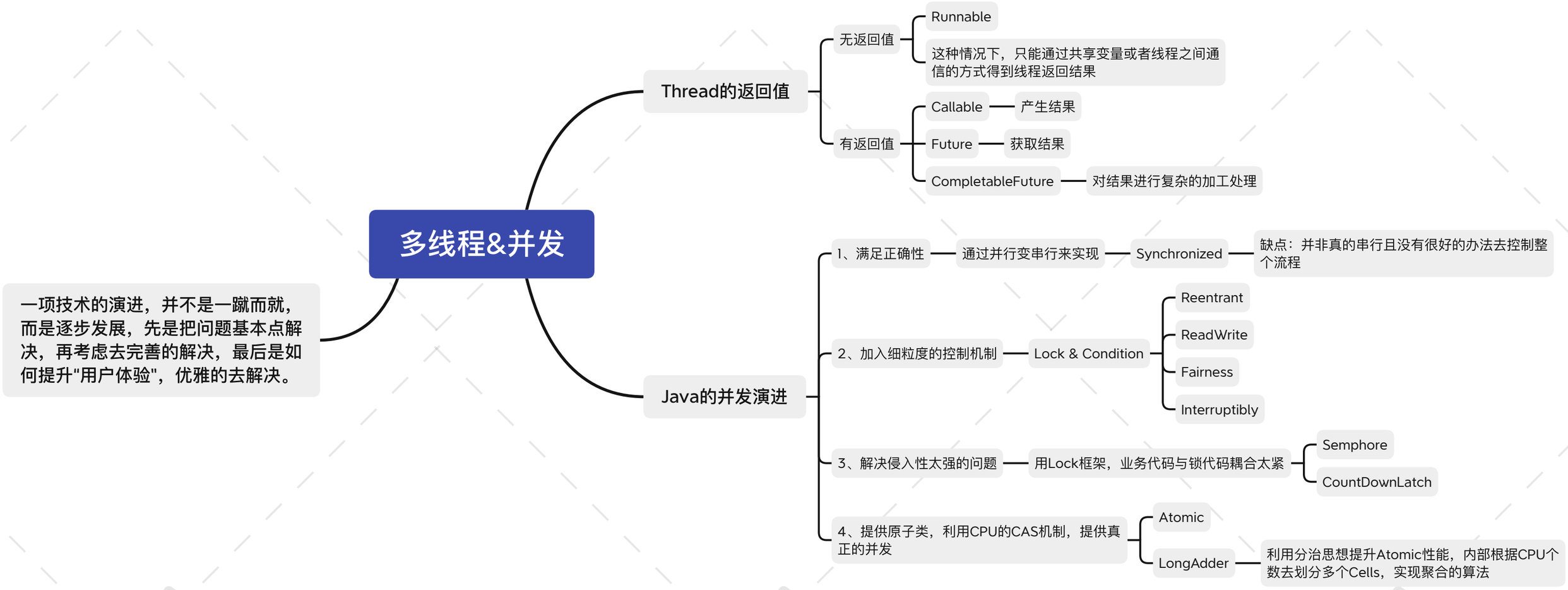
多线程&并发
一项技术的演进，并不是一蹴而就，而是逐步发展，先是把问题基本点解决，再考虑去完善的解决，最后是如何提升"用户体验"，优雅的去解决。
Thread的返回值
无返回值
Runnable
这种情况下，只能通过共享变量或者线程之间通信的方式得到线程返回结果
有返回值
Callable
产生结果
Future
获取结果
CompletableFuture
对结果进行复杂的加工处理
Java的并发演进
1、满足正确性
通过并行变串行来实现
Synchronized
缺点：并非真的串行且没有很好的办法去控制整个流程
2、加入细粒度的控制机制
Lock & Condition
Reentrant
ReadWrite
Fairness
Interruptibly
3、解决侵入性太强的问题
用Lock框架，业务代码与锁代码耦合太紧
Semphore
CountDownLatch
4、提供原子类，利用CPU的CAS机制，提供真正的并发
Atomic
LongAdder
利用分治思想提升Atomic性能，内部根据CPU个数去划分多个Cells，实现聚合的算法