ActiveMQ“

这个简单，ActiveMQ 是一个 MOM，具体来说是一个实现了 JMS 规范的系统间远程通信的消息代理。好。MOM 就是面向消息中间件（Message-oriented middleware），是用于以分布式应用或系统中的异步、松耦合、可靠、可扩展和安全通信的一类软件。MOM 的总体思想是它作为消息发送器和消息接收器之间的消息中介,这种中介提供了一个全新水平的松耦合。

Active

* 获取连接工厂
* 使用连接工厂创建连接
* 启动连接
* 从连接创建会话
* 获取 Destination
* 创建 Producer，或
  + 创建 Producer
  + 创建 message
* 创建 Consumer，或发送或接收message发送或接收 message
  + 创建 Consumer
  + 注册消息监听器（可选）
* 发送或接收 message
* 关闭资源（connection, session, producer, consumer 等)

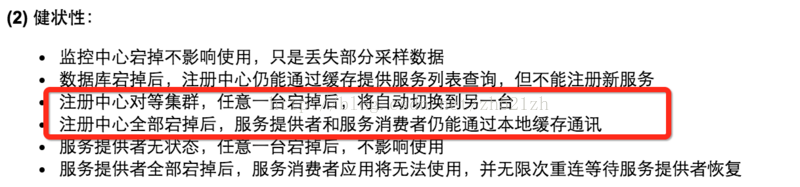
# 面试题：Dubbo中zookeeper做注册中心，如果注册中心集群全都挂掉，发布者和订阅者之间还能通信么？

可以的，消费者在启动时，消费者会从zk拉取注册的生产者的地址接口等数据，缓存在本地。

每次调用时，按照本地存储的地址进行调用

可以的，消费者本地有一个生产者的列表，他会按照列表继续工作，倒是无法从注册中心去同步最新的服务列表，短期的注册中心挂掉是不要紧的，但一定要尽快修复

挂掉是不要紧的，但前提是你没有增加新的服务，如果你要调用新的服务，则是不能办到的



Redis的雪崩和击穿

缓存穿透是指查询一个一定不存在的数据，由于缓存是不命中时需要从数据库查询，查不到数据则不写入缓存，这将导致这个不存在的数据每次请求都要到数据库去查询，造成缓存穿透。

* 解决办法：

1. 对所有可能查询的参数以hash形式存储，在控制层先进行校验，不符合则丢弃。还有最常见的则是采用布隆过滤器，将所有可能存在的数据哈希到一个足够大的bitmap中，一个一定不存在的数据会被这个bitmap拦截掉，从而避免了对底层存储系统的查询压力。
2. 也可以采用一个更为简单粗暴的方法，如果一个查询返回的数据为空（不管是数 据不存在，还是系统故障），我们仍然把这个空结果进行缓存，但它的过期时间会很短，最长不超过五分钟。

### 缓存雪崩

* 认识

如果缓存集中在一段时间内失效，发生大量的缓存穿透，所有的查询都落在数据库上，造成了缓存雪崩。

这个没有完美解决办法，但可以分析用户行为，尽量让失效时间点均匀分布。大多数系统设计者考虑用加锁或者队列的方式保证缓存的单线程（进程）写，从而避免失效时大量的并发请求落到底层存储系统上。

* 解决方法

1. 在缓存失效后，通过加锁或者队列来控制读数据库写缓存的线程数量。比如对某个key只允许一个线程查询数据和写缓存，其他线程等待。
2. 可以通过缓存reload机制，预先去更新缓存，再即将发生大并发访问前手动触发加载缓存
3. 不同的key，设置不同的过期时间，让缓存失效的时间点尽量均匀
4. 做二级缓存，或者双缓存策略。A1为原始缓存，A2为拷贝缓存，A1失效时，可以访问A2，A1缓存失效时间设置为短期，A2设置为长期。

# 1 bean配置

bean配置有三种方法：

基于xml配置Bean

使用注解定义Bean

基于java类提供Bean定义信息

## 1.1 基于xml配置Bean

 对于基于XML的配置，Spring 2.0以后使用Schema的格式，使得不同类型的配置拥有了自己的命名空间，是配置文件更具扩展性。

①默认命名空间：它没有空间名，用于Spring Bean的定义；

②xsi命名空间：这个命名空间用于为每个文档中命名空间指定相应的Schema样式文件，是标准组织定义的标准命名空间；

③aop命名空间：这个命名空间是Spring配置AOP的命名空间，是用户自定义的命名空间。

Bean的基本配置

一般情况下，Spring IOC容器中的一个Bean即对应配置文件中的一个<bean>,这种镜像对应关系应该容易理解。其中id为这个Bean的名称，通过容器的getBean("foo")即可获取对应的Bean，在容器中起到定位查找的作用，是外部程序和Spring IOC容器进行交互的桥梁。class属性指定了Bean对应的实现类

# 数据库的几种索引

(1)按照索引列值的唯一性，索引可分为唯一索引和非唯一索引

①非唯一索引：B树索引

create index 索引名 on 表名（列名） tablespace 表空间名;

②唯一索引：建立主键或者唯一约束时会自动在对应的列上建立唯一索引

(2)索引列的个数：单列索引和复合索引

(3)按照索引列的物理组织方式

①B树索引

create index 索引名 on 表名（列名） tablespace 表空间名;

②位图索引

create bitmap index 索引名 on 表名（列名） tablespace 表空间名;

③反向键索引

create index 索引名 on 表名（列名） reverse tablespace 表空间名;

④函数索引

create index 索引名 on 表名（函数名(列名)） tablespace 表空间名;

删除索引

drop index 索引名

重建索引

alter index 索引名 rebuild

### 工厂模式

工厂模式是我们最常用的实例化对象模式了，是用工厂方法代替new操作的一种模式。著名的Jive论坛 ,就大量使用了工厂模式，工厂模式在Java程序系统可以说是随处可见。因为工厂模式就相当于创建实例对象的new，我们经常要根据类Class生成实例对象，如A a=new A() 工厂模式也是用来创建实例对象的，所以以后new时就要多个心眼，是否可以考虑使用工厂模式，虽然这样做，可能多做一些工作，但会给你系统带来更大的可扩展性和尽量少的修改量。