1、线程池 构造方法 相关不同类型线程池区别？

https://www.processon.com/mindmap/5ff16df76376896cfa00426c

2、设计模式：策略模式、工厂模式

3、List<String> 添加 Integer 反射

4、sql 事务的基本要素（ACID）

事物隔离级别

read uncommited ：读到未提交数据

read committed：脏读，不可重复读

repeatable read：可重读

serializable ：串行事物

5、分布式事务 TCC

6、zookeeper 为什么能做注册中心？

- **zookeeper提供了“心跳检测”功能**，它会定时向各个服务提供者发送一个请求（实际上建立的是一个 socket 长连接），如果长期没有响应，服务中心就认为该服务提供者已经“挂了”，并将其剔除。比如100.100.0.237这台机器如果宕机了，那么zookeeper上的路径就会只剩/HelloWorldService/1.0.0/100.100.0.238:16888。
- **服务消费者会去监听相应路径（/HelloWorldService/1.0.0）**，一旦路径上的数据有任务变化（增加或减少），zookeeper都会通知服务消费方、服务提供者地址列表已经发生改变，从而进行更新。
- **更为重要的是zookeeper 与生俱来的容错容灾能力（比如leader选举）**，可以确保服务注册表的高可用性。
- **使用 zookeeper 作为注册中心时，客户端订阅服务时会向 zookeeper 注册自身**；主要是方便对调用方进行统计、管理。但订阅时是否注册 client 不是必要行为，和不同的注册中心实现有关，例如使用 consul 时便没有注册。

7、volatile i 扔存在多线程安全性问题 Java中的运算并非是原子操作，所以导致volatile声明的变量无法保证线程安全。

8、dubbo 与http 连接请求优缺点

参考https://blog.csdn.net/yyx3214/article/details/103326224

socket连接

dubbo默认使用socket长连接，即首次访问建立连接以后，后续网络请求使用相同的网络通道,这样避免了多次重复创建TCP连接的开销

http1.1协议默认使用短连接，每次请求均需要进行三次握手，而http2.0协议开始将默认socket连接改为了长连接

http协议有一个特效，会有一系列的http header，这些内容往往会占用几k的数据，访问量非常大的时候，这些数据也是一种负担，而使用dubbo，协议可以自定义，这些无关数据可

以省掉。

封装

简单来说成熟的rpc库相对http容器，更多的是封装了“服务发现”，"错误重试，负载均衡"一类面向服务的高级特性。可以这么理解，rpc框架是面向服务的更高级的封装。如果把一个http server容器上封装一层服务发现和函数代理调用，那它就已经可以做一个rpc框架了。

安全

dubbo设计之初基本都是考虑内网通讯，安全上基本没什么考虑，比http的安全差了很远

9、dubbo spi机制 与java原生区别

service provider interface

jdk mysql jdbc

1. JDK 标准的 SPI 会一次性加载实例化扩展点的所有实现，什么意思呢？就是如果你在 META-INF/service 下的文件里面加了 N 个实现类，那么 JDK 启动的时候都会一次性全部加载。那么如果有的扩展点实现初始化很耗时或者如果有些实现类并没有用到， 那么会很浪费资源
2. 如果扩展点加载失败，会导致调用方报错，而且这个错误很难定位到是这个原因

10、限流方式

dubbo 服务、方法 配置

dubbo 默认实现的是滑动窗口的方式

计数器、滑动窗口、漏斗和令牌桶算法