1. 什么 Redis?

- Redis 是 个使 C 写成 ，开源 性 key-value 关 存数据库 它支持存储 value 型 对更多，包括string(字 串) list( ) set(合) zset(sorted set --有序 合)和 hash（哈希 型） Redis 数据 基于 存 ，所以很快，每 可以处 10万次 写操作，是已 性 最快 Key-Value DB Redis也可以实 数据写入 中，保 了数据 安全不丢失， 且Redis 操作是原子性

2. Redis 哪些优 ?

- 优点
  - 写性 优异， Redis 度是110000次/s，写 度是81000次/s
  - 支持数据持久化，支持AOF和RDB两 持久化方式
  - 支持事务，Redis 所有操作 是原子性 ，同时Redis 支持对几个操作合并后 原子性执
  - 数据 构丰富， 了支持string 型 value外 支持hash set zset list 数据 构
  - 支持主从复制，主机会 动将数据同步到从机，可以 写分
- 点
  - 数据库容 受到 内存 制，不 作海 数据 性 写，因此Redis 合 场景主 局 在 小数据 性 操作和 上
  - Redis 不具备 动容 和恢复功 ，主机从机 机 会导 前 分 写 求失 ， 待机器 启或 手动切换前 IP才 恢复
  - 主机 机， 机前有 分数据未 及时同步到从机，切换IP后 会引入数据不 ， 低了 可 性
  - Redis 支持在 扩容，在 容 到上 时在 扩容会变得很复杂 为 免 ， 人员在 上 时必 保有 够 ， 对 源 成了很大 浪

3. 使 redis 哪些好处?

- (1) 度快，因为数据存在内存中， 似于HashMap，HashMap 优势就是查找和操作 时 复杂度 很低
- (2)支持丰富数据 型，支持string，list，set，sorted set，hash
- (3) 支持事务，操作 是原子性，所 原子性就是对数据 更改 么全 执 ， 么全 不执
- (4) 丰富 性：可 于 存，消息，按key 期时 ， 期后将会 动删

4. 为什么 Redis / 为什么 存

主 从“ ”“ ”两 个 。

- 性 ：
  - 假如 户 次 数据库中 某些数据 个 会比 慢，因为是从 上 取 将 户 数据存在数 存中， 样下 次再 些数据 时候就可以 接从 存中取了 操作 存就是 接操作内存，所以 度 当快 如果数据库中 对应数据改变 之后，同步改变 存中 应 数据即可!
- 并发:
  - 接操作 存 够承受 求是 大于 接 数据库 ，所以我们可以 把数据库中 分数据 到 存中去， 样 户 分 求会 接到 存 不 数据库

5. 为什么 Redis 不 map/guava 做 存?

- 存分为本地 存和分布式 存 以 Java 为例，使 带 map 或 guava 实 是本地存，最主 点是 以及快 ， 命周期 jvm 毁 束，并且在多实例 情况下，每个实例 各 保存 份 存， 存不具有 性
- 使 redis 或 memcached 之 为分布式 存，在多实例 情况下，各实例共 份 存数据， 存具有 性 点是 保持 redis 或 memcached服务 可 ，整个 序架构上 为复杂

6. Redis为什么 么快

- 1 完全基于内存， 大 分 求是 内存操作， 常快 数据存在内存中， 似于HashMap，HashMap 优势就是查找和操作 时 复杂度 是O(1);
- 2 数据 构 单，对数据操作也 单，Redis 中 数据 构是专 ;
- 3 单 ， 免了不必 上下文切换和 争条件，也不存在多 或 多 导 切换消 CPU，不 去 各 ，不存在加 放 操作，没有因为可 出 死 导 性 消 ;
- 4 使 多 I/O 复 模型， 塞 IO;
- 5 使 底层模型不同，它们之 底层实 方式以及与客户 之 信 应 协 不 样，Redis 接 己构建了 VM 机制 ，因为 函数 ，会浪 定 时 去 动和求;

7. Redis 哪些数据 型

- Redis主 有5 数据 型，包括String，List，Set，Zset，Hash，满 大 分 使 求

数据类型	可以存储的值	操作	应用场景
String	字符串、整数或者浮点数	对整个字符串或者字符串的其中一部分执行操作 对整数和浮点数执行自增或者自减操作	做简单的键值对缓存
List	列表	从两端压入或者弹出元素 对单个或者多个元素进行修剪， 只保留一个范围内的元素	存储一些列表型的数据结构，类似粉丝列表、文章的评论列表之类的数据
Set	无序集合	添加、获取、移除单个元素 检查一个元素是否存在于集合中 计算交集、并集、差集 从集合里面随机获取元素	交集、并集、差集的操作，比如交集，可以把两个人的粉丝列表整一个交集
Hash	包含键值对的无序散列表	添加、获取、移除单个键值对 获取所有键值对 检查某个键是否存在	结构化的数据，比如一个对象
ZSet	有序集合	添加、获取、删除元素 根据分值范围或者成员来获取元素 计算一个键的排名	去重但可以排序，如获取排名前几名的用户

8. Redis 应 场

- 数器

可以对 String 增 减 ，从 实 数器功 Redis 内存型数据库 写性 常 ，很 合存储 写 数

- 存

将热点数据放到内存中， 内存 最大使 以及淘汰 来保 存 命中

- 会 存

可以使 Redis 来 存储多台应 服务器 会 信息 当应 服务器不再存储 户 会 信息，也就不再具有 态， 个 户可以 求任意 个应 服务器，从 更容易实 可 性以及可伸 性

- 全　　存（FPC）

基本　会　token之外，Redis　提供很　便　FPC平台　以Magento为例，Magento提供　个插件来使　Redis作为全　　存后　　此外，对WordPress　　户来　，Pantheon有　个　常好　插件 wp-redis，　个插件　帮助你以最快　度加　你曾浏

- 查找

例如 DNS　录就很　合使　Redis　　存储　查找　和　存　似，也是利　了 Redis 快　　查找性　但是查找　　内容不　失效，　　存　内容可以失效，因为　存不作为可　　数据来源

- 消息　列(发布/　　功　)

  List 是　个双向　　，可以　　lpush 和 rpop 写入和　取消息　不　最好使　Kafka　RabbitMQ　消息中　件

- 分布式　实

在分布式场景下，无法使　单机　境下　　来对多个　点上　　同步　可以使　Redis　带 SETNX 命令实　分布式　，　此之外，　可以使　官方提供　RedLock 分布式　实

- 其它

  Set 可以实　交　　并　　操作，从　实　共同好友　功　　ZSet 可以实　有序性操作，从　实　排　榜　功

9. 持久化

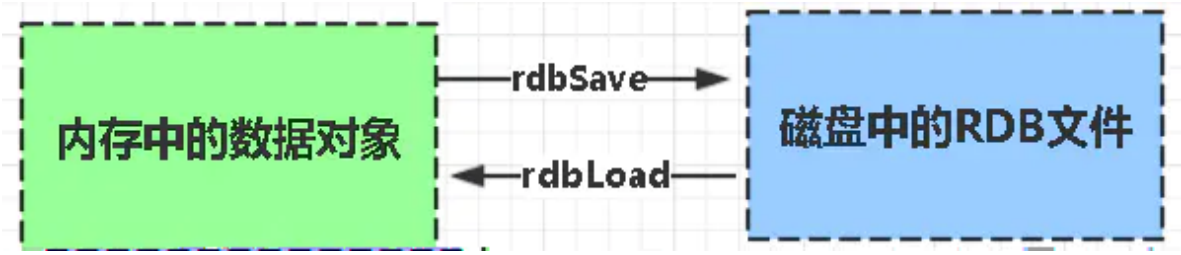
- 什么是Redis持久化？ 持久化就是把内存　数据写到　　中去，　止服务　机了内存数据丢失

10. Redis　持久化　制　什么？各　　优　　？

- Redis 提供两　持久化机制 RDB（　　）和 AOF 机制:

RDB：　Redis DataBase　写快

- RDB是Redis　　持久化方式　按照　定　时　将内存　数据以快照　形式保存到　　中，对应产　　数据文件为dump.rdb　　文件中　save参数来定义快照　周期



- 优点:

  1　只有　个文件 dump.rdb，方便持久化

  2　容灾性好，　个文件可以保存到安全

  3　性　最大化，fork 子　　来完成写操作，　主　　处　命令，所以是 IO 最大化　使　单子　　来　　持久化，主　　不会　　任何 IO 操作，保　了 redis　　性

  4.　对于数据　大时，比 AOF　启动效　更

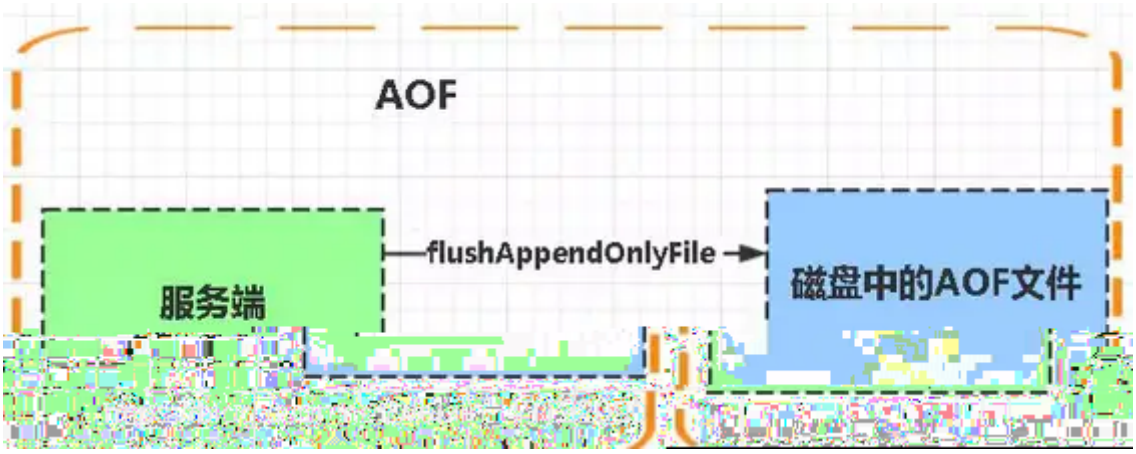
- 　点:

1　数据安全性低　RDB 是　　段时　　持久化，如果持久化之　redis 发　故　，会发　数据丢失　所以　　方式更　合数据　求不严　　时候)

2　AOF（Append-only file)持久化方式：是指所有　命令　　录以 redis 命令　求协　　格式完全持久化存储)保存为 aof 文件

AOF：持久化：

- AOF持久化(即Append Only File持久化)，则是将Redis执　　每次写命令　录到单　　日志文件中，当　启Redis会　新将持久化　日志中文件恢复数据
- 当两　方式同时开启时，数据恢复Redis会优先　择AOF恢复



- 优点：

  1　数据安全，aof 持久化可以　　appendfsync 属性，有 always，每　　次 命令操作就　录到 aof 文件中　次

  2　　append 模式写文件，即使中　服务器　机，可以　　redis-check-aof 工具　决数据性

  3　AOF 机制　rewrite 模式　AOF 文件没　rewrite 之前（文件　大时会对命令　　合并写），可以删　其中　某些命令（比如　操作　flushall））

- 　点：

  1　AOF 文件比 RDB 文件大，且恢复　度慢

  2　数据　大　时候，比 rdb 启动效　低

- 　　持久化　优　点是什么?
  - AOF文件比RDB更新　　，优先使　AOF　原数据
  - AOF比RDB更安全也更大
  - RDB性　比AOF好
  - 如果两个　　了优先加　AOF

11. 如何　择合　　持久化　式

- 　　来　，如果想　到　以　　PostgreSQL　数据安全性，你应　同时使　两　持久化功　　在　　情况下，当 Redis　启　时候会优先　入AOF文件来恢复原始　数据，因为在　常情况下AOF文件保存　数据　　比RDB文件保存　数据　　完整
- 如果你　常关心你　数据，但仍然可以承受数分　以内　数据丢失，　么你可以只使　RDB持久化
- 有很多　户　只使　AOF持久化，但并不推　　方式，因为定时　成RDB快照（snapshot）常便于　　数据库备份，并且 RDB 恢复数据　　度也　比AOF恢复　　度　快，　此之外，使　RDB　可以　免AOF　序　bug
- 如果你只希望你　数据在服务器　　时候存在，你也可以不使　任何持久化方式

12. Redis持久化数据和 存 么做扩容?

- 如果Redis 当做 存使 , 使 性哈希实 动态扩容 容
- 如果Redis 当做 个持久化存储使 , 必 使 固定 keys-to-nodes映射关 , 点 数 旦 定不 变化 否则 (即Redis 点 动态变化 情况) , 必 使 可以在 时 数据再平 套 , 当前只有Redis 可以做到 样

13. Redis 删

17. Redis 内存 哪些

`Redis Redis 于 不 ， 么 入且 。`

- 全局 择性
  - noeviction：当内存不 以容 新写入数据时，新写入操作会报
  - allkeys-lru：当内存不 以容 新写入数据时，在 中， 最 最少使 key （个是最常 ）
  - allkeys-random：当内存不 以容 新写入数据时，在 中， 机 某个key
- 期时 择性
  - volatile-lru：当内存不 以容 新写入数据时，在 了 期时 中， 最 最少使 key
  - volatile-random：当内存不 以容 新写入数据时，在 了 期时 中， 机 某个key
  - volatile-ttl：当内存不 以容 新写入数据时，在 了 期时 中，有更早 期时 key优先
- 总

`Redis 不会 key 。 于 不 ； 于 。`

18. Redis主 什么 ?

- 内存

19. Redis 内存 完了会发 什么?

- 如果 到 上 ，Redis 写命令会 回 信息（但是 命令 可以正常 回 ）或 你可以 内存淘汰机制，当Redis 到内存上 时会冲刷掉旧 内容

20. Redis如何做内存优化?

- 可以好好利 Hash,list,sorted set,set 合 型数据，因为 常情况下很多小 Key-Value可以 更 凑 方式存放到 尽可 使 散列 （hashes），散列 （是 散列 存储 数少）使 内存 常小，所以你应 尽可 将你 数据模型抽 到 个散列 比如你 web 中有 个 户对 ，不 为 个 户 名 ，姓氏， ，密 单 key， 是应 把 个 户 所有信息存储到 张散列

型

21. Redis 型

Redis基于Reactor模式开发了 事件处 器， 个处 器 为文件事件处 器（file event handler） 它 成 构为4 分：多个套接字 IO多 复 序 文件事件分派器 事件处 器 因为文件事件分派器 列 消 是单 ，所以Redis才叫单 模型

- 文件事件处 器使 I/O 多 复 （multiplexing） 序来同时 听多个套接字，并根据套接字前执 任务来为套接字关 不同 事件处 器
- 当 听 套接字准备好执 接应 （accept） 取（read） 写入（write） 关（close） 操作时，与操作 对应 文件事件就会产 ， 时文件事件处 器就会 套接字之前关 好 事件处 器来处 些事件

然文件事件处 器以单 方式 ，但 使 I/O 多 复 序来 听多个套接字，文件事件处 器既实 了 性 信模型，又可以很好地与 redis 服务器中其他同样以单 方式模块 对接， 保持了 Redis 内 单 单性

事务

22. 什么 事务?

- 事务是 个单 操作：事务中 所有命令 会序列化 按 序地执 事务在执中，不会 其他客户 发 来 命令 求所打断
- 事务是 个原子操作：事务中 命令 么全 执 ， 么全 不执

23. Redis事务 念

- Redis 事务 本 是 MULTI EXEC WATCH 命令 合 事务支持 次执 多个命令， 个事务中所有命令 会 序列化 在事务执 ，会按照 序串 化执 列中 命令，其他客户 提交 命令 求不会插入到事务执 命令序列中
- 总 ：redis事务就是 次性 序性 排他性 执 个 列中 列命令

24. Redis事务　三个

1. 事务开始 MULTI
2. 命令入
3. 事务执　EXEC

```
事　　中，　　EXEC、DISCARD、WATCH、MULTI之　　，　会　　入　中
```

25. Redis事务　关命令

Redis事务功　是　　MULTI　EXEC　DISCARD和WATCH 四个原　实

Redis会将　个事务中　所有命令序列化，然后按　序执

1. **redis 不支持回　**，“Redis 在事务失　时不　　回滚，　是　　执　余下　命令”，所以 Redis 内　可以保持　单且快
2. **如　在　个事务中　命令出　　，　么所　　命令　不会执　**；
3. **如　在　个事务中出　　，　么　　命令会　执**

- WATCH 命令是　个乐　　，可以为 Redis 事务提供 check-and-set （CAS）　为　可以　控　个或多个　，　旦其中有　个　　修改（或删　），之后　事务就不会执　，　控　　持　到EXEC命令
- MULTI命令　于开启　个事务，它总是　回OK　MULTI执　之后，客户　可以　　向服务器发任意多条命令，　些命令不会　即　执　，　是　放到　个　列中，当EXEC命令　　时，所有　列中　命令才会　执
- EXEC：执　所有事务块内　命令　　回事务块内所有命令　　回值，按命令执　　先后　序排列　当操作　打断时，　回　值 nil
- 　　DISCARD，客户　可以清　事务　列，并放弃执　事务，并且客户　会从事务　态中出
- UNWATCH命令可以取消watch对所有key　　控

26. 事务　　(ACID)

- 原子性（Atomicity）
  原子性是指事务是　个不可分割　工作单位，事务中　操作　么　发　，　么　不发
- 　　性（Consistency）
  事务前后数据　完整性必　保持
- 　　性（Isolation）
  多个事务并发执　时，　个事务　执　不应影响其他事务　执
- 持久性（Durability）
  持久性是指　个事务　旦　提交，它对数据库中数据　改变就是永久性　，接下来即使数据库发　故　也不应　对其有任何影响

```
Redis　事　　ACID中　一　　，　他　　不　　。　　_AOF_　久
下，　且appendfsync　　值为always　，事　也　　久　。
```

27. Redis事务支持　　吗

- Redis 是单　　序，并且它保　在执　事务时，不会对事务　　中断，事务可以　　到执　完所有事务　列中　命令为止　因此，**Redis　事务　　带**

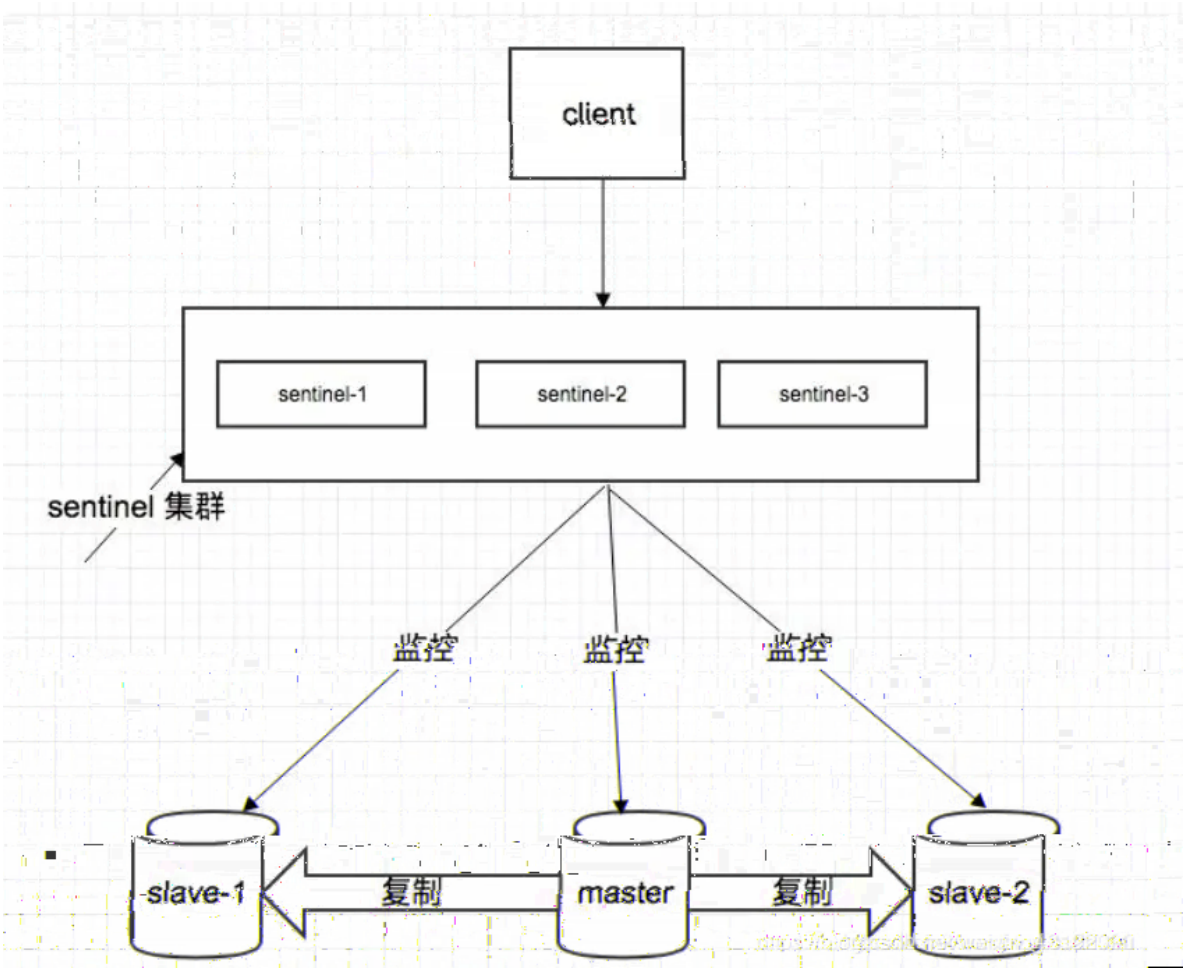
28. Redis事务保　原子　吗，支持回　吗

- Redis中，单条命令是原子性执　　，但**事务不保　原子　，且　　回**　　事务中任意命令执　失　，其余　命令仍会　执

29. Redis事务其他实

- 基于Lua　本，Redis可以保　　本内　命令　次性　按　序地执　，　其同时也不提供事务　　回滚，执　　中如果　分命令　　，剩下　命令　是会　　完*　基于中　标　变　，　　另外　标　变　来标　事务是否执　完成，　取数据时先取　标　变　判断是否事务执　完成　但　样会　　外写代　实　，比

30.　兵　式



兵　介

`sentinel，中　　兵。　兵　redis　　中　　一个　件，主　以下　：`

- 　　控:　　控 redis master 和 slave　　是否正常工作
- 消息　　：如果某个 redis 实例有故　，　么　兵　　发　消息作为报　　员
- 故　　：如果 master node 挂掉了，会　动　　到 slave node 上
- 　　中心：如果故　　发　了，　　client 客户　新　master 地址

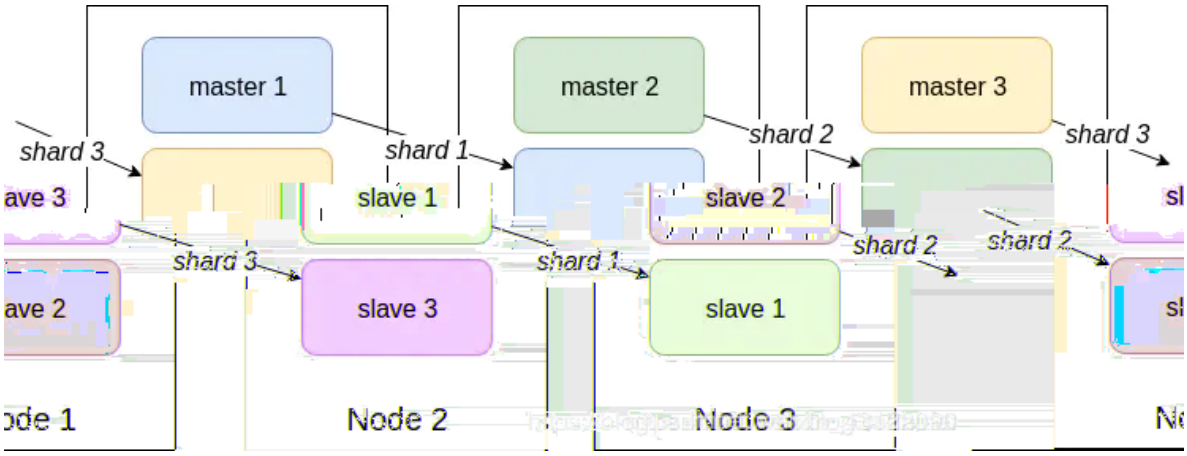
兵　于实　redis　　可　，本　也是分布式　，作为　个　兵　　去　　，互　协同工作

- 故　　时，判断　个 master node 是否　机了，　　大　分　　兵　同意才　，涉及到了分布式　举
- 即使　分　兵　点挂掉了，　兵　　是　正常工作　，因为如果　个作为　可　机制　　成分　故　　本　是单点　，　就很　　了

兵　　心

- 　兵　少　　3 个实例，来保　　己　健　性
- 　兵 + redis 主从　　架构，是**不保　数据　丢失**　，只　保　redis　　可　性
- 对于　兵 + redis 主从　　复杂　　架构，尽　在测　　境和　产　境，　　充　　测　和演

31. 官　Redis Cluster　　（　务　　）



- redis　　模式　工作原　　下么？在　　模式下，redis　key 是如何寻址　？分布式寻址　有哪些　法？了　　性 hash　法吗？

介

- Redis Cluster是　　服务　Sharding技术，3.0　本开始正式提供　Redis Cluster并没有使　性hash，　是　　slot(槽)　概念，　共分成16384个槽　将　求发　到任意　点，接收到　求　点会将查　　求发　到正　　点上执

1.　　哈希　方式，将数据分　，每个　点均分存储　定哈希槽(哈希值)区　　数据，　　分　了 16384 个槽位
2. 每份数据分　会存储在多个互为主从　多　点上
3. 数据写入先写主　点，再同步到从　点(支持　　为　塞同步)
4. 同　分　多个　点　　数据不保持　　性
5.　取数据时，当客户　操作　key没有分　在　　点上时，redis会　回　向指令，指向正　　点
6. 扩容时时　　把旧　点　数据　　分到新　点

- 在 redis cluster 架构下，每个 redis　放开两个　口号，比如　个是 6379，另外　个就是 加1w　口号，比如 16379
    - 16379　口号是　来　　点　　信　，也就是 cluster bus　东　，cluster bus　　信，　来　　故　检测　　更新　故　　授权　cluster bus　了另外　　二　制　协　，`gossip` 协　，　于　点　　效　数据交换，占　更少　　带宽和处　时

内　　信　制

- 基本　信原

- 元数据 护有两 方式: 中式 Gossip 协 redis cluster 点 gossip 协信

分布式寻址

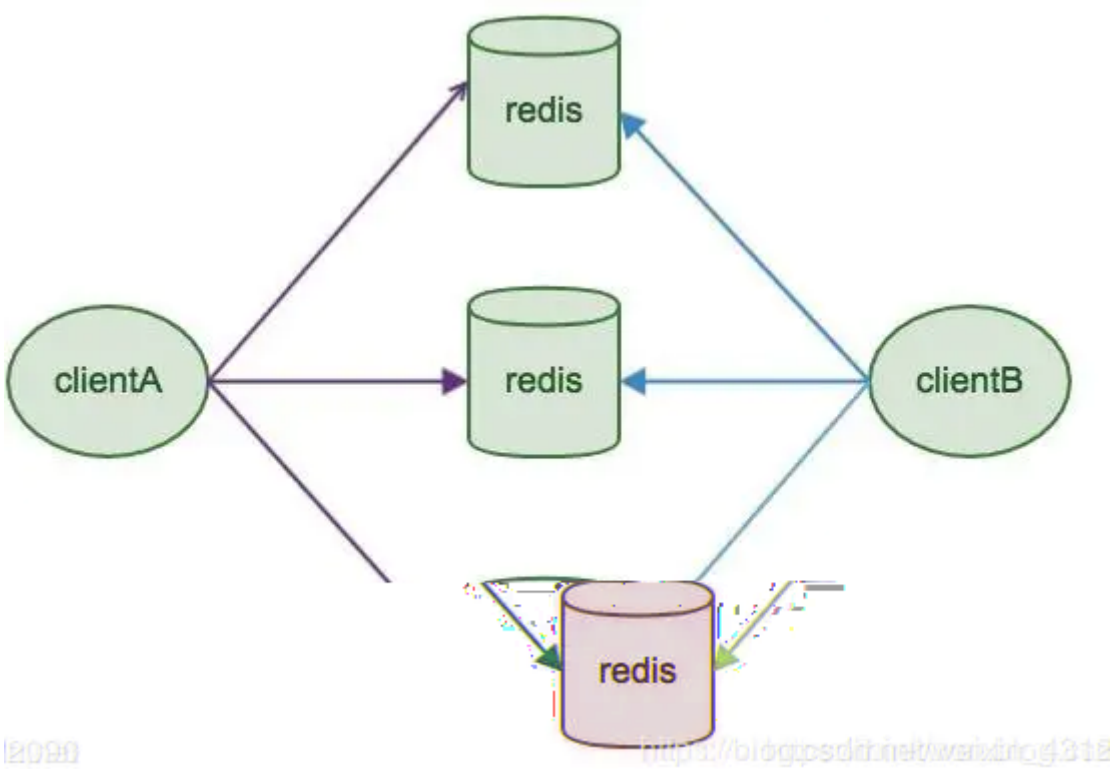
- hash 法(大 存 建)
- 性 hash 法(动 存)+ 拟 点(动 均)
- redis cluster hash slot 法

优

- 无中心架构，支持动态扩容，对业务 明
- 具备Sentinel 控和 动Failover(故) 力
- 客户 不 接 所有 点， 接 中任何 个可 点即可
- 性 ，客户 redis服务，免去了proxy代 损

- 也很复杂，数据 人工干
- 只 使 0号数据库
- 不支持批 操作(pipeline 操作)
- 分布式 和存储模块 合

32. 基于客户 分



介

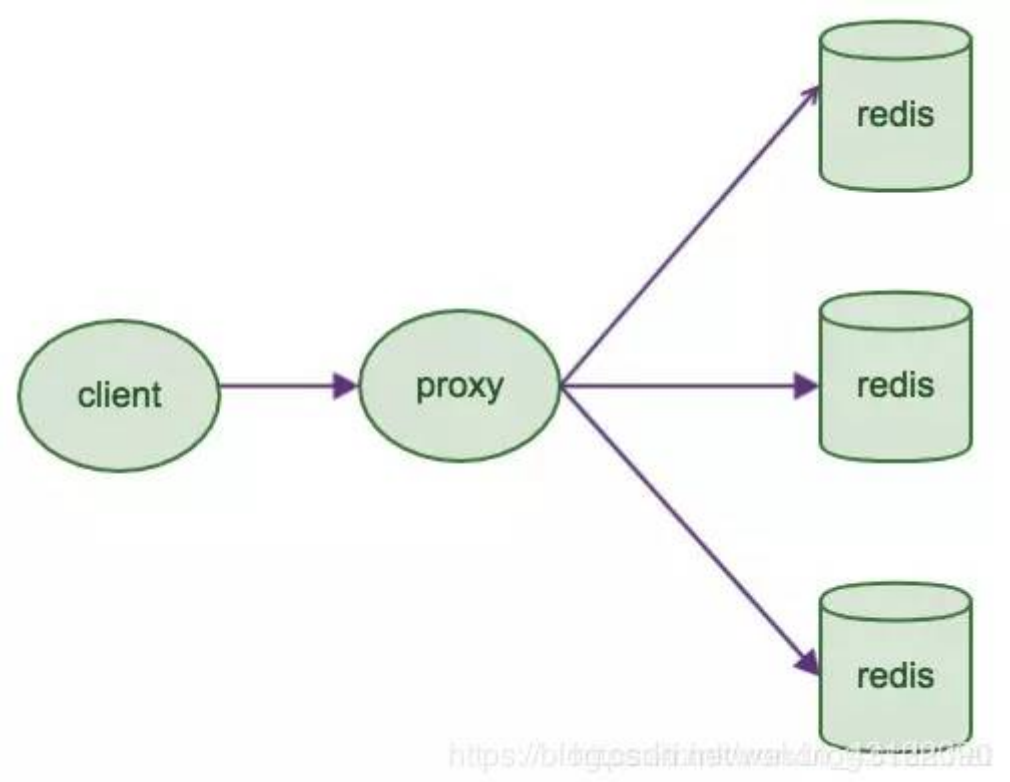
- Redis Sharding是Redis Cluster出来之前，业 普 使 多Redis实例 方法 其主 思想是 哈希 法将Redis数据 key 散列， hash函数， 定 key会映射到 定 Redis 点上 Java redis客户 动jedis，支持Redis Sharding功 ，即ShardedJedis以及 合 存池 ShardedJedisPool

优

- 优势在于 常 单，服务 Redis实例彼此 ， 互无关 ，每个Redis实例像单服务器 样 ， 常容易 性扩展， 灵活性很强

- 于sharding处 放到客户 ， 模 步扩大时 带来挑战
- 客户 sharding不支持动态增删 点 服务 Redis实例 拓扑 构有变化时，每个客户 更新 整 接不 共享，当应 模增大时， 源浪 制 优化

33. 基于代 务器分



介

- 客户 发 求到 个代 件，代 析客户 数据，并将 求 发 正 点，最后将果回复 客户

征

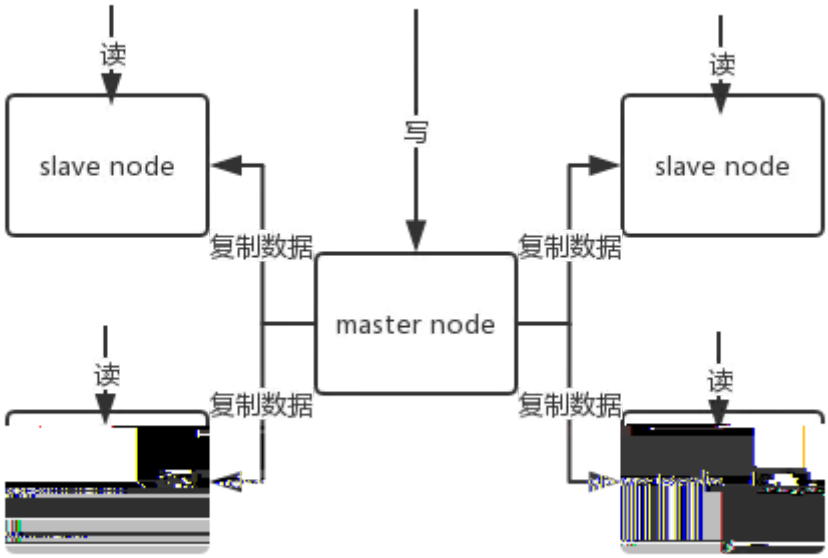
- 明接入，业务 序不 关心后 Redis实例，切换成本低
- Proxy 和存储 是
- 代 层多了 次 发，性 有所损

业 开

- Twtter开源 Twemproxy
- 开源 Codis

34. Redis 主从

- 单机 redis， 够承 QPS 大概就在上万到几万不 对于 存来 ， 是 来支 **并发** 因此架构做成主从(master-slave)架构， 主多从，主 写，并且将数据复制到其它 slave 点，从 点 所有 **全 从** 样也可以很 松实 水平扩容，**支 并发**




```
redis replication -> 主从      ->          ->
```

redis replication   心   制

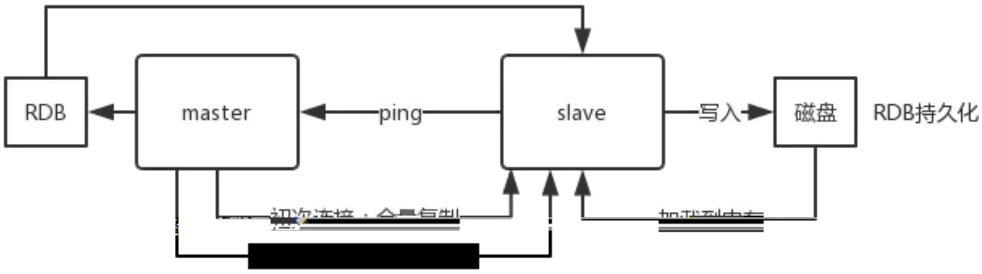
- redis   **异   式**复制数据到 slave   点，不   redis2.8 开始，slave node 会周期性地   己 每次复制   数据   ；
-     个 master node 是可以   多个 slave node   ；
- slave node 也可以   接其他   slave node；
- slave node 做复制   时候，不会 block master node   正常工作；
- slave node 在做复制   时候，也不会 block 对   己   查   操作，它会   旧   数据   来提供服务；但是复制完成   时候，   删   旧数据   ，加   新数据   ，   个时候就会暂停对外服务了；
- slave node 主   来   横向扩容，做   写分   ，扩容   slave node 可以提   吞吐

意：

- 如果   了主从架构，   么建   必   **开启** master node   持久化，不建   slave node 作为 master node   数据热备，因为   样   ，如果你关掉 master   持久化，可   在 master   机启   时候数据是   ，然后可   复制，slave node   数据也丢了
- 另外，master   各   备份方案，也   做   万   本地   所有文件丢失了，从备份中挑   份 rdb 去恢复 master，   样才   **保启动   候，   数据**   ，即使   了后   可   机制，slave node 可以   动接   master node，但也可   sentinel   没检测到 master failure，master node 就   动   启了，   是可   导   上   所有   slave node 数据   清

redis 主从复制   心原

- 当启动   个 slave node   时候，它会发   个



原

1. 当从库和主库建　MS关　后，会向主数据库发　SYNC命令
2. 主库接收到SYNC命令后会开始在后台保存快照(RDB持久化　　)，并将期　接收到　写命令　存　　来
3. 当快照完成后，主Redis会将快照文件和所有　存　写命令发　　从Redis
4. 从Redis接收到后，会　入快照文件并且执　收到　　存　命令
5. 之后，主Redis每当接收到写命令时就会将命令发　从Redis，从　保　数据

- 所有　slave　点数据　复制和同步　　master　点来处　，会照成master　点压力太大，使　主从从　构来　决

35. Redis　　主从复制　型　　?

- 为了使在　分　点失　或　大　分　点无法　信　情况下　　仍然可　，所以　　使　了主从复制模型，每个　点　会有N-1个复制品

36.　产　境中　redis　　么　　?

- redis cluster，10 台机器，5 台机器　　了 redis 主实例，另外 5 台机器　　了 redis　从实例，每个主实例挂了　个从实例，5 个　点对外提供　写服务，每个　点　　写　峰qps可　可以　到每　5 万，5 台机器最多是 25 万　写　求/s
- 机器是什么　　？32G 内存+ 8 核 CPU + 1T　　，但是分　　redis　　是10g内存，　　上　产　境，redis　内存尽　不　　10g，　　10g 可　会有
- 5 台机器对外提供　写，　共有 50g 内存
- 因为每个主实例　挂了　个从实例，所以是　可　　，任何　个主实例　机，　会　动故　　，redis 从实例会　动变成主实例　　提供　写服务
- 你往内存　写　是什么数据？每条数据　大小是多少？商品数据，每条数据是 10kb　100 条数据是 1mb，10 万条数据是 1g　常　内存　是 200 万条商品数据，占　内存是 20g，仅仅不到总内存　50%　　前　峰期每　就是 3500 左右　　求

```
公　，会　　team　　。
```

37.　　Redis哈希　　念?

- Redis　　没有使　　性hash，是引入了哈希槽　概念，Redis　　有16384个哈希槽，每个key　　CRC16校　后对16384取模来决定放　哪个槽，　　每个　点　　分hash槽

38. Redis 会 写操作丢失吗？为什么？

- Redis并不 保 数据 强 性， 意味 在实 中 在 定 条件下可 会丢失写操作

39. Redis 之 如何复制 ？

- 异步复制

40. Redis 大 个数 多少？

- 16384个

41. Redis 如何 择数据库？

- Redis 前无法做数据库 择， 在0数据库

分区

42. Redis 单 ，如何提 多 CPU 利 ？

- 可以在同 个服务器 多个Redis 实例，并把他们当作不同 服务器来使 ，在某些时候，无 如何 个服务器是不够 ， 所以，如果你想使 多个CPU，你可以 下分 （shard）

43. 为什么 做Redis分区？

- 分区可以 Redis 更大 内存，Redis将可以使 所有机器 内存 如果没有分区，你最多只 使 台机器 内存 分区使Redis 力 单地增加 机得到成倍提升，Redis 带宽也会 机和 卡 增加 成倍增

44. 你 哪些Redis分区实 ？

- 客户 分区就是在客户 就已 决定数据会 存储到哪个redis 点或 从哪个redis 点 取 大多数客户 已 实 了客户 分区
- 代 分区 意味 客户 将 求发 代 ，然后代 决定去哪个 点写数据或 数据 代 根据分区 则决定 求哪些Redis实例，然后根据Redis 响应 果 回 客户 redis和 memcached 代 实 就是Twemproxy
- 查 (Query routing) 意思是客户 机地 求任意 个redis实例，然后 Redis将 求发 正 Redis 点 Redis Cluster实 了 混合形式 查 ，但并不是 接将 求从 个redis 点 发到另 个redis 点， 是在客户 帮助下 接redirected到正 redis 点

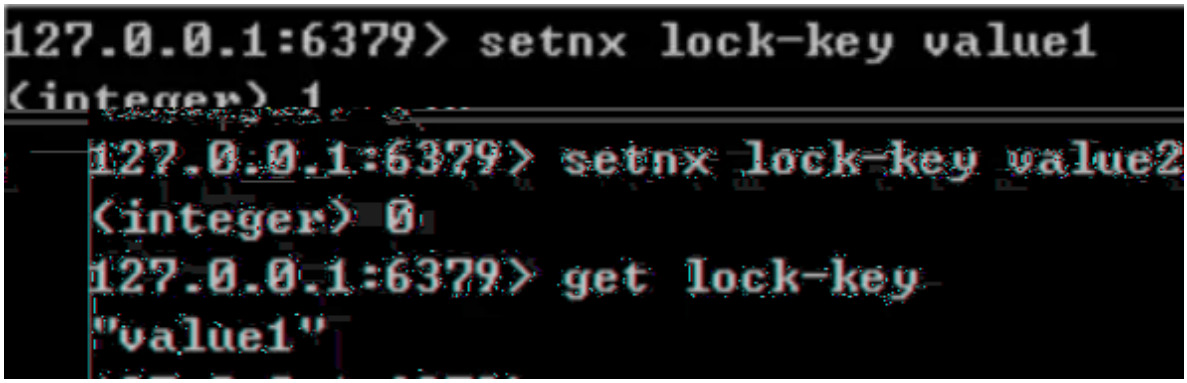
45. Redis分区 什么 ？

- 涉及多个key 操作 常不会 支持 例如你不 对两个 合求交 ，因为他们可 存储到不同 Redis实例（实 上 情况也有办法，但是不 接使 交 指令）
- 同时操作多个key,则不 使 Redis事务.
- 分区使 度是key，不 使 个 常 排序key存储 个数据 （The partitioning granularity is the key, so it is not possible to shard a dataset with a single huge key like a very big sorted set）
- 当使 分区 时候，数据处 会 常复杂，例如为了备份你必 从不同 Redis实例和主机同时收 RDB / AOF文件
- 分区时动态扩容或 容可 常复杂 Redis 在 时增加或 删 Redis 点， 做到最大 度对 户 明地数据再平 ，但其他 些客户 分区或 代 分区方法则不支持 性 然 ，有 分 技术也可以 好 决 个

分布式

46. Redis实 分布式

- Redis为单 单 模式， 列模式将并发 变成串 ，且多客户 对Redis 接 并不存在 争关 Redis中可以使 setNx命令实 分布式
- 当且仅当 key 不存在，将 key 值 为 value 定 key 已 存在，则 setNx不做任何动作
- SETNX 是 SET if Not eXists (如果不存在，则 SET) 写
- 回值: 成功， 回 1 失 ， 回 0



- 使 setNx完成同步 流 及事 如下:
- 使 SETNX命令 取 ， 回0（key已存在， 已存在）则 取失 ，反之 取成功
- 为了 止 取 后 序出 异常，导 其他 / setNx命令总是 回0 入死 态， 为 key 个“合 ”期时 放 ，使 DEL命令将 数据删

47. 如何 决 Redis 并发 争 Key

- 所 Redis 并发 争 Key 也就是多个 同时对 个 key 操作，但是最后执 序和我们期望 序不同， 样也就导 了 果 不同!
- 推 方案：分布式 （zookeeper 和 redis 可以实 分布式 ） （如果不存在 Redis 并发 争 Key ，不 使 分布式 ， 样会影响性 ）
- 基于zookeeper临时有序 点可以实 分布式 大 思想为：每个客户 对某个方法加 时，在zookeeper上 与 方法对应 指定 点 录下， 成 个唯 时有序 点 判断是否 取 方式很 单，只 判断有序 点中序号最小 个 当 放 时候，只 将 个时 点删 即可 同时，其可以 免服务 机导 无法 放， 产 死 完成业务流 后，删 对应 子 点 放

中， 从以 为主。 以 zookeeper。

48. 分布式Redis 前 做 后 上 了再做好？为什么？

- 既然Redis是如此 （单实例只使 1M内存），为 止以后 扩容，最好 办法就是 开始就启动 多实例 即便你只有 台服务器，你也可以 开始就 Redis以分布式 方式 ，使 分区，在同 台服务器上启动多个实例
- 开始就多 几个Redis实例，例如32或 64个实例，对大多数 户来 操作 来可 比烦，但是从 久来 做 点 是值得
- 样 ，当你 数据不断增 ， 更多 Redis服务器时，你 做 就是仅仅将Redis实例从 台服务 到另外 台服务器 已（ 不 新分区 ） 旦你添加了另 台服务器，你 将你 半 Redis实例从 台机器 到 二台机器

49. 什么 RedLock

- Redis 官方 提出了 权威 基于 Redis 实 分布式 方式名叫 *Redlock*，此 方式比原先单 点 方法更安全 它可以保 以下 性：
  1. 安全 性：互斥 ，即永 只有 个 client 拿到
  2. 免死 ：最 client 可 拿到 ，不会出 死 情况，即使原本 住某 源 client crash 了或 出 了 分区
  3. 容 性：只 大 分 Redis 点存活就可以正常提供服务

存异常

50. 什么 redis ？

- 就是 户 求 redis去 求mysql服务器，导 mysql压力 但 个web服务 ，极容易出 就是mysql，所以才 redis去分担mysql 压力，所以 是万万 免
- 决方法：
  1. 从 存取不到 数据，在数据库中也没有取到， 时也可以将key-value对写为key-null，存有效时 可以 点，如30 （ 太 会导 正常情况也没法使 ） 样可以 止攻击 户反复 同 个id暴力攻击
  2. 接口层增加校 ，如 户 权校 ，id做基 校 ，id<=0 接拦截；
  3. 布 滤器，将所有可 存在 数据哈希到 个 够大 bitmap 中， 个 定不存在数据会 个 bitmap 拦截掉，从 免了对底层存储 查 压力

51. 什么 redis 崩?

- 就是redis服务 于 大 机，导 mysql 大也 机，最 整个
- 决方法：
  1. redis ，将原来 个人干 工作，分发 多个人干
  2. 存 热（关 外 ，先开启mysql， 热 本将热点数据写入 存中，启动存 开启外 服务）
  3. 数据不 同 存时 ，不然 期时，redis压力会大

52. 什么 redis ？

- 并发下， 于 个key失效， 导 多个 去mysql查同 业务数据并存到redis（并发下，存了多份数据）， 段时 后，多份数据同时失效 导 压力 增
- 决方法：
  1. 分 存（ 存两份数据， 二份数据 存时 点作为备份， 份数据 于 求命中，如果 二份数据 命中 明 份数据已 期， 去mysql 求数据 新 存两份数据）
  2. 划任务（假如数据 存时 为30分 ， 划任务就20分 执 次更新 存数据）

53. 存

- **存** 就是 上 后，将 关 存数据 接加 到 存 样就可以 免在 户 求时候，先查 数据库，然后再将数据 存 ！ 户 接查 事先 热 存数据！
- **决**
  1. 接写个 存刷新 ，上 时手工操作 下；
  2. 数据 不大，可以在 启动 时候 动 加 ；
  3. 定时刷新 存；

54. 存

- 当 剧增 服务出 （如响应时 慢或不响应）或 核心服务影响到核心流 性 时，仍然 保 服务 是可 ，即使是有损服务 可以根据 些关 数据 动 ，也可以 开关实 人工
- **存** 最 是保 核心服务可 ，即使是有损 且有些服务是无法 （如加入 ）
- 在 之前 对 ， 是不是可以丢 保 ；从 出哪些必 死保护，哪些可 ；比如可以参 日志 别 案：
  1. ：比如有些服务偶尔因为 抖动或 服务正在上 时，可以 动 ；
  2. 告：有些服务在 段时 内成功 有波动（如在95~100%之 ），可以 动 或人工 ，并发 告 ；
  3. ：比如可 低于90%，或 数据库 接池 打爆了，或 然 增到 承受 最大 值，此时可以根据情况 动 或 人工 ；
  4. 严 ：比如因为 殊原因数据 了，此时 急人工
- 服务 ，是为了 止Redis服务故 ，导 数据库 发 崩 因此，对于不 存数据，可以 取服务 ，例如 个比 常 做法就是，Redis出 ，不去数据库查 ， 是 接 回 值 户

55. 数据和冷数据

- 热点数据， 存才有价值
- 对于冷数据 ，大 分数据可 没有再次 到就已 挤出内存，不仅占 内存， 且价值不大 修改 数据， 情况 使 存
- 对于热点数据，比如我们 某IM产品， 日 模块，当天 寿星列 ， 存以后可 取数十万次 再举个例子，某导 产品，我们将导 信息， 存以后可 取数 万次
- 数据更新前 少 取两次， 存才有意义 个是最基本 ，如果 存 没有 作 就失效了， 就没有太大价值了
- 存不存在，修改 很 ，但是又不得不 存 场景 ？有！比如， 个 取接口对数据库压力很大，但是又是热点数据， 个时候就 存手段，减少数据库 压力，比如我们 某助手产品 ，点 数，收 数，分享数 是 常典型 热点数据，但是又不断变化，此时就将数据同步保存到Redis 存，减少数据库压力

56. 存 key

- 存中 个Key(比如 个促 商品)，在某个时 点 期 时候，恰好在 个时 点对 个Key有大 并发 求 来， 些 求发 存 期 会从后 DB加 数据并回 到 存， 个时候大并发 求可 会 把后 DB压

决

- 对 存查 加 ，如果KEY不存在，就加 ，然后查DB入 存，然后 ；其他 如果发 有 就 待，然后 后 回数据或 入DB查

常 工具

57. Redis支持 Java客户 哪些? 官 推 哪个?

- Redisson Jedis lettuce ，官方推 使 Redisson

58. Redis和Redisson 什么关 ?

- Redisson是 个 分布式协 Redis客服 ， 帮助 户在分布式 境中 松实 些Java 对 (Bloom filter, BitSet, Set, SetMultimap, ScoredSortedSet, SortedSet, Map, ConcurrentMap, List, ListMultimap, Queue, BlockingQueue, Deque, BlockingDeque, Semaphore, Lock, ReadWriteLock, AtomicLong, CountDownLatch, Publish / Subscribe, HyperLogLog)

59. Jedis与Redisson对 什么优 ?

- Jedis是Redis Java实 客户 ，其API提供了比 全 Redis命令 支持；Redisson实 了分布式和可扩展 Java数据 构，和Jedis 比，功 为 单，不支持字 串操作，不支持排序 事务 分区 Redis 性 Redisson 宗旨是促 使 对Redis 关注分 ，从 使 够将 力更 中地放在处 业务 上

其他

60. Redis与Memcached 区别

- 两 是 关 型内存 值数据库， 在公司 是 Redis 来实 存， 且 Redis 也来 强大了！Redis 与 Memcached 主 有以下不同：

对参数	Redis	Memcached
型	1. 支持内存 2. 关 型数据库	1. 支持内存 2. 值对形式 3. 存形式
数据存储 型	1. String 2. List 3. Set 4. Hash 5. Sort Set ZSet	1. 文本型 2. 二 制 型
查 操作 型	1. 批 操作 2. 事务支持 3. 每个 型不同 CRUD	1.常 CRUD 2. 少 其他命令
加功	1. 发布/ 模式 2. 主从分区 3. 序列化支持 4. 本支持 Lua 本	1. 多 服务支持
IO 模型	1. 单 多 IO 复 模型	1. 多 , 塞IO模式
事件库	封 易事件库 AeEvent	族 LibEvent事件库
持久化支持	1. RDB 2. AOF	不支持
模式	原 支持 cluster 模式，可以实 主从复制， 写分	没有原 模式， 依 客户 来实 往 中分写入数据
内存 机制	在 Redis 中，并不是所有数据存储在内存中，可以将 些很久没 value 交换到	Memcached 数据则会 在内存中，Memcached 将内存分割成 定 度 块来存储数据，以完全 决内存 但是 方式会使得内存 利 不 ，例如块 大小为 128 bytes，只存储 100 bytes 数据， 么剩下 28 bytes 就浪 掉了
场景	复杂数据 构，有持久化， 可求，value存储内容 大	key-value，数据 常大，并发 常大 业务

1. memcached所有 值均是 单 字 串，redis作为其替代 ，支持更为丰富 数据 型
2. redis 度比memcached快很多
3. redis可以持久化其数据

61. 如何保 存与数据库双写 数据 ？

- 你只　　存，就可　会涉及到　存与数据库双存储双写，你只　是双写，就　定会有数据　　性　　，　么你如何　决　　性　　？
- 　　来　，就是如果你　　不是严格　求　存+数据库必　　性　　，　存可以　微　　数据库偶尔有不　　情况，最好不　做　个方案，　　求和写　求串　化，串到　个内存　列　去，　样就可以保　　定不会出　不　　情况
- 串　化之后，就会导　　吞吐　会大幅度　　低，　比正常情况下多几倍　机器去支　　上　个　求
- 　有　　方式就是可　会暂时产　不　　情况，但是发　　几　　别小，就是**先　　数据库，后再删　　存**

场	描	决
先写　存，再写数据库，　存写成功，数据库写失	存写成功，但写数据库失　或响应延　，则下次　取（并发　）　存时，就出	个写　存　方式，本　就是　　，　改为先写数据库，把旧　存　为失效；　取数据　时候，如果　存不存在，则　取数据库再写　存
先写数据库，再写存，数据库写成功，存写失	写数据库成功，但写　存失　，则下次　取（并发　）　存时，则不到数据	存使　时，假如　　存失　，先　数据库，再回写　存　方式实
存异步刷新	指数据库操作和写　存不在　个操作步　中，比如在分布式场景下，无法做到同时写　存或　　异步刷新（　　措施）时候	定哪些数据　合此　场景，根据　值　定合　　数据不　　时　，　户数据刷新　时

62. Redis常　　和　决　　？

1. Master最好不　做任何持久化工作，包括内存快照和AOF日志文件，　别是不　启　内存快照做持久化
2. 如果数据比　关　，某个Slave开启AOF备份数据，　　为每　同步　次
3. 为了主从复制　　度和　接　　定性，Slave和Master最好在同　个局域网内
4. 尽　　免在压力　大　主库上增加从库
5. Master　　BGREWRITEAOF　写AOF文件，AOF在　写　时候会占大　　CPU和内存　源，导服务load　　，出　　暂服务暂停
6. 为了Master　　定性，主从复制不　　图　　构，　单向　　构更　定，即主从关　为：Master<–Slave1<–Slave2<–Slave3…，　样　　构也方便　决单点故　　，实　Slave对Master　替换，也即，如果Master挂了，可以　　启　Slave1做Master，其他不变

63. Redis官　为什么不提供Windows　　？

- 因为　前Linux　本已　　当　定，　且　户　很大，无　开发windows　本，反　会带来兼容性

64.　个字　串　型　值　存储　大容　　多少？

- 512M

65. Redis如何做大　数据插入?

- Redis2.6开始redis-cli支持　　新　　之为pipe mode　新模式　于执　大　数据插入工作

66. 假如Redis　　1亿个key，其中　10w个key　以　个固定已　　前　开头　，如　将它们全　找出　?

- 使　keys指令可以扫出指定模式　key列
- 对方接　　：如果　个redis正在　　上　业务提供服务，　使　keys指令会有什么　　?　　个时候你　回　redis关　　个　性：redis　单　　keys指令会导　　塞　段时　，　上服务会停　，　到指令执　完毕，服务才　恢复　　个时候可以使　scan指令，scan指令可以无　塞　提取出指定模式　key列　，但是会有　定　　复概　，在客户　做　次去　就可以了，但是整体所　　时　会比　接　keys指令

67. 使　Redis做　异　　列吗，　如何实

- 使　list　型保存数据信息，rpush　产消息，lpop消　消息，当lpop没有消息时，可以sleep　段时　，然后再检查有没有信息，如果不想sleep　　，可以使　blpop, 在没有信息　时候，会　塞，　到信息　到来　redis可以　　pub/sub主　　模式实　　个　产　，多个消　　，当然也存在　定　　点，当消　　下　时，　产　消息会丢失

68. Redis如何实　延　　列

- 使　sortedset，使　时　戳做score, 消息内容作为key,　　zadd来　产消息，消　　使　zrangbyscore　取n　之前　数据做　　处

69. Redis回收　　如何工作　?

1. 　个客户　　了新　命令，添加了新　数据
2. Redis检查内存使　情况，如果大于maxmemory　　制，则根据　定好　　回收
3. 　个新　命令　执　，
4. 所以我们不断地　　内存　制　　，　　不断　到　　然后不断地回收回到　　以下

```
　一个　令　　使　（例　　交　保　　一个　　），不　　久　　会　个　　使　　。
```

70. Redis回收使　　什么　　?

- LRU　法