

毕业作业

1.JVM

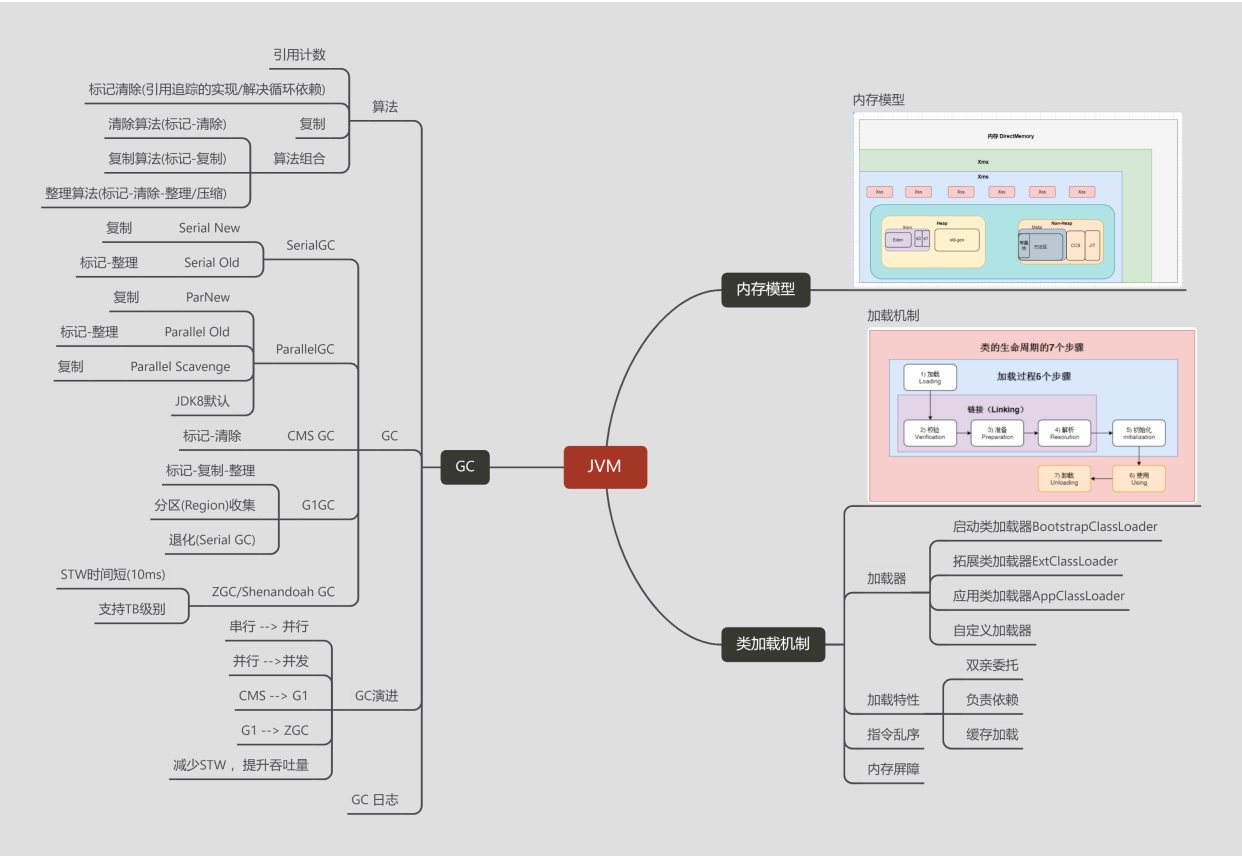
JVM是学好JAVA的基础。

了解了类加载机制，就可以知道类是如何加载的。指令重排序对代码执行顺序有什么影响。

了解了数据模型，可以知道不同类型的数据是如何存储的，如何定义可以线程安全。

了解了GC原理，可以知道怎么去调整GC算法，找到适用的收集器，减少STW，增加吞吐量。

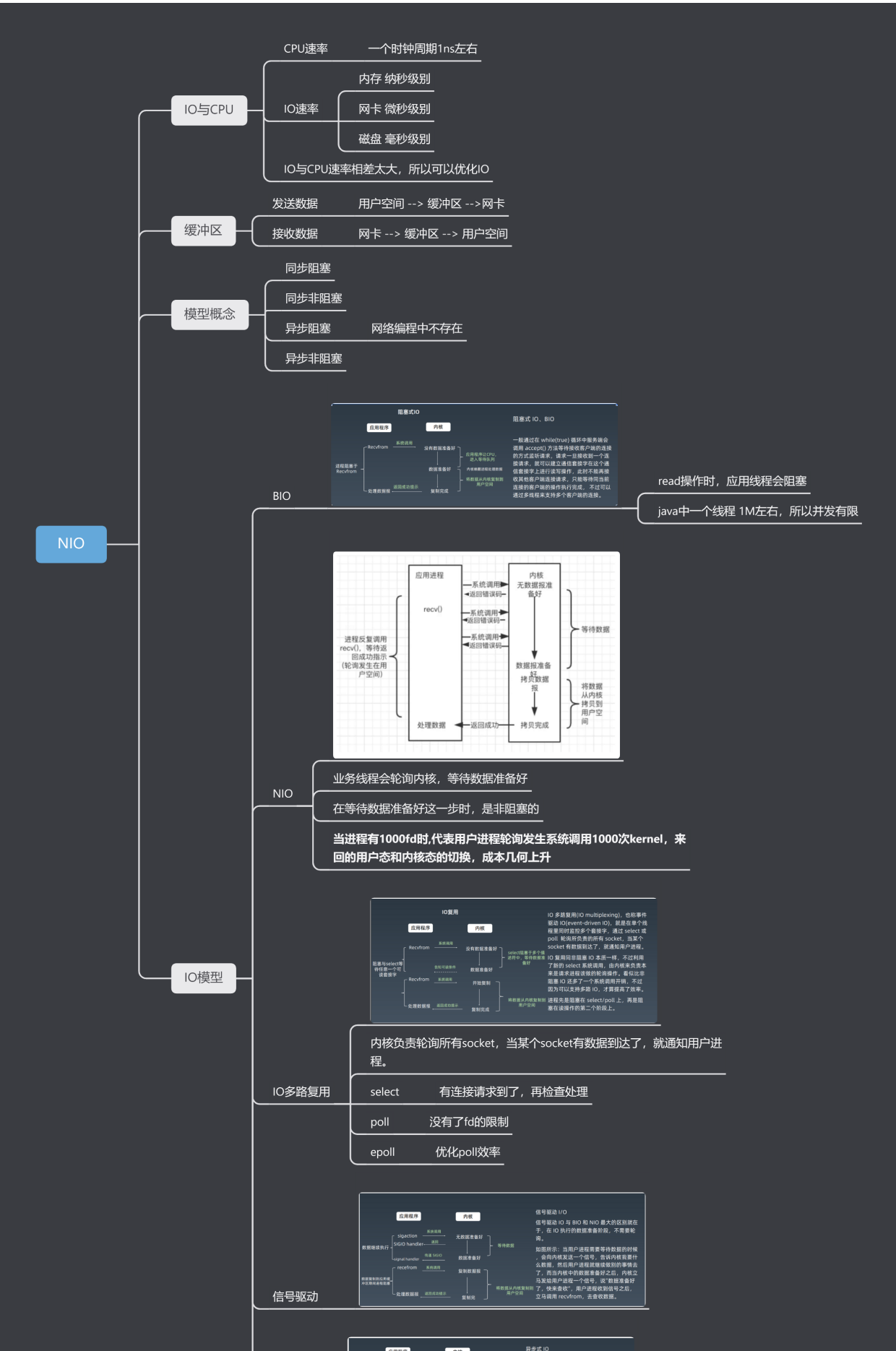
所以，不先了解底层的原理，在JAVA学习中，总会遇到这样那样的问题，理解得似是而非，从而记不住，学不精。

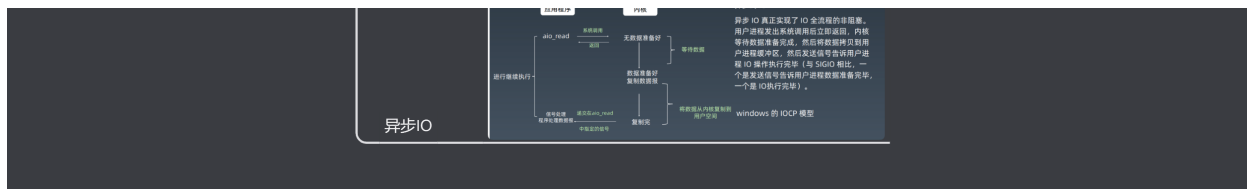


2.NIO

NIO，即非阻塞IO。相对于一般的IO方式(BIO),NIO不需要阻塞等待数据准备好，而是单独有线程轮询内核，等数据准备好了，再继续往下执行。其它时间可以去干其它事情，CPU不会在那傻傻等待。这样，使CPU得到了充分利用，不用耗费大量时钟周期在无用的等待上。

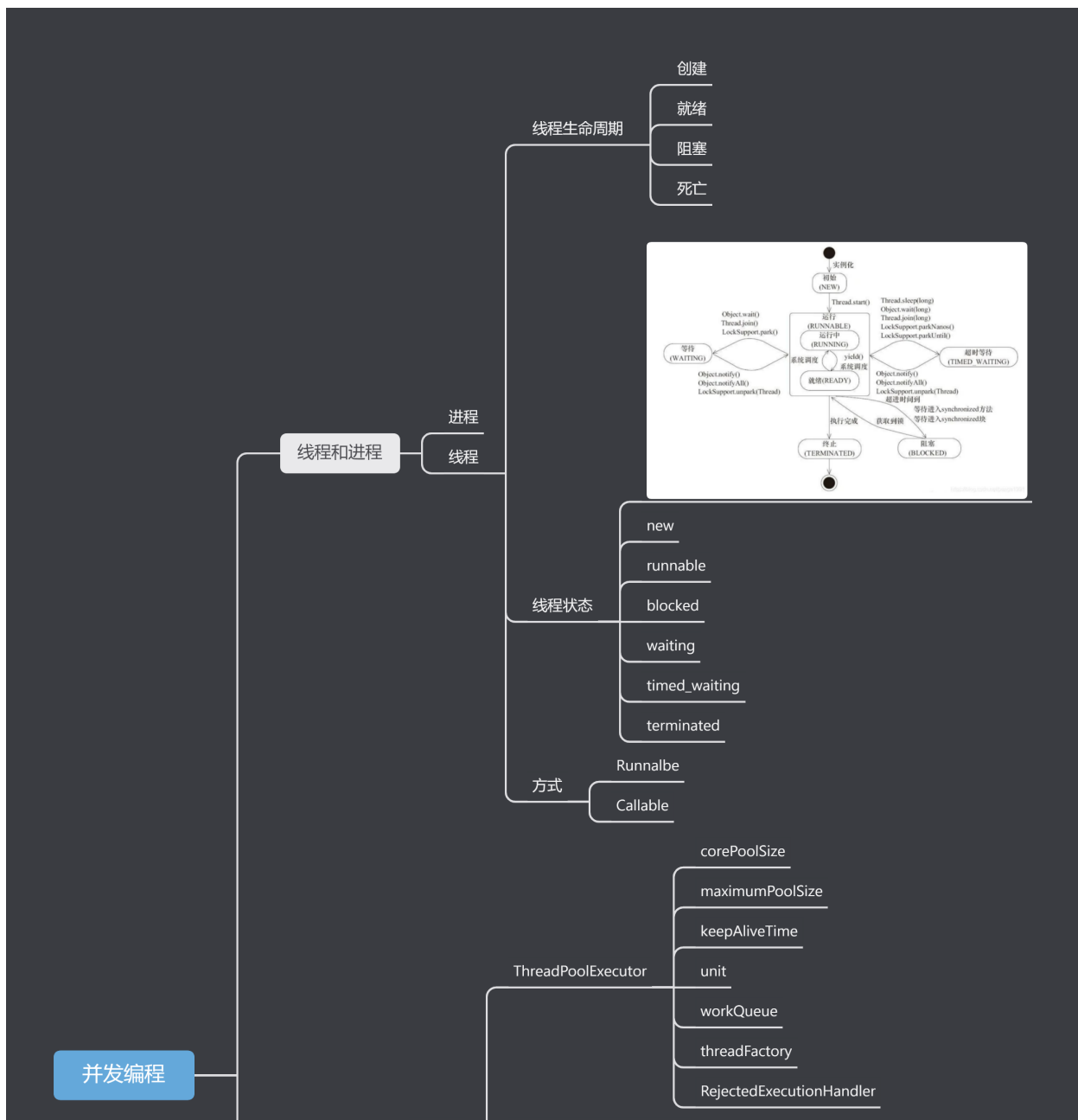
NIO的实现相对复制，如果不是为了研究底层源码，可以直接使用现有的封装好的框架。
Netty已成为NIO方面最知名的框架。

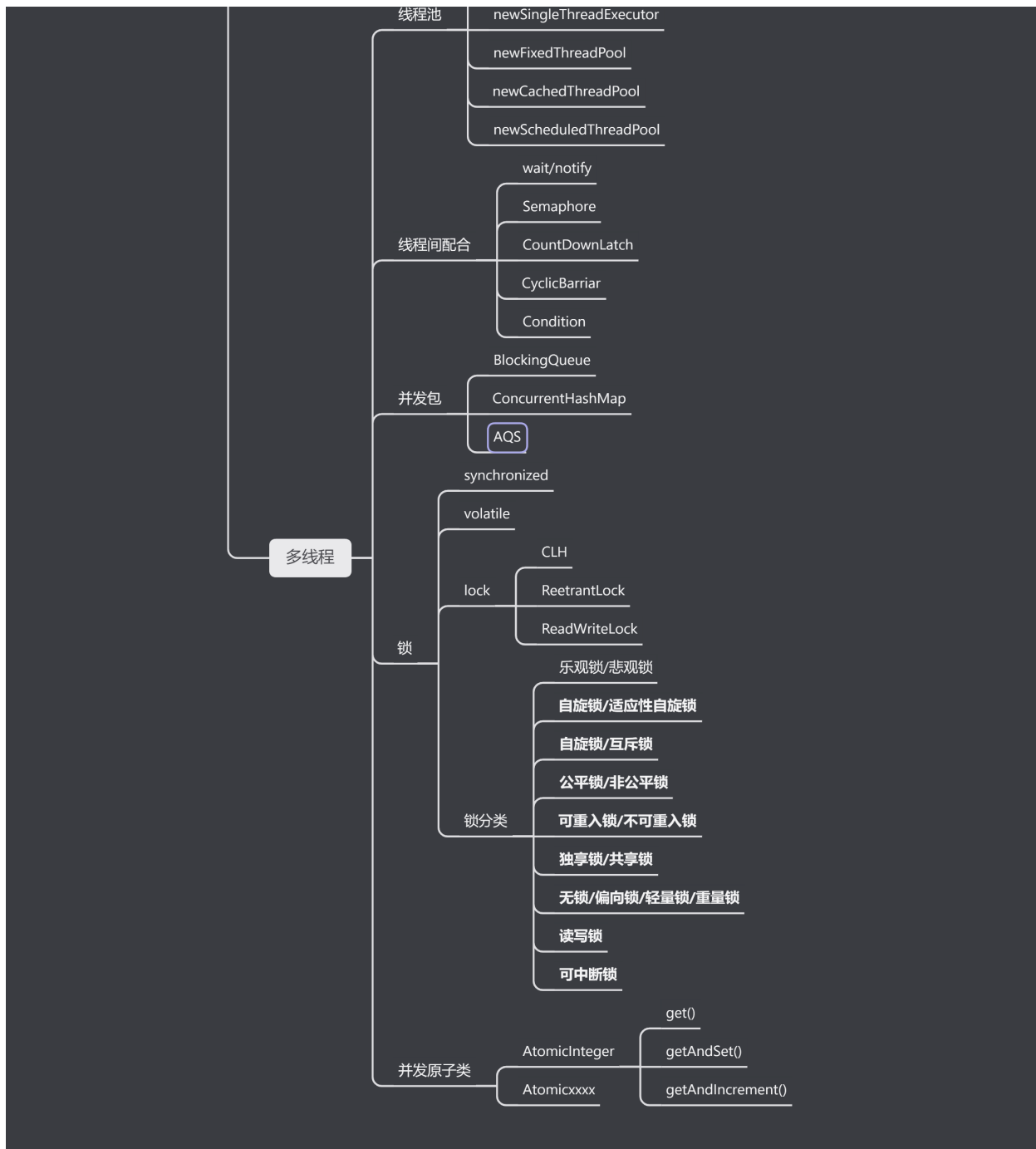




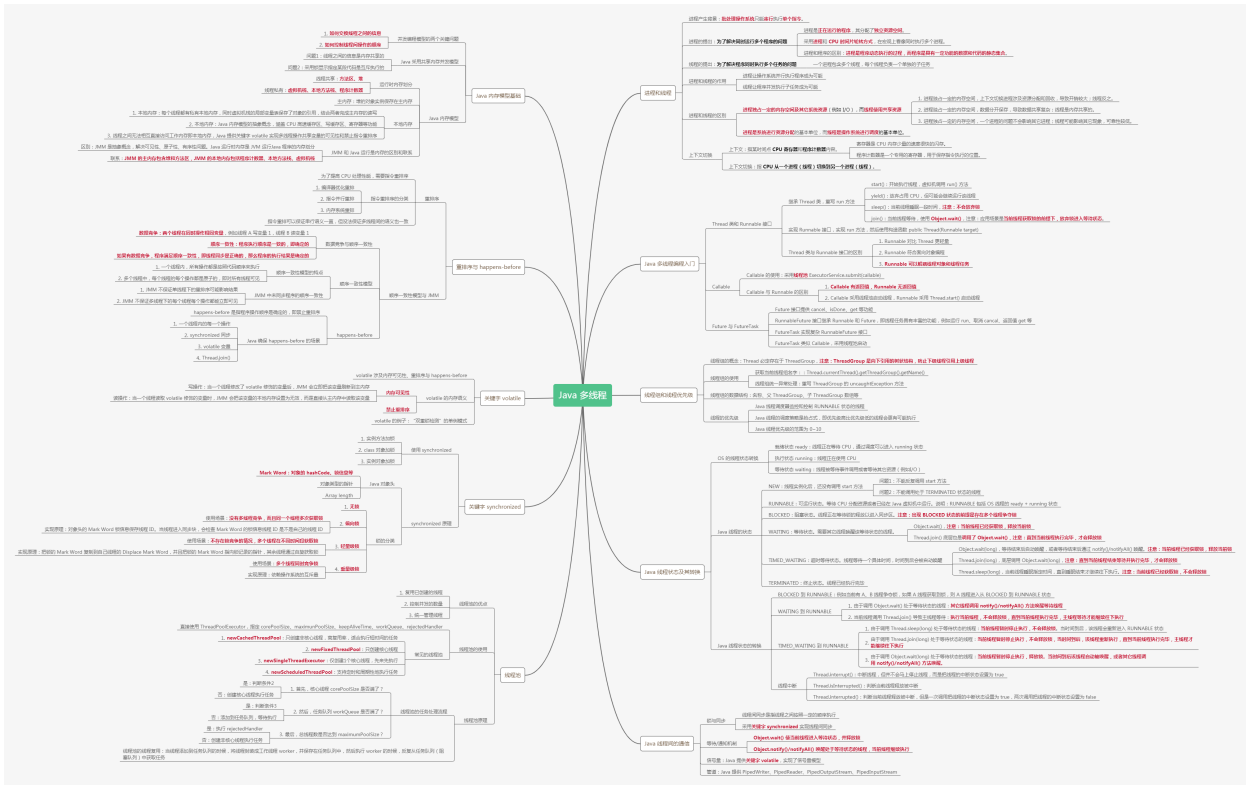
3.并发编程

一直以来，对多线程这一块都是零零散散的了解一点。东一块，西一块，从未认真去梳理过其中知识点。画了思维导图后，发现好像自己能串起来一部分了。然后网上看了人家画的思维导图，又觉得自己实在还有很多东西不知道。两张图一起放这里了，后面会按照第二张图，针对性的去学习补充知识，然后再将多线程知识点串通。





图片来源: <https://www.jianshu.com/p/79aeff70d9c0>



4.Spring 和 ORM 等框架

Spring已经是现在JavaWeb开发的事实标准。从我个人经验来看，一般工作中，只需要能管理好自己的bean, 不要产生冲突，不要产生循环依赖，不要注入失败就行了。因为工作内容大部分都是些简单的CURD,所以很少有需要去使用Spring的扩展点的时候。但是，AOP需要熟练掌握，很多功能都需要AOP来解耦代码，简化逻辑。

对于ORM框架，这里就只是举Mybatis。mybatis的SqlSession这个确是需要我们熟悉的，

某些时候，我们需要手动的切换mapper,就可以通过它来实现我们的逻辑。

spring和orm等框架

Spring

生命周期

bean的生命周期

- 实例化
- 属性赋值
- 初始化
- 销毁

各种扩展点

- aware
- postProcess

上下文

- 容器 beanFactory

循环依赖

- 三级缓存

核心原理

IOC

DI

AOP

常用功能

事务

- 隔离级别
- 传播机制

web

- 接口
- 参数校验

配置

- 配置类
- start
- environment

生态

Springboot

SpringCloud

Mybatis

缓存

- 一级缓存
- 二级缓存

MapperProxy

JDK动态代理

Executor

- simpleExecutor
- ReuseExecutor
- BatchExecutor
- CacheExecutor

SqlSession

SimpleStatementHandler

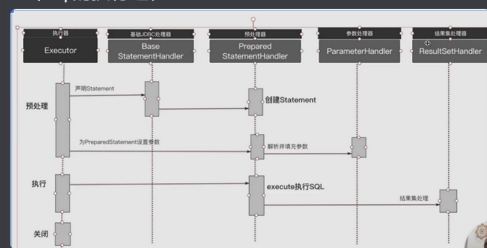
StatementHandler

PreparedStatementHandler

ResultSetHandler

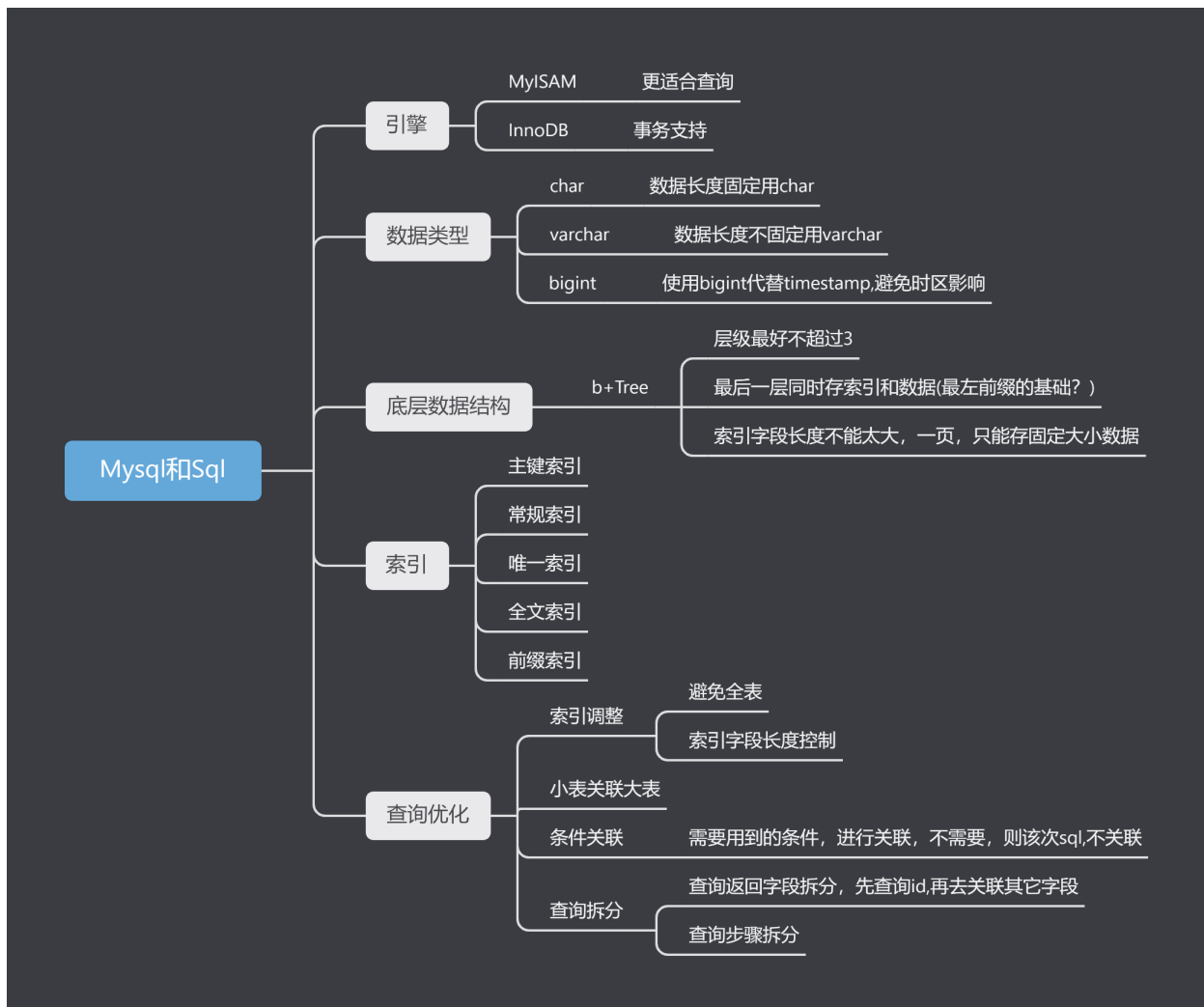
CallableStatementHandler(存储过程)

一个sql的执行过程



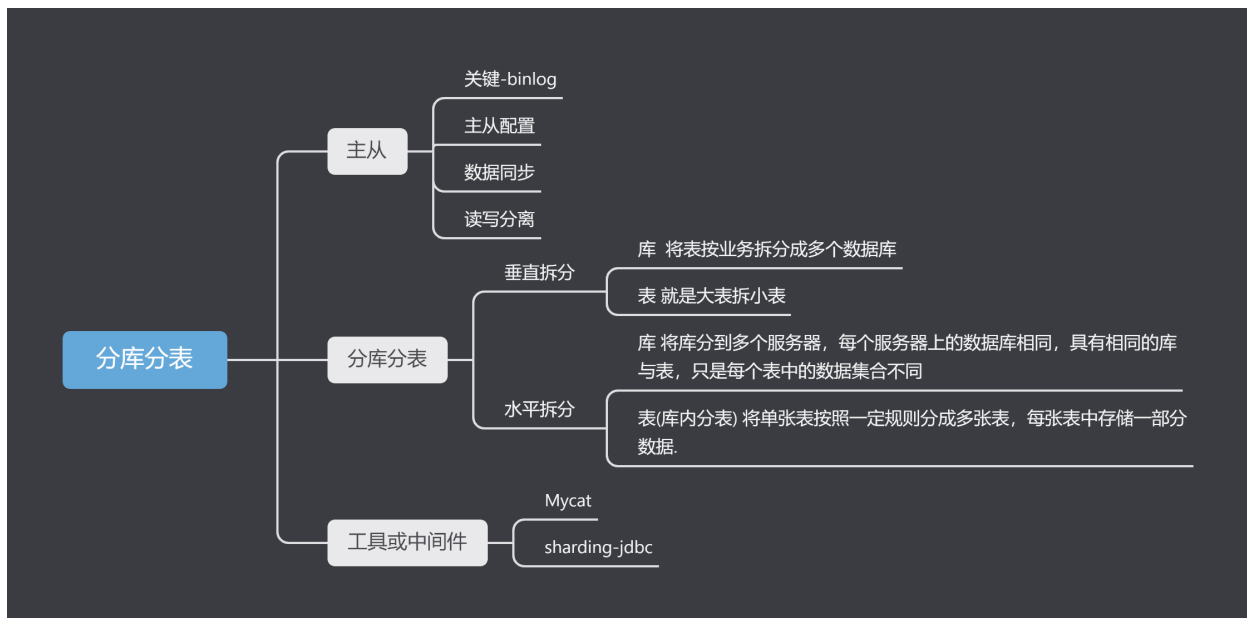
5.MySQL 数据库和 SQL

我工作中接触的都是运营业务。现在运营商的都在去Oracle, 转而通常使用Mysql或者PG, 禁止使用存储过程等(现在的趋势)。所以这里没有列存储过程等平时不常用的。根据我的实际经验, 一般Mysql的使用还是相对简单的, 在几百万数据的时候, 一般定义好数据类型, 构建好合适的索引就行。实在有复杂查询, 则针对性的去优化, 自己手动实现分页, 符合索引规则等。千万级的数据操作, 则再加上分库分表。数据量特别大的查询, 现在一般也不会用Mysql去查了, 而是转到ES或者Hbase等。



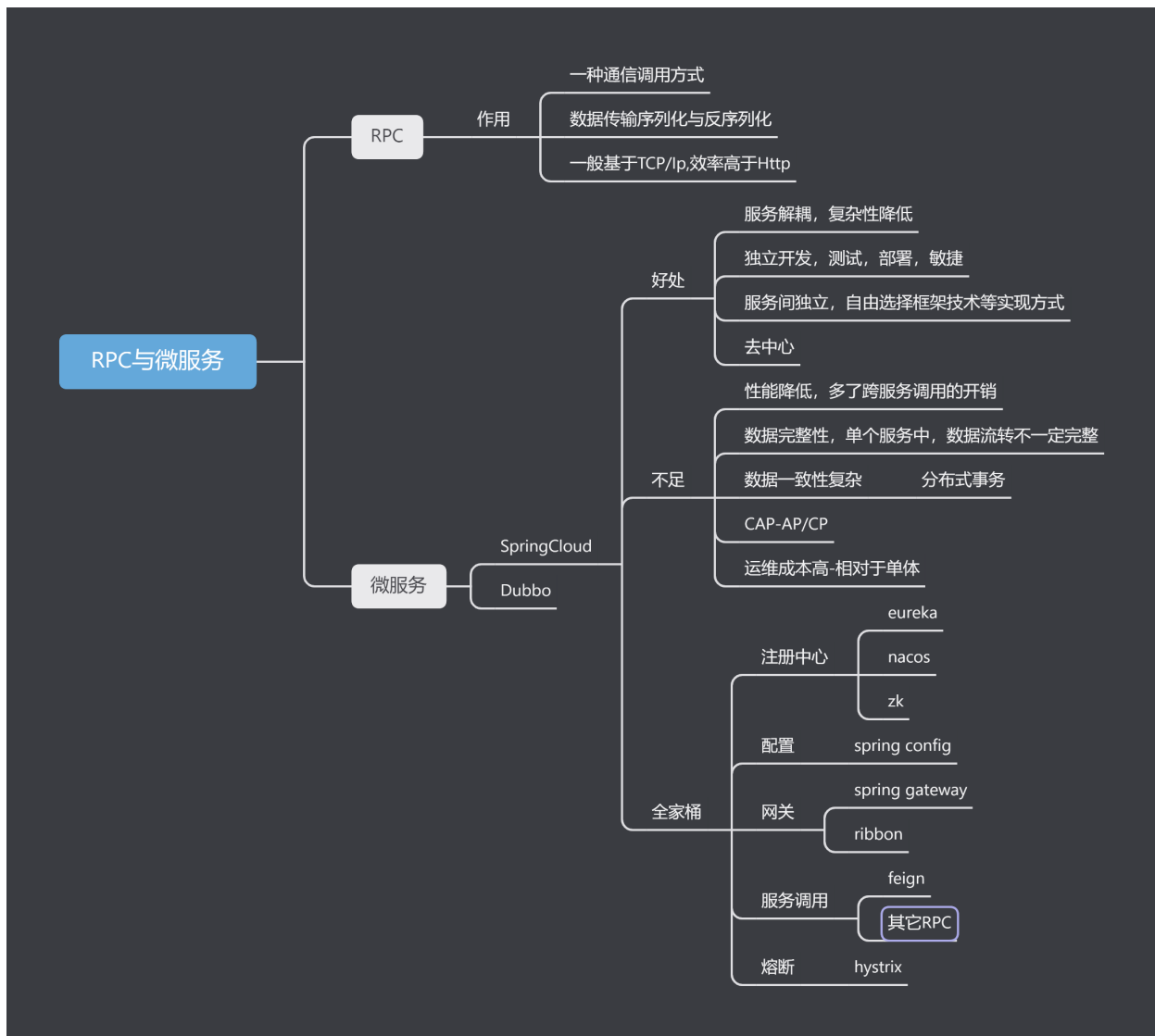
6.分库分表

分库分表, 我只在自己电脑上尝试过, 实际工作中, 并没有去使用过。后面有机会, 再把它进行完善。



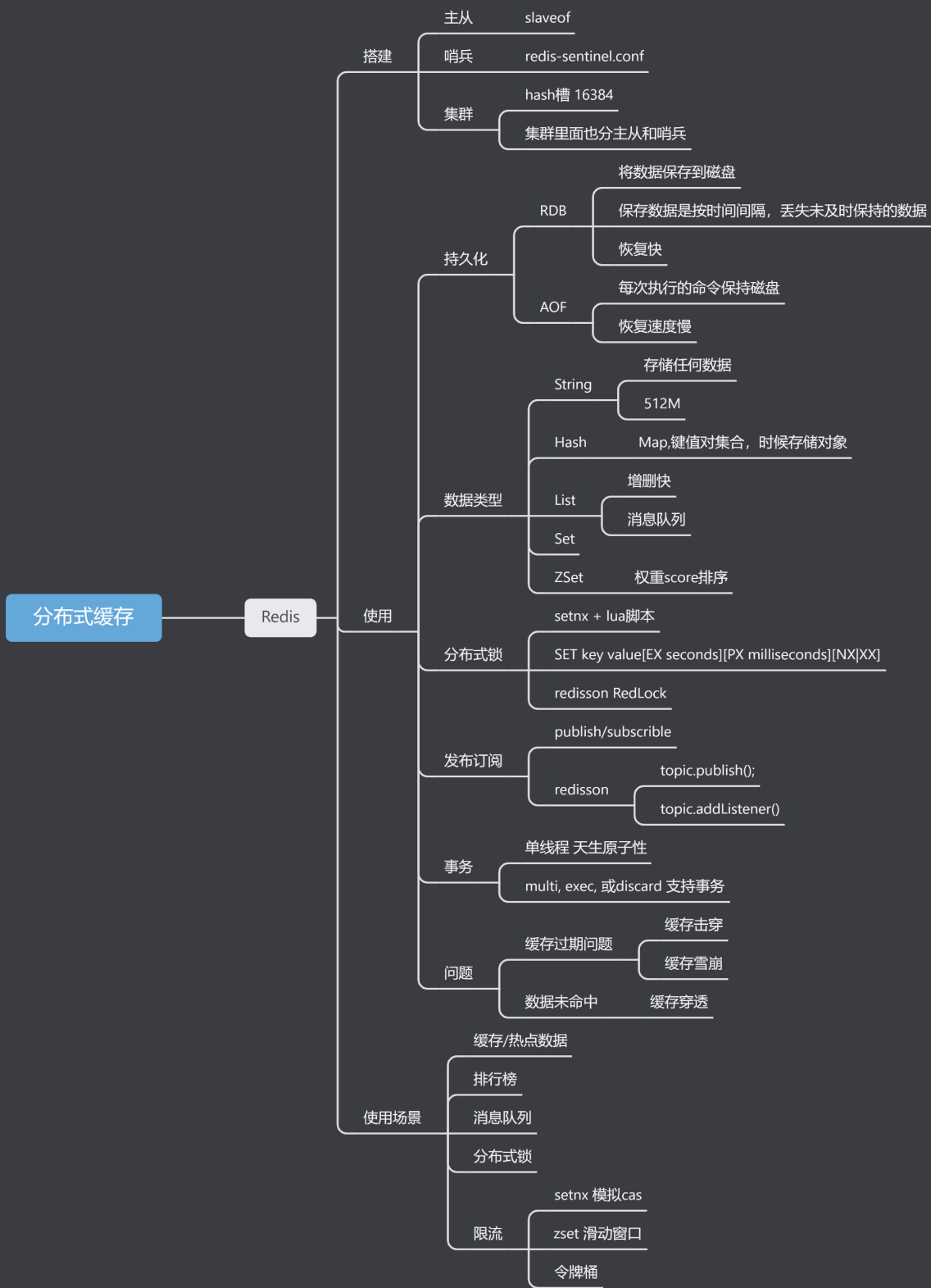
7.RPC 和微服务

微服务化是当前复制项目的主流趋势。目前正在做一个项目，就是spring cloud，之前都只是了解，未实践。等这个项目完成，结合实际开发中的经验积累，再来补充这一部分内容。



8.分布式缓存

redis是目前使用最多的缓存数据库。基于内存，速度快，用来缓存热点数据。单线程，天生原子操作，实现分布式锁。同时可以实现简单的消息队列和发布订阅。使用redis,要会搭建高可用的redis集群，配置主从和哨兵，合理分配所有hash槽，同时根据业务需要选择合适的数据类型。在使用缓存时，要合理的设置缓存的过期时间，避免发生击穿和雪崩等。



9.分布式消息队列

消息队列使用不多，只简单使用过kafka,后面补上。

