<https://www.cnblogs.com/liumt/p/6095357.html>

**一、Redis简介**

**1.什么是Redis**

  Redis 是一个开源的使用ANSI C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库，并提供多种语言的API。从2010 年3 月15 日起，Redis 的开发工作由VMware 主持。

  Redis 是一个Key-Value 存储系统。和Memcached 类似，它支持存储的value 类型相对更多， 包括string(字符串)、hash(散列)、list(链表)、set(集合)和zset(有序集合)。这些数据类型支持push/pop、add/remove 及取交集并集和差集和更丰富的操作，而且这些操作都是原子性的。在此基础上，Redis 支持各种不同方式的排序。与memcached 一样，为了保证效率，数据都是缓存在内存中。区别的是Redis 会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。

**2.Redis思想**

  Redis 的作者antirez（Salvatore Sanfilippo）曾经发表了一篇名为Redis 宣言（Redis Manifesto） 的文章，文中列举了Redis 的七个原则，以向大家阐明Redis的思想。

1、Redis 是一个操作数据结构的语言工具，它提供基于TCP 的协议以操作丰富的数据结构。在Redis 中，数据结构这个词的意义不仅表示在某种数据结构上的操作，更包括了结构本身及这些操作的时间空间复杂度。

2、Redis 定位于一个内存数据库，正是由于内存的快速访问特性，才使得Redis 能够有如此高的性能，才使得Redis 能够轻松处理大量复杂的数据结构，Redis 会尝试其它的存储方面的选择，但是永远不会改变它是一个内存数据库的角色。

3、Redis 使用基础的API 操作基础的数据结构，Redis 的API 与数据结构一样，都是一些最基础的元素，你几乎可以将任何信息交互使用此API 格式表示。作者调侃说，如果有其它非人类的智能生物存在，他们也能理解Redis 的API。因为它是如此的基础。

4、Redis 有着诗一般优美的代码，经常有一些不太了解Redis 有的人会建议Redis 采用一些其它人的代码，以实现一些Redis 未实现的功能，但这对我们来说就像是非要给《红楼梦》接上后四十回一样。

5、Redis 始终避免复杂化，我们认为设计一个系统的本质，就是与复杂化作战。我们不会为了一个小功能而往源码里添加上千行代码，解决复杂问题的方法就是让复杂问题永远不要提复杂的问题。

6、Redis 支持两个层面的API，第一个层面包含部分操作API，但它支持用于分布式环境下的Redis。第二个层面的API 支持更复杂的multi-key 操作。它们各有所长，但是我们不会推出两者都支持的API，但我们希望能够提供实例间数据迁移的命令，并执行multi-key 操作。

7、我们以优化代码为乐，我们相信编码是一件辛苦的工作，唯一对得起这辛苦的就是去享受它。如果我们在编码中失去了乐趣，那最好的解决办法就是停下来。我们决不会选择让Redis 不好玩的开发模式。

  Redis 的作者antirez 曾笑称Redis 为一个数据结构服务器（data structures server），其实可以认为这是一个非常准确的表述，Redis 的所有功能就是将数据以其固有的几种结构来保存，并提供给用户操作这几种结构的接口。本文将介绍Redis 支持的各种数据类型及其操作接口。

**3.Key-Value 存储系统简介**

  Key-Value Store 是当下比较流行的话题，尤其在构建诸如搜索引擎、IM、P2P、游戏服务器、SNS 等大型互联网应用以及提供云计算服务的时候，怎样保证系统在海量数据环境下的高性能、高可靠性、高扩展性、高可用性、低成本成为所有系统架构们挖苦心思考虑的重点，而怎样解决数据库服务器的性能瓶颈是最大的挑战。

  按照分布式领域的CAP 理论（Consistency、 Availability、Tolerance to network Partitions 这三部分在任何系统架构实现时只可能同时满足其中二点，没法三者兼顾）来衡量，传统的关系数据库的ACID 只满足了Consistency、Availability，因此在Partition tolerance 上就很难做得好。另外传统的关系数据库处理海量数据、分布式架构时候在Performance、Scalability、 Availability 等方面也存在很大的局限性。

  而Key-Value Store 更加注重对海量数据存取的性能、分布式、扩展性支持上，并不需要传统关系数据库的一些特征，例如：Schema、事务、完整SQL 查询支持等等，因此在分布式环境下的性能相对于传统的关系数据库有较大的提升。



图1.一些数据库和缓存服务器的特性与功能

**4.Redis 实际应用案例**

  目前全球最大的Redis 用户是新浪微博，在新浪有200 多台物理机，400 多个端口正在运行着Redis, 有+4G 的数据跑在Redis 上来为微博用户提供服务。



图2.新浪微博Redis应用

  在新浪微博Redis 的部署场景很多，大概分为如下的2 种:

  第一种是应用程序直接访问Redis 数据库

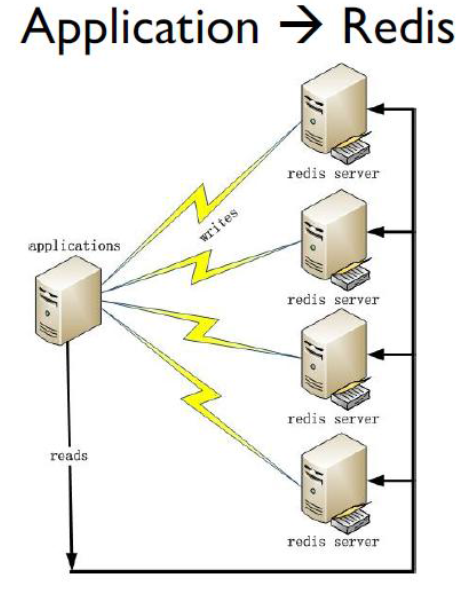


图3. 应用程序直接访问Redis 数据库

  第二种是应用程序直接访问Redis，只有当Redis 访问失败时才访问MySQL

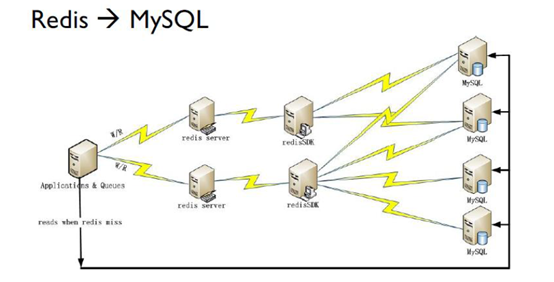


图4. 应用程序访问Redis和MySQL数据库

**二、初识Redis**

**1.数据类型**

  作为Key-value 型数据库，Redis 也提供了键（Key）和键值（Value）的映射关系。但是，除了常规的数值或字符串，Redis 的键值还可以是以下形式之一：

-Lists （列表）

-Hash （散列）

-Sets （集合）

-Sorted sets （有序集合）

-Hashes （哈希表）

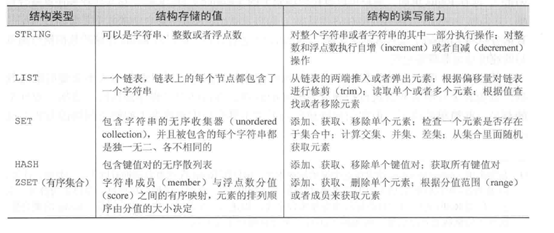


图5.Redis提供的五种结构

  键值的数据类型决定了该键值支持的操作。Redis 支持诸如列表、集合或有序集合的交集、并集、查集等高级原子操作；同时，如果键值的类型是普通数字，Redis 则提供自增等原子操作。

**2.持久化**

  通常，Redis 将数据存储于内存中，或被配置为使用虚拟内存。通过两种方式可以实现数据持久化：使用截图的方式，将内存中的数据不断写入磁盘；或使用类似MySQ-的日志方式， 记录每次更新的日志。前者性能较高，但是可能会引起一定程度的数据丢失；后者相反。

**3.主从同步**

  Redis 支持将数据同步到多台从库上，这种特性对提高读取性能非常有益。

**4.性能**

  相比需要依赖磁盘记录每个更新的数据库，基于内存的特性无疑给Redis 带来了非常优秀的性能。读写操作之间有显著的性能差异。

**5.提供API的语言**

-C

-C++

-C#

-Clojure

-Common Lisp

-Erlang

-Haskell

-Java

-Javascript

-Lua

-Objective-C

-Perl

-PHP

-Python

-Ruby

-Scala

-Go

-Tcl

**6.适用场合**

  毫无疑问，Redis 开创了一种新的数据存储思路，使用Redis，我们不用在面对功能单调的数据库时，把精力放在如何把大象放进冰箱这样的问题上，而是利用Redis 灵活多变的数据结构和数据操作，为不同的大象构建不同的冰箱。希望你喜欢这个比喻。

  下面是Redis 适用的一些场景:

**1、取最新N 个数据的操作**

  比如典型的取你网站的最新文章，通过下面方式，我们可以将最新的5000 条评论的ID 放在Redis 的List 集合中，并将超出集合部分从数据库获取。

  使用LPUSH latest.comments<ID>命令，向list 集合中插入数据

  插入完成后再用LTRIM latest.comments 0 5000 命令使其永远只保存最近5000 个ID。

  如果你还有不同的筛选维度，比如某个分类的最新N 条，那么你可以再建一个按此分类的List，只存ID 的话，Redis 是非常高效的。

**2、排行榜应用，取TOP N 操作**

  这个需求与上面需求的不同之处在于，前面操作以时间为权重，这个是以某个条件为权重， 比如按顶的次数排序，这时候就需要我们的sorted set 出马了，将你要排序的值设置成sorted set 的score，将具体的数据设置成相应的value，每次只需要执行一条ZADD 命令即可。

**3、需要精准设定过期时间的应用**

  比如你可以把上面说到的sorted set 的score 值设置成过期时间的时间戳，那么就可以简单地通过过期时间排序，定时清除过期数据了，不仅是清除Redis 中的过期数据，你完全可以把Redis 里这个过期时间当成是对数据库中数据的索引，用Redis 来找出哪些数据需要过期删除，然后再精准地从数据库中删除相应的记录。

**4、计数器应用**

  Redis 的命令都是原子性的，你可以轻松地利用INCR，DECR 命令来构建计数器系统。

**5、Uniq 操作，获取某段时间所有数据排重值**

  这个使用Redis 的set 数据结构最合适了，只需要不断地将数据往set 中扔就行了，set 意为集合，集合中的元素是唯一的，所以会自动排重。

**6、实时系统，反垃圾系统**

  通过上面说到的set 功能，你可以知道一个终端用户是否进行了某个操作，可以找到其操作的集合并进行分析统计对比等。没有做不到，只有想不到。

**7、Pub/Sub 构建实时消息系统**

  Redis 的Pub/Sub 系统可以构建实时的消息系统，比如很多用Pub/Sub 构建的实时聊天系统的例子。

**8、构建队列系统**

  使用list 可以构建队列系统，使用sorted set 甚至可以构建有优先级的队列系统。

**9、缓存**

  这个不必说了，性能优于Memcached，数据结构更多样化

**三、推荐一些Redis学习资源**

**1.网站**

[http://redis.io](http://redis.io/)                                                                 　　 Redis官网

<https://github.com/antirez/redis>                                       　  Redis源代码

<https://github.com/huangz1990/redis-3.0-annotated>       　　 注释版的Redis 3.0源码

[http://doc.redisfans.com](http://doc.redisfans.com/)                                                　　  Redis命令参考中文翻译版

**2.书籍**

《[redis入门指南（第2版](https://www.baidu.com/s?wd=redis%E5%85%A5%E9%97%A8%E6%8C%87%E5%8D%97%20%E7%AC%AC2%E7%89%88&rsp=0&f=1&oq=redis%E5%85%A5%E9%97%A8%E6%8C%87%E5%8D%97&tn=baiduhome_pg&ie=utf-8&usm=1&rsv_idx=2&rsv_pq=bef814c1000389ca&rsv_t=5849twvOSfIk1%2Fg8orC0qzIwBLBz4VR2zl4M3vXX%2FwHB8X%2BcfESaidh%2FzIbjkerhS%2BCG&rqlang=cn&rsv_ers=xn0&rs_src=0&rsv_pq=bef814c1000389ca&rsv_t=5849twvOSfIk1%2Fg8orC0qzIwBLBz4VR2zl4M3vXX%2FwHB8X%2BcfESaidh%2FzIbjkerhS%2BCG)）》     　　　　　　　　　　　　　　　 一本适合Redis入门的书籍

《Redis 设计与实现》              　　　　　　　　　　　　　　　剖析了Redis的源码

《Redis实战》                        　　　　　　　　　　　　　　　Redis在实战中的应用

  上面这三本书分别从入门、源码、 实战三个维度讲解了Redis，这也是我认为Redis非常值得学习的三个方面。redis的源码只有三万多行，很值得研究学习下，同时也方便自己团队在使用过程中去根据业务需求量身打造Redis。

参考资料：部分文字内容或插图来自于以上书籍或网络搜集

标签: [[大道至简]Nosql之Redis数据库](http://www.cnblogs.com/liumt/tag/%5B%E5%A4%A7%E9%81%93%E8%87%B3%E7%AE%80%5DNosql%E4%B9%8BRedis%E6%95%B0%E6%8D%AE%E5%BA%93/), [[淡泊明志]工作室技术部后台开发部长工作](http://www.cnblogs.com/liumt/tag/%5B%E6%B7%A1%E6%B3%8A%E6%98%8E%E5%BF%97%5D%E5%B7%A5%E4%BD%9C%E5%AE%A4%E6%8A%80%E6%9C%AF%E9%83%A8%E5%90%8E%E5%8F%B0%E5%BC%80%E5%8F%91%E9%83%A8%E9%95%BF%E5%B7%A5%E4%BD%9C/)