

大数据库系统

第3章 键值数据库Redis

第3章 键值数据库Redis

◆ 本章内容

3.1 Redis简介

3.2 Redis的安装、五大数据类型概述

3.3 Redis数据类型及操作

3.3.1 通用命令

3.3.2 STRING（字符串）相关命令与操作

3.3.3 LIST（列表）相关命令与操作

3.3.4 SET（无序集合）相关命令与操作

3.3.5 ZSET（有序集合）相关命令与操作

3.3.6 HASH（哈希表）相关命令与操作

3.4 Redis的事务及消息订阅功能

3.4.1 Redis事务功能

3.4.2 Redis消息订阅功能

3.5 Redis的持久化

3.5.1 RDB持久化

3.5.2 AOF持久化

3.6 Redis集群配置，组从复制

3.6.1 Redis集群简介及配置

3.6.2 Redis主从复制功能

3.7 Redis运维

3.7.1 Redis运维相关命令简介

3.7.2 Redis哨兵模式的配置及使用

3.1 Redis简介

◆ 本节内容

3.1.1 Redis起源

3.1.2 Redis是什么

3.1.3 Redis的特点

3.1.4 Redis的应用

3.1.1 Redis起源

◆ NoSQL的四大类型

| | |
|---|---|
| 文档数据库 | 图数据库 |
|    |   |
| 键值数据库 | 列族数据库 |
|     |      |

3.1.1 Redis起源



◆ 起源

- Redis是由意大利的一家创业公司Merzia的创始人Salvatore Sanfilippo于2009年开发的一款数据库
- Salvatore Sanfilippo早年是系统管理员，2004到2006年做嵌入式工作，之后接触web
- 2007年和朋友共同创建一个网站LLOOGG.com

3.1.1 Redis起源

◆ LLOOGG.com为什么需要redis

- LLOOGG.com是一个访客信息网站
- 随着网站的用户越来越愈多，需要维护的列表数量也越来越多，要执行的用户操作也越来越多
- LLOOGG使用的数据库是Mysql
 - 每次Mysql执行用户操作，都需要进行硬盘的读写操作
 - 程序的性能严重受制于硬盘的I/O
 - 很多操作都在排队
 - 最后导致网站崩溃

3.1.1 Redis起源

◆开发Redis

2009年开发了Redis数据库，旨在解决LLOOGG.com的负载问题

Salvatore Sanfilippo希望有更多人能够使用它，将其开源发布

继续开发Redis，并不断地完善至今

现在，使用Redis数据库的用户已经不计其数

[GET STARTED](#)[DOCS](#)[COMMANDS](#)[RESOURCES](#)[COMMUNITY](#)[SUPPORT](#)[Try Redis Cloud](#)[Download](#)

Redis

The open-source, **in-memory data store** used by millions of developers as a **cache**, **vector database**, **document database**, **streaming engine**, and **message broker**.

[Get Started](#)[Read the docs](#)

```
redis> PING
"PONG"
redis> HSET user:1 name antirez vocation artist
(integer) 2
redis> SET e 2.71
"OK"
redis> INCRBYFLOAT e 0.43
"3.14"
redis> RENAME e pi
"OK"
redis>
```

3.1.2 Redis是什么

◆ 什么是Redis?

Redis:REmote DIctionary Server

远程字典服务器

Introduction to Redis

Redis is an open source (BSD licensed), in-memory **data structure store**, used as database, cache and message broker. It supports data structures such as **strings**, **hashes**, **lists**, **sets**, **sorted sets** with range queries, **bitmaps**, **hyperloglogs** and **geospatial indexes** with radius queries. Redis has built-in **replication**, **Lua scripting**, **LRU eviction**, **transactions** and different levels of **on-disk persistence**, and provides high availability via **Redis Sentinel** and automatic partitioning with **Redis Cluster**.

3.1.3 Redis的特点

◆ 特点

a.支持多种类型的数据结构

总共8种数据结构

适应不同场景下的存储需求

3.1.3 Redis的特点

◆ 特点

b. 读/写速度快，性能高，纯粹为应用而产生

官方数据：

- 读的速度是110 000次/s
- 写的速度是81 000次/s

不同的服务器配置性能略有不同

c. 支持持久化

- 异步将内存中的数据写到硬盘上，同时不影响继续服务

3.1.3 Redis的特点

◆ 特点

d. 实现分布式集群和高可用

高可用主从复制

- 数据备份
- 读写分离
- 平衡集群中服务器的负荷
- 容灾
- Redis Sentinel（哨兵模式）自动监控服务器情况，支持高可用

3.1.3 Redis的特点

◆ 特点

e. 核心代码简单且功能强大

还包括其他功能，如消息订阅发布数据库事务等

功能不依赖外部库，它的所有操作都是原子性的，使用简单

3.1.3 Redis的特点

◆ 特点

f. 支持多种计算机编程语言

如Java、C、C++、Python、PHP、Lua、Ruby、Node.js、
C#、GoLand等

为多种计算机编程语言提供API

3.1.4 Redis的应用

1、做缓存

- 缓存速度和查询速度更快，使用更方便
- 实现数据查询、缓存新闻消息内容、缓存热门商品内容或购物车等

3.1.4 Redis的应用

2、做计数器/限速器应用

- Redis的命令具有原子性，可以使用INCR、DECR、GETSET、INCRBY等相关命令来构建计数器/限速器系统
- 用户点赞数、用户访问数、热门帖子的转发数、评论数、秒杀计数、搜索频次间隔限制、登录频次、短信发送频次等
- 限制某个用户访问某个API的频率，常用的有抢购时，防止用户疯狂点击带来不必要的压力
- 这类操作如果用MySQL，频繁的读写会带来相当大的压力

3.1.4 Redis的应用

3、实现消息队列、实时消息系统

- Redis拥有消息订阅发布、阻塞队列的命令，List数据结构具备队列和堆栈的功能，让一个程序在执行时被另一个程序添加到队列中
- 到货通知、邮件发送、简单的在线聊天系统
- 不需要高可靠，但是会带来非常大的DB压力
- 利用Redis的Pub/Sub命令构建消息系统

3.1.4 Redis的应用

4、实现排行榜应用

- Redis的有序集合数据类型给排行榜的实现提供了条件
- 对上百万个用户的排名，打分系统（评委的权值）采用其他数据库来实现是非常困难的
- 12306等超时未支付情况
- 利用Redis的ZADD、ZREVRANGE、ZRANK等命令可以轻松实现排名并获取排名的用户

3.1.4 Redis的应用

5、数据过期处理

- Redis支持对key进行生命周期设定，查询，修改
- Redis会自动通过key的生命周期属性找出过期的数据进行删除
- 可以查询生命周期属性来确认一个关键字在什么时候应该被删除。

3.1.4 Redis的应用

6、做大型社交网络

- 利用集合的一些命令，比如求交集、并集、差集等，方便完成共同好友、共同爱好等功能
- Redis可以很好地与社交网络相结合，如新浪微博、Twitter等
 - 共同好友，共同爱好，好友中有谁点赞了哪个微博，有谁评论了哪个微博
 - 聊天工具里未读消息通常也会放在缓存
 - 提示有新的帖子或者评论
- 使用QQ、微博等工具中，信息的刷新、浏览、查看、实时聊天需要Redis的支持

3.1.4 Redis的应用

◆ 其他应用

新闻消息系统

广告系统

实时向众多用户推送信息

实时显示最新的项目列表

统计网站的在线人数

帖子转发次数

实现游戏排名及其他相关排名等

3.1.4 Redis的应用

一个简单的实例：SNS的常用功能，获取共同好友

```
<?php
//phpredis实例  sns 中获取 共同好友的简单示例
header('content-type:text/html;charset=utf-8');
$redis = new Redis();
$redis->connect('10.10.26.117',6379);
//清空当前数据库
$redis->flushDB();
//张三朋友
$redis->sAdd('zhangsan', 'friend_1');
$redis->sAdd('zhangsan', 'friend_2');
$redis->sAdd('zhangsan', 'friend_3');
$redis->sAdd('zhangsan', 'friend_1');//此条语句会返回0，没有起作用
echo "张三的朋友\n";
var_dump($redis->sMembers('zhangsan'));
//李四朋友
$redis->sAdd('lisi', 'friend_1');
$redis->sAdd('lisi', 'friend_3');
$redis->sAdd('lisi', 'friend_5');
echo "李四的朋友\n";
var_dump($redis->sMembers('lisi'));
//取张三和李四的共同好友
echo "张三和李四共同的朋友\n";
var_dump($redis->sInter('zhangsan', 'lisi'));
```

```
张三的朋友
array(3) {
    [0]=>
        string(8) "friend_1"
    [1]=>
        string(8) "friend_2"
    [2]=>
        string(8) "friend_3"
}
李四的朋友
array(3) {
    [0]=>
        string(8) "friend_5"
    [1]=>
        string(8) "friend_1"
    [2]=>
        string(8) "friend_3"
}
张三和李四共同的朋友
array(2) {
    [0]=>
        string(8) "friend_1"
    [1]=>
        string(8) "friend_3"
}
```

3.1.4 Redis的应用

◆ 谁在使用Redis?



Instagram



twitter



.....

3.1.4 Redis的应用

◆ 新浪微博

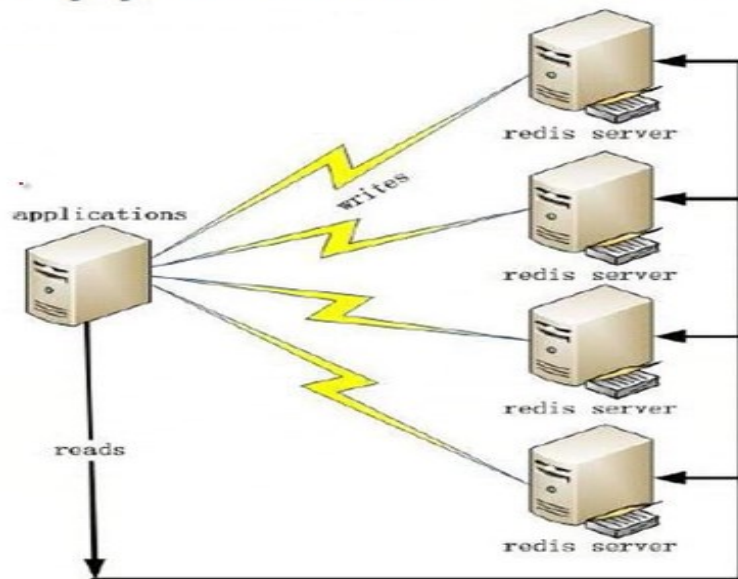
➤ 国内最大的Redis用户是新浪微博

- 新浪有200多台物理机
- 400多个端口正在运行Redis

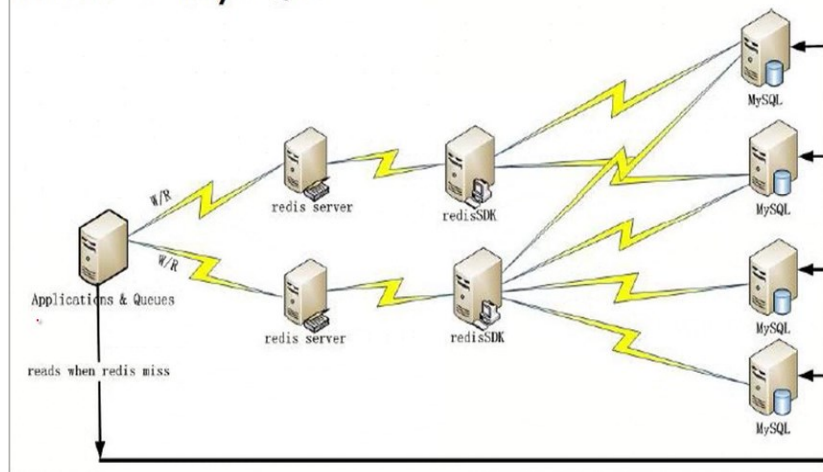
➤ 新浪微博Redis部署场景大致分为两种

- 1、应用程直接访问Redis数据库
- 2、应用程序直接访问Redis，只有当Redis访问失败时才访问数据库

Application → Redis



Redis → MySQL



总结

◆ 主要内容

3.1.1 Redis起源

3.1.2 Redis是什么

3.1.3 Redis的特点

3.1.4 Redis的应用

与Redis相关的几个网站

- ◆ Redis官网:

<http://redis.io>

- ◆ Redis中文官网:

<http://www.redis.cn/>

- ◆ Redis命令，包括用法，示例等

<http://redisdoc.com/>

◆ <https://www.redis.net.cn/order/>

[首页](#)[Redis教程](#)[Redis命令](#)[Redis客户端](#)[Redis下载](#)[Redis数据库](#)

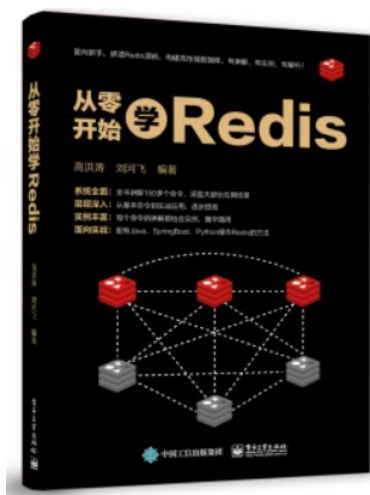
Redis 键(key) 命令

| 命令 | 描述 |
|------------------------------------|--|
| Redis Type 命令 | 返回 key 所储存的值的类型。 |
| Redis PEXPIREAT 命令 | 设置 key 的过期时间以毫秒计。 |
| Redis PEXPIREAT 命令 | 设置 key 过期时间的时间戳(unix timestamp) 以毫秒计 |
| Redis Rename 命令 | 修改 key 的名称 |
| Redis PERSIST 命令 | 移除 key 的过期时间, key 将持久保持。 |
| Redis Move 命令 | 将当前数据库的 key 移动到给定的数据库 db 当中。 |
| Redis RANDOMKEY 命令 | 从当前数据库中随机返回一个 key 。 |
| Redis Dump 命令 | 序列化给定 key , 并返回被序列化的值。 |
| Redis TTL 命令 | 以秒为单位, 返回给定 key 的剩余生存时间(TTL, time to live)。 |
| Redis Expire 命令 | seconds 为给定 key 设置过期时间。 |
| Redis DEL 命令 | 该命令用于在 key 存在是删除 key。 |
| Redis Pttl 命令 | 以毫秒为单位返回 key 的剩余的过期时间。 |
| Redis Renamenx 命令 | 仅当 newkey 不存在时, 将 key 改名为 newkey 。 |
| Redis EXISTS 命令 | 检查给定 key 是否存在。 |
| Redis Expireat 命令 | EXPIREAT 的作用和 EXPIRE 类似, 都用于为 key 设置过期时间。 不同在于 EXPIREAT 命令接受的时间参数是 UNIX 时间戳(unix timestamp)。 |
| Redis Keys 命令 | 查找所有符合给定模式(pattern)的 key 。 |

Redis 字符串(String) 命令

| 命令 | 描述 |
|-----------------------------------|------------------------|
| Redis Setnx 命令 | 只有在 key 不存在时设置 key 的值。 |
| Redis Getrange 命令 | 返回 key 中字符串值的子字符 |

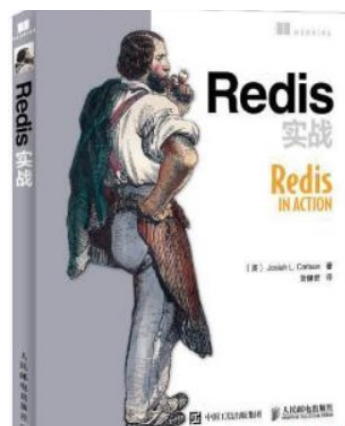
学习Redis推荐书籍



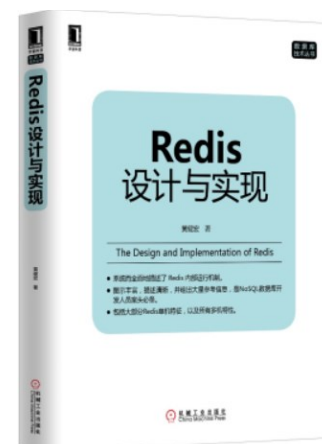
书名：从零开始学Redis
作者：高洪涛,刘河飞
出版社：电子工业出版社



书名：Redis使用手册
作者：黄健宏
出版社：机械工业出版社



书名：Redis实战
作者：约西亚L.卡尔森
(JosiahL.Carlson)
出版社：人民邮电出版社



书名：Redis设计与实现
作者：黄健宏
出版社：机械工业出版社

几点注意

◆ 注意

总共学习命令将近100个，涵盖了多数常用命令

理解的基础上，多动手，多写，多试

注意不要与SQL语言弄混了，Redis用的不是SQL语言

课时所限，学习的内容只是Redis的冰山一角，更多的命令、功能，比如Redis与其他编程工具的联合开发，更高级的运用，希望同学们课下自主可以多探索

3.1.3 Redis的特点

◆ Redis与Memcached等其他key-value内存缓存产品对比

1. Redis不仅仅支持key-value类型的数据，同时还提供list， set， sorted set， hash等数据结构存储， Memcached数据结构单一
2. Redis支持数据的主从复制，即master-slave模式的数据备份，读写分离，负载均衡， Memcached无数据冗余机制
3. Redis支持数据的持久化，可以将内存中的数据保存到磁盘中，重启的时候可以再次加载原有数据， Memcached断电数据丢失
4. Redis单个value的最大限制是1GB， Memcached单个value只能保存1MB的数据