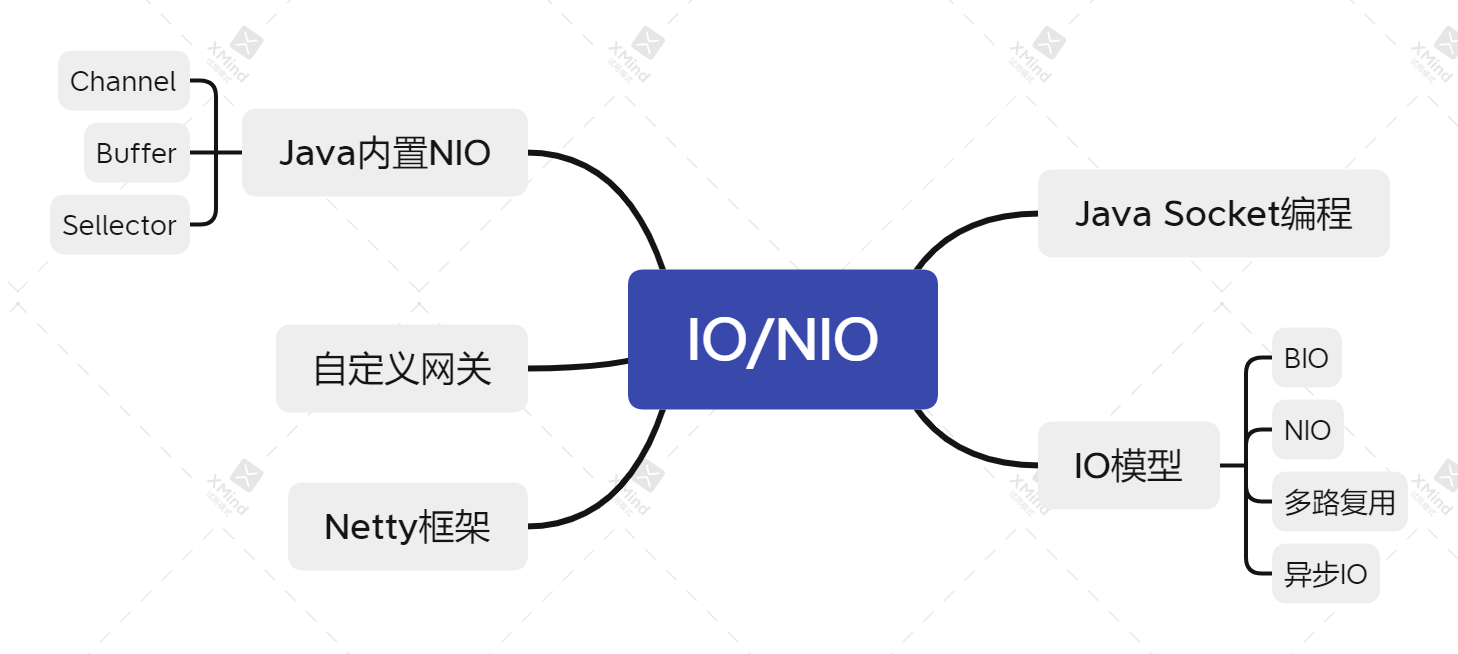
# JVM



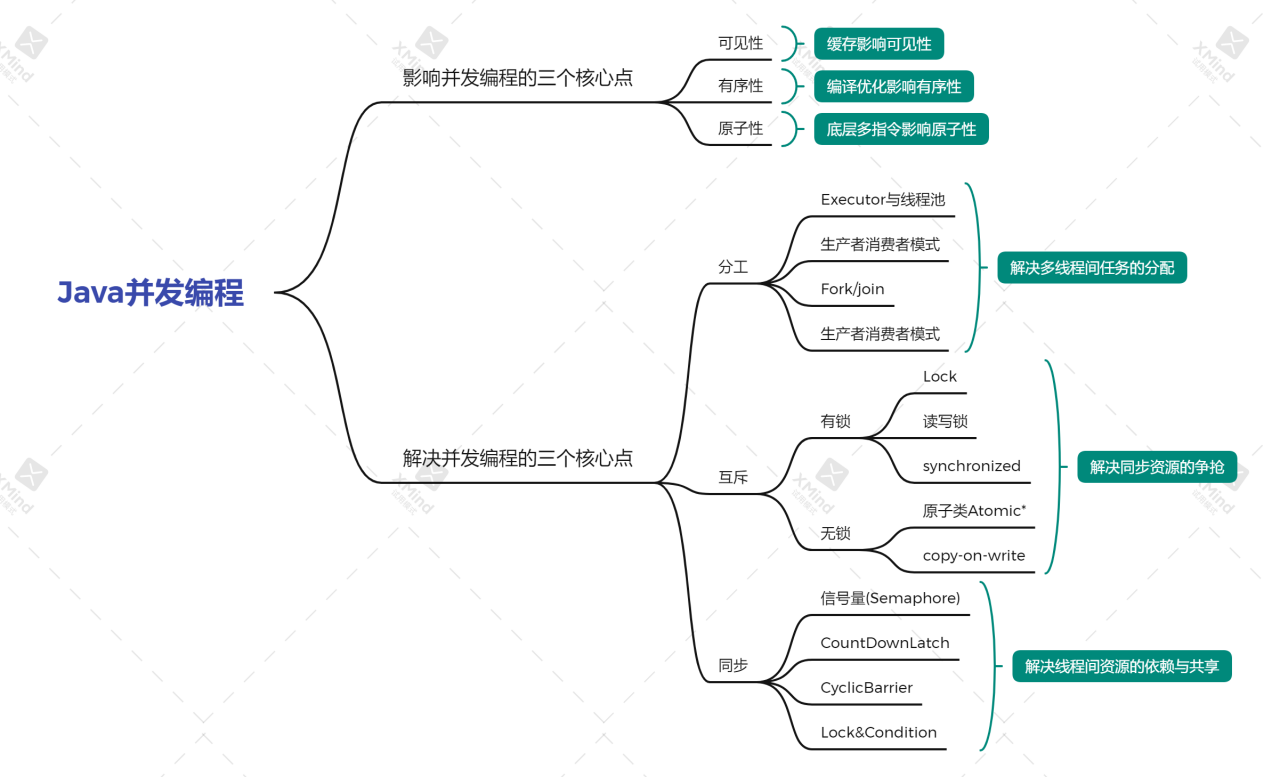
JVM是训练营的第一课,作为java开发,不了解jvm的组成及相关运行机制是无法在工作中排查一些潜在的问题的,更无法排查类似系统性能问题,服务意外宕机问题,内存泄漏问题等,同时了解JVM也有助于代码的编写,一点点的小优化累计起来就是性能提升的基础.通过学习JVM这一节课的内容,学会了JVM内存模型,了解内存模型后,理解了为什么java在全世界如此受欢迎,学会了如何设置堆区大小,元空间及栈的深度;了解了GC的基本原理以及不同GC的特点,在不用场景下应该如果使用;最后学习了GC日志的分析,知道了STW触发的时机以及GC回收的细节,掌握了几种日志抓取及分析工具,可以应用到日后的工作中解决和分析JVM相关问题.

# NIO



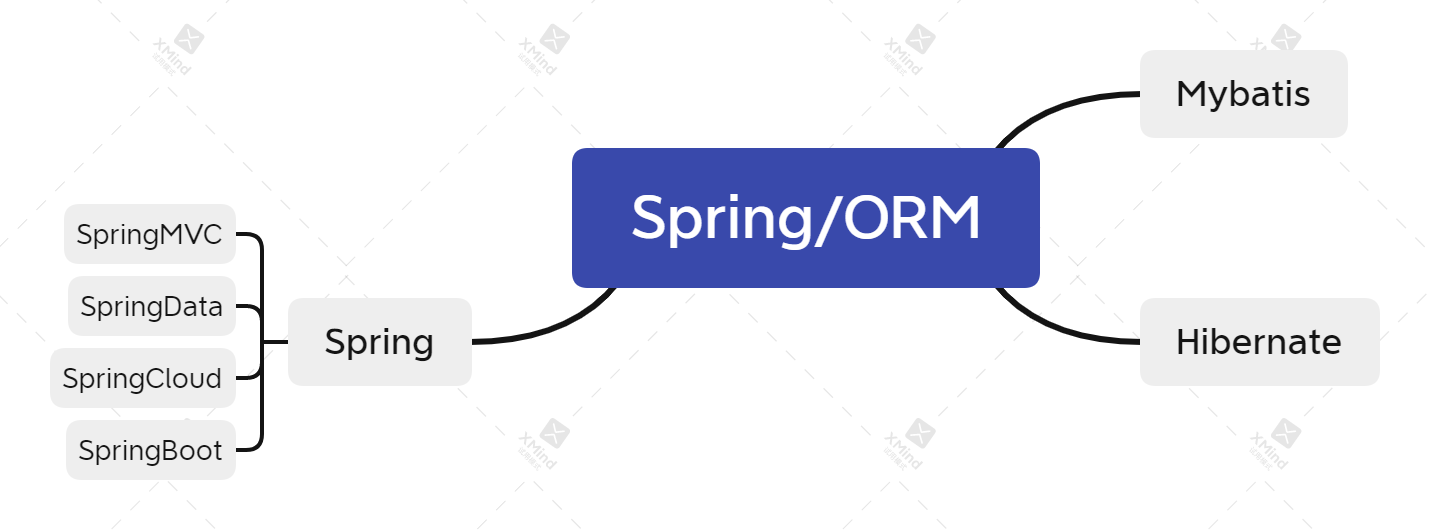
IO是java开发不可或缺的一部分,性能是IO开发中必须重要关注的内容,尤其是网络IO中,通过本节课的学习,复习了java基础socket编程,掌握了java内置的nio相关的API,学习了多种IO模型,多路复用是当前主流的解决方案.学习了netty框架,netty框架是一个基于多路复用的网络编程框架,也是目前最重要的网络编程框架,在此基础上开发了自定义网关,学会了如何使用netty框架.

# 并发编程



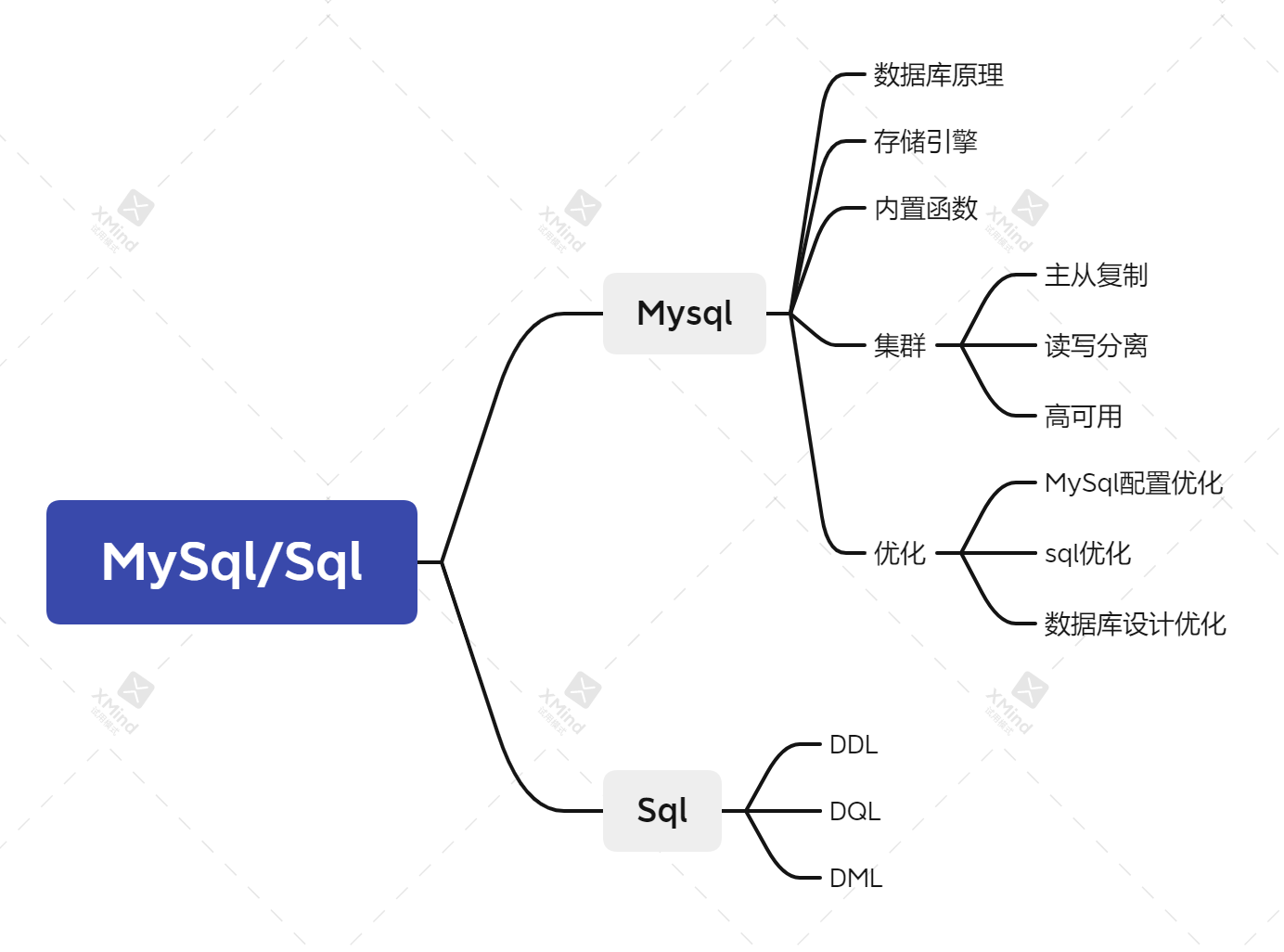
并发编程是java中高级开发的重要基础,同时并发编程也是java开发中最难处理以及最难调试的一部分,通过本节课的学习,掌握了多种并发编程工具,如CyclicBarrier,Fork/Join等,理解了分时多核cpu带来的并发基础问题,包括可见性,有序性,原子性.解决这些问题就需要处理分工,互斥与同步,java提供了多种工具来帮助开发人员更方便的进行多线程开发,包括读写锁,原子类,信号量等.完整的学了并发编程相关原理及工具后,对并发编程的本质也有了新的理解,写好并发代码需要首先解决共享资源的分配,围绕并发资源的分配加上java内置工具就可以有效的解决并发问题了.

# Spring和ORM框架



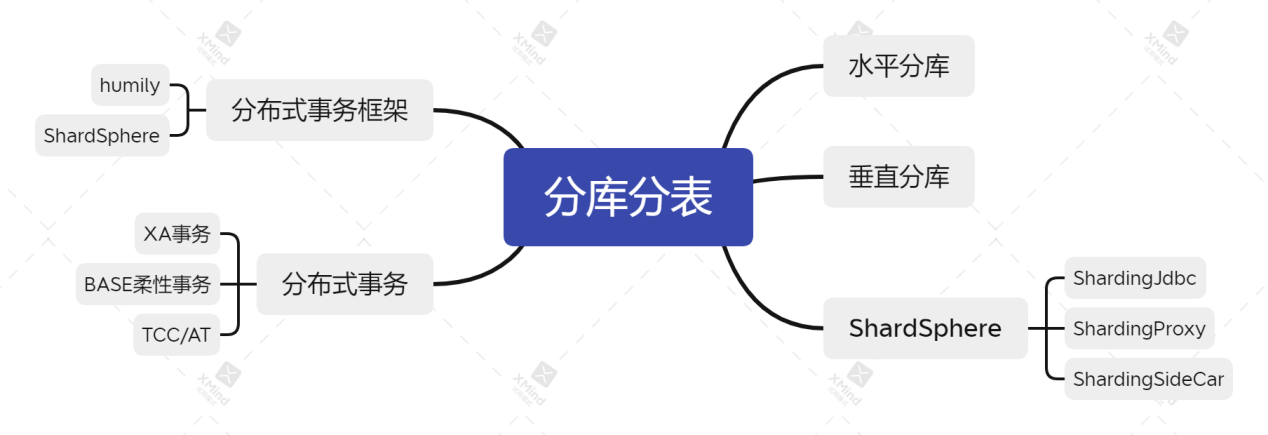
Spring是java程序员必须掌握的一部分,Spring通过其强大的生态使java开发变的更快速更便捷,可以在短时间内快速搭建项目,SpringCloud全家桶提供了微服务全套的功能,包括注册中心,网关,熔断与限流,监控等功能,SpringData封装了一些缓存及数据库的相关操作,如RedisTemplate,JdbcTemplate等,Springboot采用适配器设计模式,提供了多种中间件的启动器,使开发人员在引入中间件时可以更容易.Mybatis与Hibernate是两个最重要的ORM框架,Mybatis通过xml及注解,将sql从代码中分离出来,使开发和维护可以变得更清晰更易维护,Hibernate通过字段与类属性的映射,可以在不手动编写sql的情况下完成对数据的僧删改查,简化了开发步骤及sql编写错误的概率.框架是简化开发的重要手段,学会使用框架可以是代码开发事半功倍,同时也要学习框架源码,防止出现问题时无从下手,要做到知己知彼.

# MySql数据库与SQL



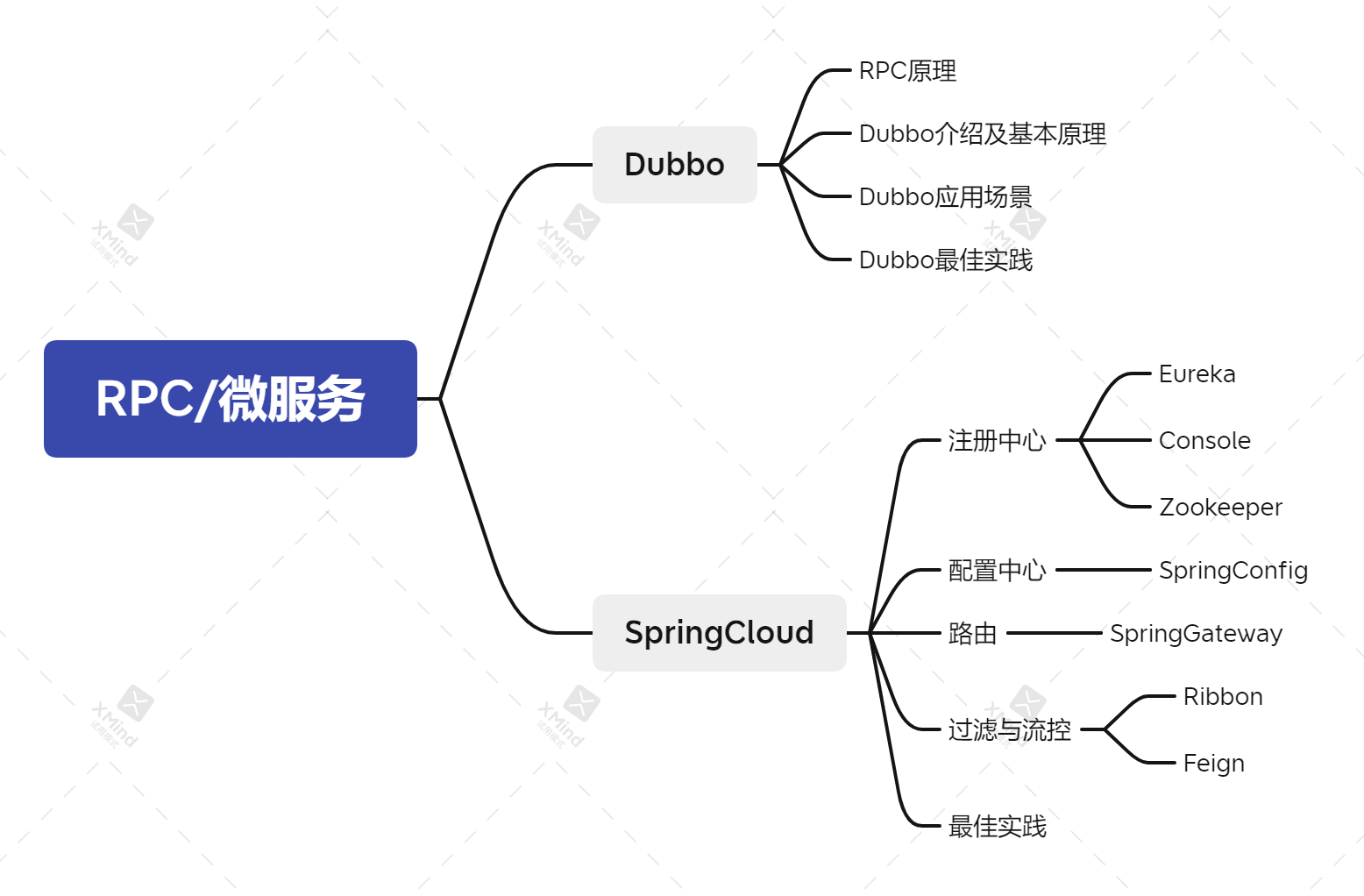
数据库是数据持久化的存储工具,MySql通过其开源免费,简单易用,轻量等特性拥有大量的用户,在分布式微服务大量普及与使用的情况下,轻量级的MySql得到了更大范围的使用,学习使用MySql是java开发必须掌握的,通过本节的学习,掌握了MySql的一些基本特性,了解了多种存储引擎的特点及使用场景,以及集群模式下的读写分离,主从复制及高可.MySql提供了多种内置函数,极大的简化了代码的编写,MySql支持sql92/99语法,良好的sql编写规范也是提升系统性能的重要手段,通过本节课的学习,掌握基本sql的写法以及MySql性能调优的几种方式,包括数据库设计优化,数据库配置优化,sql优化.

# 分库分表



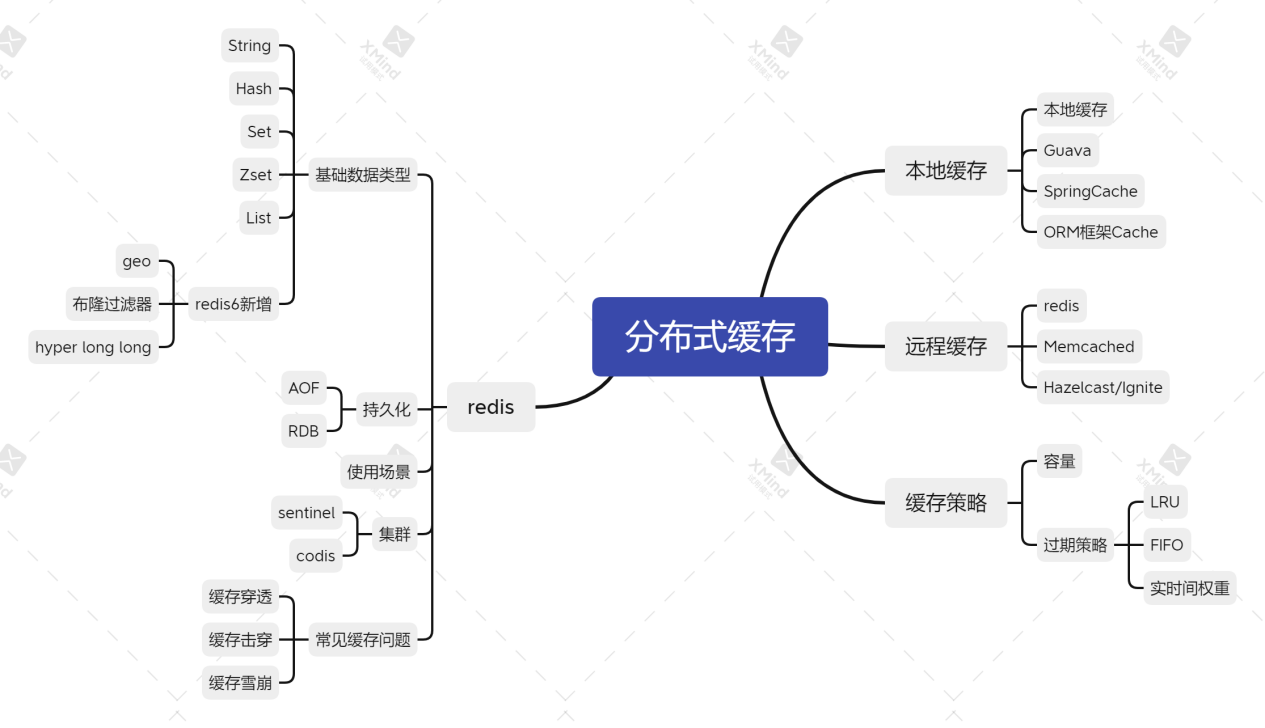
分库分表是在单机容量无法满足业务增长时进行扩容的方式,分库分表分为垂直分库与水平分库,采用水平分库时,集群以扩展,可以随着业务的变化动态伸缩,垂直分库通过业务维度将不同种类的数据拆分出来,在垂直分库的基础上也可以再进行水平分库,本节主要介绍了ShardingSphere框架,ShardingSphere提供了多种模式,包括jdbc,java程序通过引入依赖,配置文件就可以轻松完成分库分表的操作,也可以通过proxy代理模式代理多个数据库,而程序无需只需修改连接地址即可完成分库分表.在分库分表模式下,分布式事务就是首先要解决的问题,ShardingSphere提供了XA的事物的实现,humily提供了Tcc事务的实现,通过课后作业学习使用了两种框架,也理解了分布式事务不好处理的原因.

# RPC和微服务



本节课学了RPC与微服务,作为一个java初级开发人员,在最近的两年中微服务一直是最热门的技术也是比较难掌握的,微服务的体系庞大,应用场景也较为复杂,中小型系统一般不采用微服务架构,通过本节课的学习,不仅学习了解了RPC和SpringCloud技术体系的基本组成和原理,也学习到了不同框架的最佳实践,技术体系中的每一个组件都是值得深入学习和了解的.

# 分布式缓存



缓存是解决请求处理时间和IO瓶颈的重要手段,redis凭借高性能和内置的集群处理方式,多种数据类型,高效的持久化方式获得了广泛的使用,使用高效正确的使用redis是java开发必须具备的能力,同时也要评估本地缓存与远程缓存的开销,优先使用本地缓存,在使用缓存时也要考虑容量和实际的业务情况采用合理的缓存策略.使用redis时也要注意缓存穿透,缓存雪崩,缓存击穿等问题.

# 分布式消息队列



由于业务的快速增长,业务系统由单机演变为分布式系统,但是流量的高峰是无法预知的,消息队列可以在流量洪峰到达时将请求信息保存至消息队列,再由后端系统继续处理,达到削峰填谷的效果,消息队列的种类有很多,通过本节课的学习,学习掌握了最全面的额ActiveMQ,拥有优秀特性与集群处理方案的Kafka,阿里开源的RocketMQ,以及最近火热的ApachePlusar,每一种消息队列的本质是相同的,但是每一种消息队列都有最佳的实践场景.