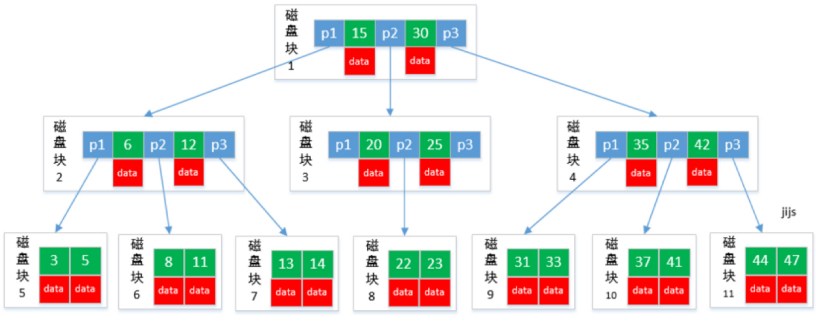
> 腾讯-0305

1. 从简历项目中选一个项目，说说你在其中遇到了什么重大挑战？以及你的解决问题的思路？
2. 一段代码要执行多个redis命令，不加锁的情况下如何保证原子性？

   使用lua脚本：https://segmentfault.com/a/1190000009811453
3. 谈谈数据结构，比如二叉树、红黑树？

   理解这篇：https://juejin.im/post/5a27c6946fb9a04509096248
4. 说说B-tree、B+tree的区别和使用场景？

   1. B-tree：

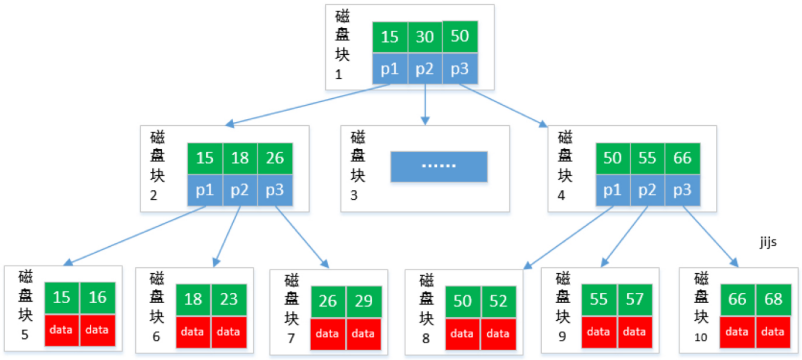


   B-tree 利用了磁盘块的特性进行构建的树。每个磁盘块一个节点，每个节点包含了很关键字。把树的节点关键字增多后树的层级比原来的二叉树少了，减少数据查找的次数和复杂度。

   B-tree巧妙利用了磁盘预读原理，将一个节点的大小设为等于一个页（每页为4K），这样每个节点只需要一次I/O就可以完全载入。

   B-tree 的数据可以存在任何节点中。

   2. B+tree：



   B+tree 是 B-tree 的变种，B+tree 数据只存储在叶子节点中。这样在B树的基础上每个节点存储的关键字数更多，树的层级更少所以查询数据更快，所有指关键字指针都存在叶子节点，所以每次查找的次数都相同所以查询速度更稳定；
5. mysql哪个版本哪个存储引擎的索引使用的B+tree，为什么不使用红黑树？

   需要先理解B+tree、红黑树的实现原理。B+tree带有顺序访问指针，是红黑树不具备的。
6. 说说几种常见的消息中间件的区别？

特性	ActiveMQ	RabbitMQ	RocketMQ	kafka
开发语言	java	erlang	java	scala
单机吞吐量	万级	万级	10万级	10万级
时效性	ms级	us级	ms级	ms级以内
可用性	高(主从架构)	高(主从架构)	非常高(分布式架构)	非常高(分布式架构)
功能特性	成熟的产品，在很多公司得到应用；有较多的文档；各种协议支持较好	基于erlang开发，所以并发能力很强，性能极其好，延时很低;管理界面较丰富	MQ功能比较完备，扩展性佳	只支持主要的MQ功能，像一些消息查询，消息回溯等功能没有提供，毕竟是为大数据准备的，在大数据领域应用广。

1. 中小型公司首选RabbitMQ：管理界面简单，高并发。
2. 大型公司可以选择RocketMQ：更高并发，可对rocketmq进行定制化开发。
3. 日志采集功能，首选kafka，专为大数据准备。

7. rabbitmq如何保证消息的可靠性?

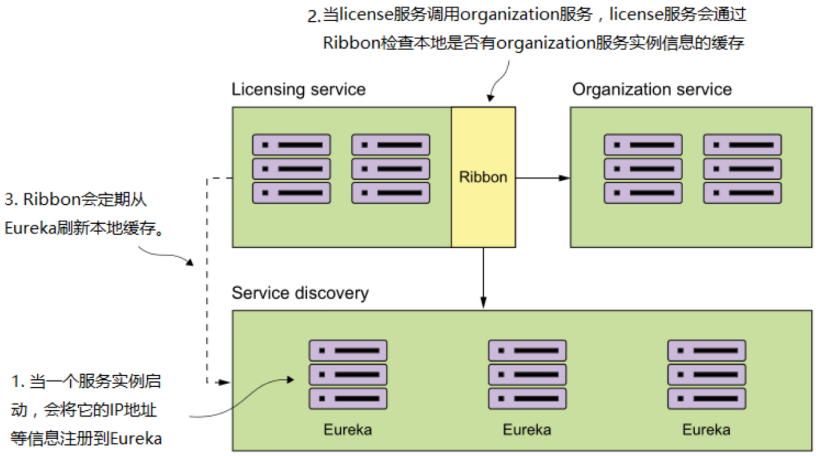
   详见“面试题库/rabbitmq”

8. springcloud服务发现原理?

   a. 每30s发送心跳检测重新进行租约，如果客户端不能多次更新租约，它将在90s内从服务器注册中心移除。

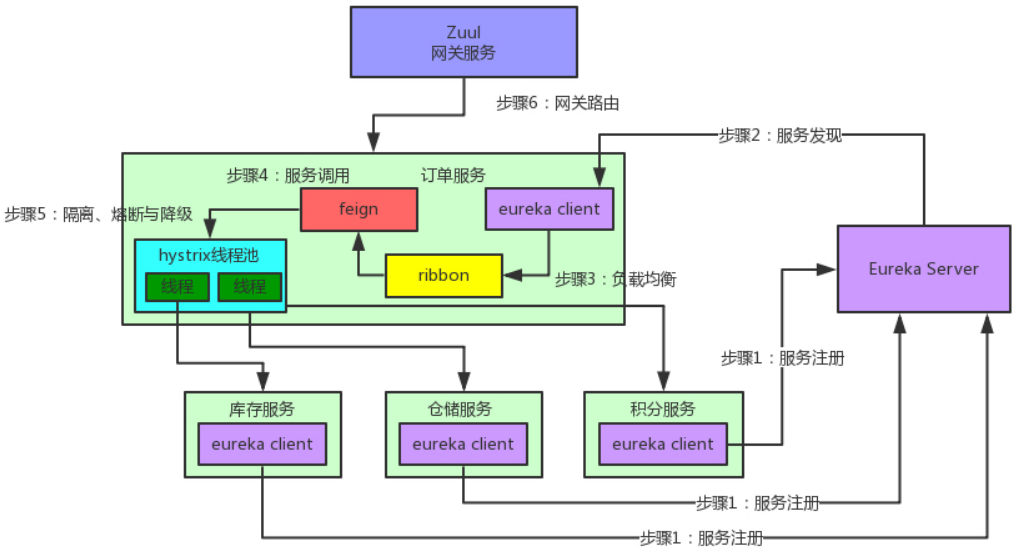
   a. 注册信息和更新会被复制到其他Eureka 节点，来自任何区域的客户端可以查找到注册中心信息，每30s发生一次复制来定位他们的服务，并进行远程调用。

   b. 客户端还可以缓存一些服务实例信息，所以即使Eureka全挂掉，客户端也是可以定位到服务地址的。



9. 介绍下springcloud各个组件？springcloud的注册中心除了eureka还可以用什么？

   springcloud的工作原理



springcloud由以下几个核心组件构成：

Eureka：各个服务启动时，Eureka Client都会将服务注册到Eureka Server，并且Eureka Client还可以反过来从Eureka Server拉取注册表，从而知道其他服务在哪里

Ribbon：服务间发起请求的时候，基于Ribbon做负载均衡，从一个服务的多台机器中选择一台

Feign：基于Feign的动态代理机制，根据注解和选择的机器，拼接请求URL地址，发起请求

Hystrix：发起请求是通过Hystrix的线程池来走的，不同的服务走不同的线程池，实现了不同服务调用的隔离，避免了服务雪崩的问题

Zuul：如果前端、移动端要调用后端系统，统一从Zuul网关进入，由Zuul网关转发请求给对应的服务

注册中心还可以用zookeeper。

10. 微服务有几种限流方式？

    spring cloud gateway：https://windmt.com/2018/05/09/spring-cloud-15-spring-cloud-gateway-ratelimiter/

11. 限流的情况下，服务隔离还有没有必要？

    https://www.javazhiyin.com/25964.html

12. dubbo有几种负载均衡？负载均衡是在服务端还是客户端？

    Dubbo负载均衡在客户端，dubbo内置了4种负载均衡策略：

    a. RandomLoadBalance:随机负载均衡。随机的选择一个。是Dubbo的默认负载均衡策略。

    b. RoundRobinLoadBalance:轮询负载均衡。轮询选择一个。

    c. LeastActiveLoadBalance:最少活跃调用数，相同活跃数的随机。活跃数指调用前后计数差。使慢的 Provider 收到更少请求，因为越慢的 Provider 的调用前后计数差会越大。

    d. ConsistentHashLoadBalance:一致性哈希负载均衡。相同参数的请求总是落在同一台机器上。

13. 如何实现redis分布式锁？需要注意什么问题？

    了解这篇：https://juejin.im/post/5bbb0d8df265da0abd3533a5

14. 说说你看过的源码？其中用到了什么设计模式或者设计亮点？

    具体分析，面试前需要熟读一些源码，如spring源码。

15. 如何实现aop？项目中哪些地方用到了aop？

    掌握：https://juejin.im/post/5bf4fc84f265da611b57f906

16. 后置处理器的作用？

    Spring中bean后置处理器BeanPostProcessor：https://www.jianshu.com/p/f80b77d65d39

17. spring bean作用域，什么时候使用request作用域。

作用域	描述
singleton	该作用域将 bean 的定义的限制在每一个 Spring IoC 容器中的一个单一实例(默认)。
prototype	该作用域将单一 bean 的定义限制在任意数量的对象实例。
request	该作用域将 bean 的定义限制为 HTTP 请求。只在 web-aware Spring ApplicationContext 的上下文中有效。
session	该作用域将 bean 的定义限制为 HTTP 会话。 只在web-aware Spring ApplicationContext的上下文中有效。
global-session	该作用域将 bean 的定义限制为全局 HTTP 会话。只在 web-aware Spring ApplicationContext 的上下文中有效。

详读：https://blog.csdn.net/icarus_wang/article/details/51586776

18. 说说下面这道题的结果?

```
package com.giveu.web;

public class VolatileTest {
   public static volatile int race = 0;

   public static void increase() {
      race++;
   }

    private static final int THREADS_COUNT = 10;

    public static void main(String[] args) {
       Thread[] threads = new Thread[THREADS_COUNT];
       for (int i = 0; i < THREADS_COUNT; i++) {
          threads[i] = new Thread(new Runnable() {
             @Override
             public void run() {
                for (int i = 0; i < 10000; i++) {
                   increase();
                }
             }
          });
          threads[i].start();
       }
       while (Thread.activeCount() > 1) {
          Thread.yield();
       }
       System.out.println(race);
    }
}
```

程序不结束，并且没有打印。