

😊 缓存对比

📖 公众号：java思维导图

简介

redis



属于独立的运行程序，需要单独安装后，使用Java中的Jedis来操纵。因为它是独立，所以如果你写个单元测试程序，放一些数据在redis中，然后又写一个程序去拿数据，那么是可以拿到这个数据的。

ehcache



与Redis明显不同，它与java程序是绑在一起的，java程序活着，它就活着。譬如，写一个独立程序放数据，再写一个独立程序拿数据，那么是拿不到数据的。只能在独立程序中才能拿到数据。

集群

redis

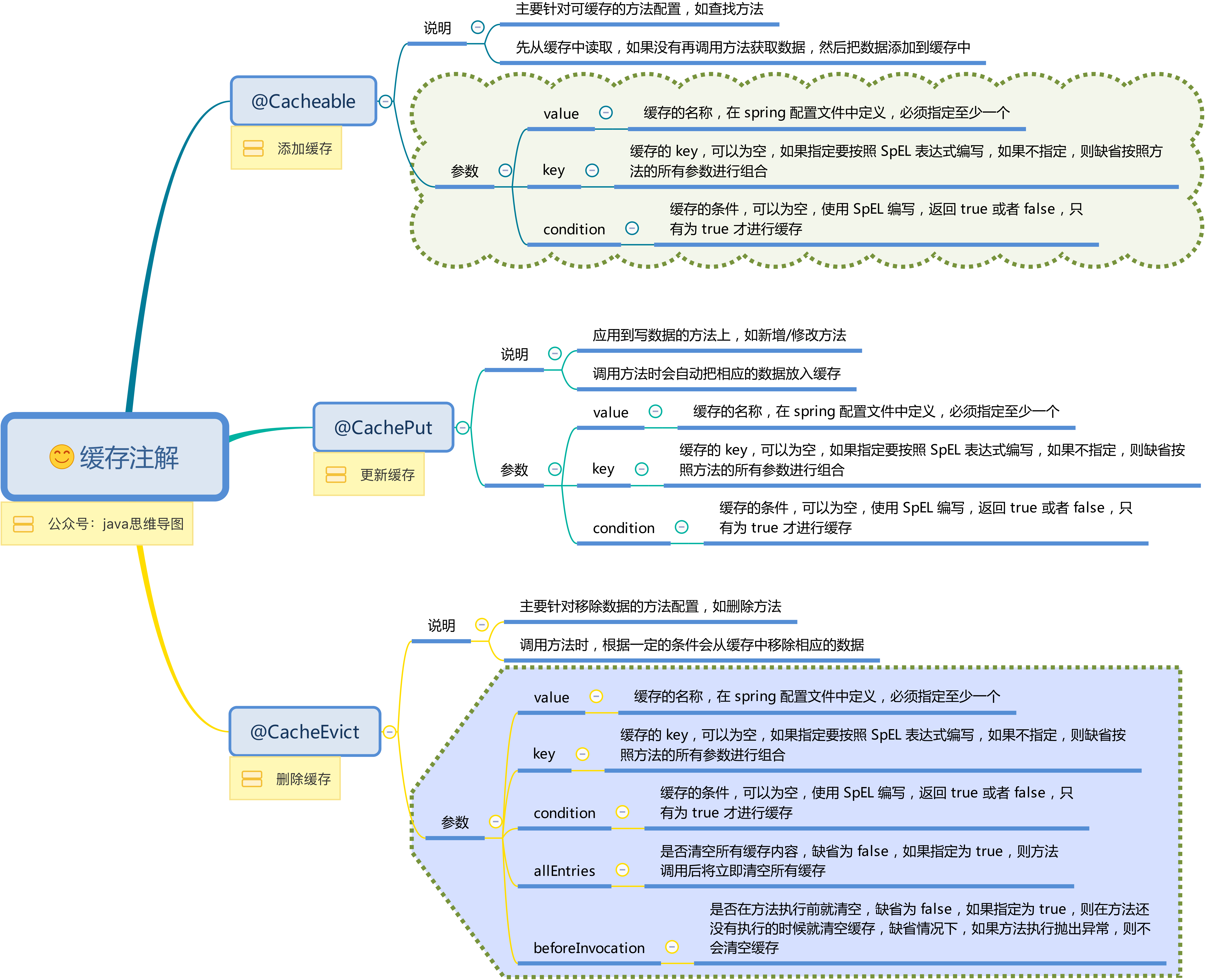


Redis可以做主从来集群，例如，在A电脑上装个Redis，作为主库；在其他电脑上装Redis，作为从库；这样主库拥有读和写的功能，而从库只拥有读的功能。每次主库的数据都会同步到从库中。

ehcache



创建一个ehcache.xml文件，里面配置cache的信息，这个配置是包含了集群的配置



😊 缓存作用



公众号：java思维导图

1 操作系统磁盘缓存->减少磁盘机械操作

2 数据库缓存->减少文件系统I/O

3 应用程序缓存->减少对数据库的查询

4 Web服务器缓存->减少应用服务器请求

5 客户端浏览器缓存->减少对网站的访问

😊 特性与缺憾

📖 公众号：java思维导图

特性

- 1.通过少量的配置 annotation 注解即可使得既有代码支持缓存
- 2.支持开箱即用 Out-Of-The-Box，即不用安装和部署额外第三方组件即可使用缓存
- 3.支持 Spring Express Language，能使用对象的任何属性或者方法来定义缓存的 key 和 condition
- 4.支持 AspectJ，并通过其实现任何方法的缓存支持（默认基于AOP方案，采用AspectJ会更灵活）
- 5.支持自定义 key 和自定义缓存管理者，具有相当的灵活性和扩展性
- 6.支持各种缓存实现，默认是基于ConcurrentMap实现的ConcurrentMapCache，同时支持ehcache实现。若需要使用Redis等缓存，引入redis的实现包即可。

缺憾

- 1.不支持TTL，也就是不能设置expires time。这点挺遗憾的，spring-cache认为这是各个cache实现自己去完成的事情，有方案但是只能设置统一的过期时间，这明显不够灵活。比如用户的抽奖次数、有效期等业务，当天有效，或者3天、一周有效，我们倾向于设置缓存有效期解决这个问题，而spring-cache却无法完成。
- 2.无法根据查询结果中的内容生成缓存key，比如getUser(uid)方法，想通过查询出来的user.email生成缓存key就无法实现了。
- 3.调试起来麻烦