Redis 资料

1. 背景介绍

1.1. 定义

Redis 是 **RE**mote **DI**ctionary **S**erver 的简称，它是一个用 C 语言开发的，开源的、高性能的、基于键值对的缓存与存储系统。

1.2. 历史与发展

意大利一家创业公司 Merzia 推出一款基于 MySQL 的网站实时统计系统 LLOOGG，它是一个访客信息追踪网站，如下图所示：

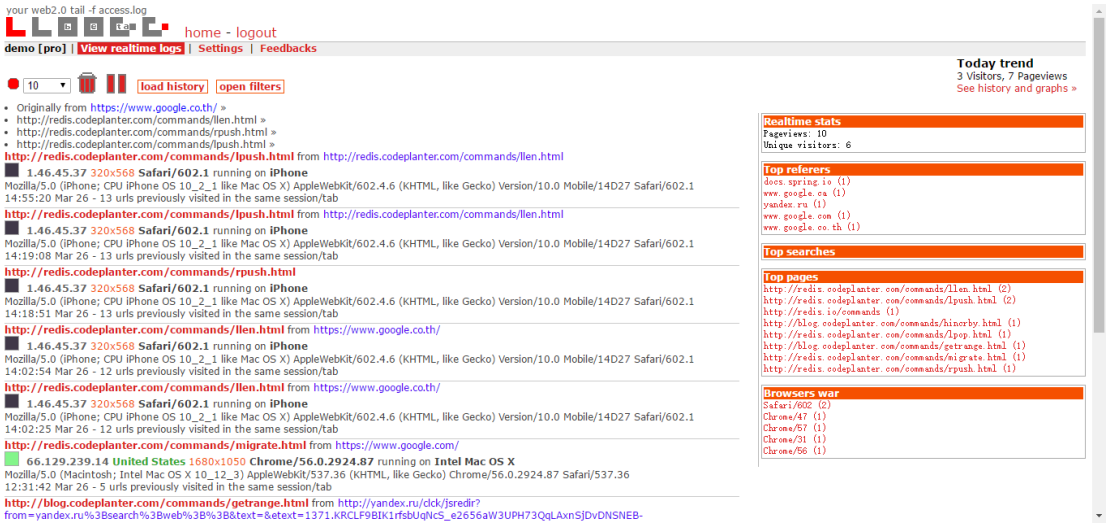

图 1 LLOOGG 网站界面

右侧是网站访问的排名状况。大量的访客信息统计使得网站达到了性能瓶颈。LLOOGG 的开发者，意大利人 Salvatore Sanfilippo（Antirez ）对 MySQL 的性能感到失望，于是他决定亲自为 LLOOGG 量身定做一个数据库。

图 2 Redis 之父——Antirez

这款数据库于 2009 年开发完成，命名为 Redis，并成功的应对了 LLOOGG 网站的性能问题。Antirez 不满足只将 Redis 应用于 LLOOGG，他将 Redis 开源发布，短短的几年时间，Redis 就拥有了庞大的用户群体。国内外很多大型互联网公司，如 GitHub、Stack Overflow、暴雪、Instagram、新浪微博、知乎等都是 Redis 的用户。

VMware 公司从 2010 年开始赞助 Redis 的开发，Antirez 于同年加入 VMware，全职开发 Redis。

1.3. 特性

Redis 究竟有哪些魅力？

1.3.1. 丰富的数据类型

支持五大数据类型：

- 字符串类型
- 散列类型

- 列表类型
- 集合类型
- 有序集合类型

1.3.2. 内存存储与持久化

- Redis 的所有数据存储在内存中，所以读写速度远快于基于硬盘的数据库。

  官方评估在一台普通笔记本电脑上，每秒可读写 10 万个键值对。

- Redis 也提供了持久化机制，可以将内存数据异步的吸入硬盘，而不影响提供服务。

1.3.3. 应用场景丰富

Redis 能做什么？

- **缓存：**通过提供键的生存时间机制实现了缓存系统的。
- **消息队列：**通过提供列表类型实现队列，并支持阻塞读取，可以容易实现一个高性能的优先级队列，并支持发布订阅的消息模式。
- **排行榜：**通过列表和有序集合，可以搭建各种排行榜系统。
- **会话管理：**分布式集群的共享会话可以保存在 Redis。

更多。。。

Redis 不能做什么？

- 数据量巨大的存储不适合 Redis。
- 冷数据不需要存放在 Redis。

1.3.4. 简单稳定

- 数据类型直观。
- 命令语句简单。

- 支持多种语言访问。
- 代码规模不大，开发人员吃透源代码，可定制化开发。
- Redis 的开发社区活跃，开发者众多。

Redis 的 GitHub 主页：https://github.com/antirez/redis

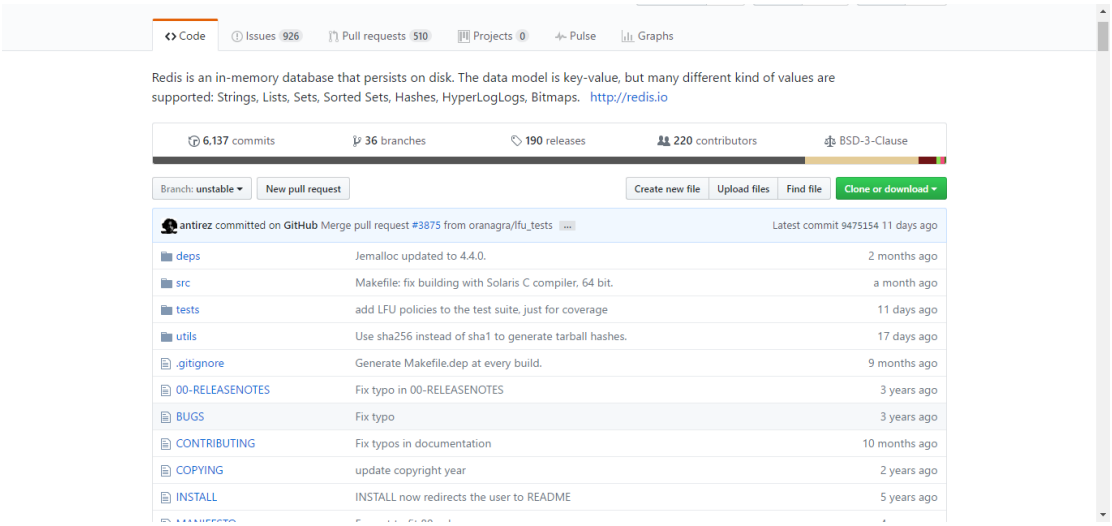

图 3 Redis 的 GitHub 更新状况

2. 安装部署

Redis 的官方主页：https://redis.io/，如下图：

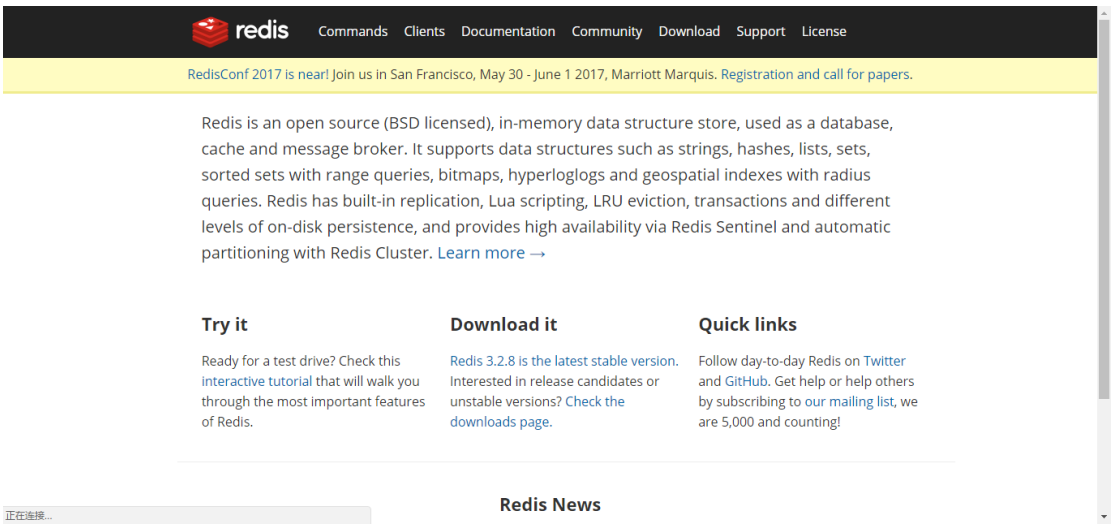

图 4 Redis 的官方主页

官方主页提供了命令参考、技术文档、下载地址等信息。

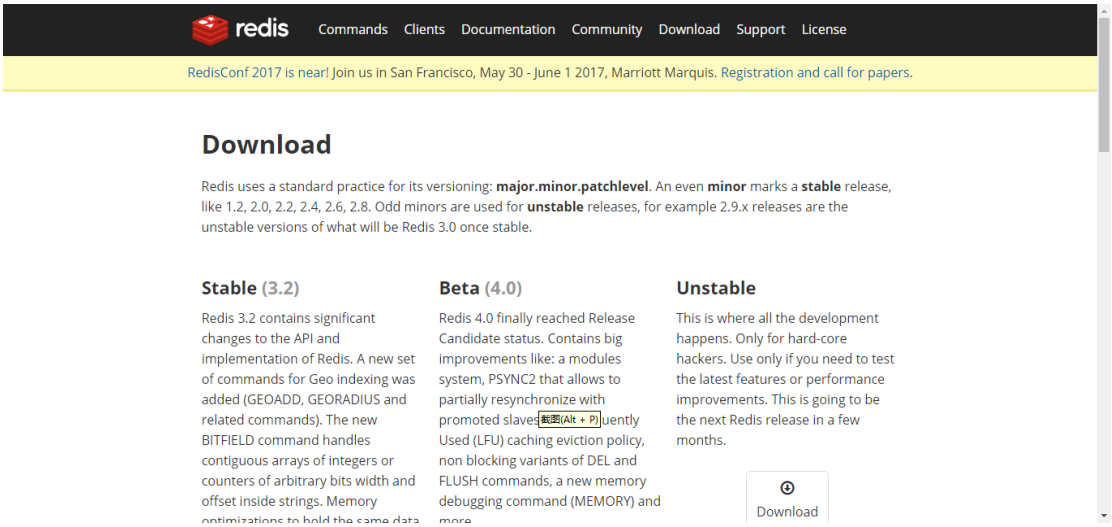

图 5 当前最新稳定版本是 Redis 3.2.8

http://download.redis.io/releases/redis-3.2.8.tar.gz

2.1. Posix 系统安装

在 Linux、OS X 和 BSD 等系统，直接通过下载源码，编译后安装 Redis。命令如下：

```
$ wget http://download.redis.io/releases/redis-3.2.8.tar.gz
$ tar xzf redis-3.2.8.tar.gz
$ cd redis-3.2.8
$ make
```

源代码解压后的目录如下图所示：

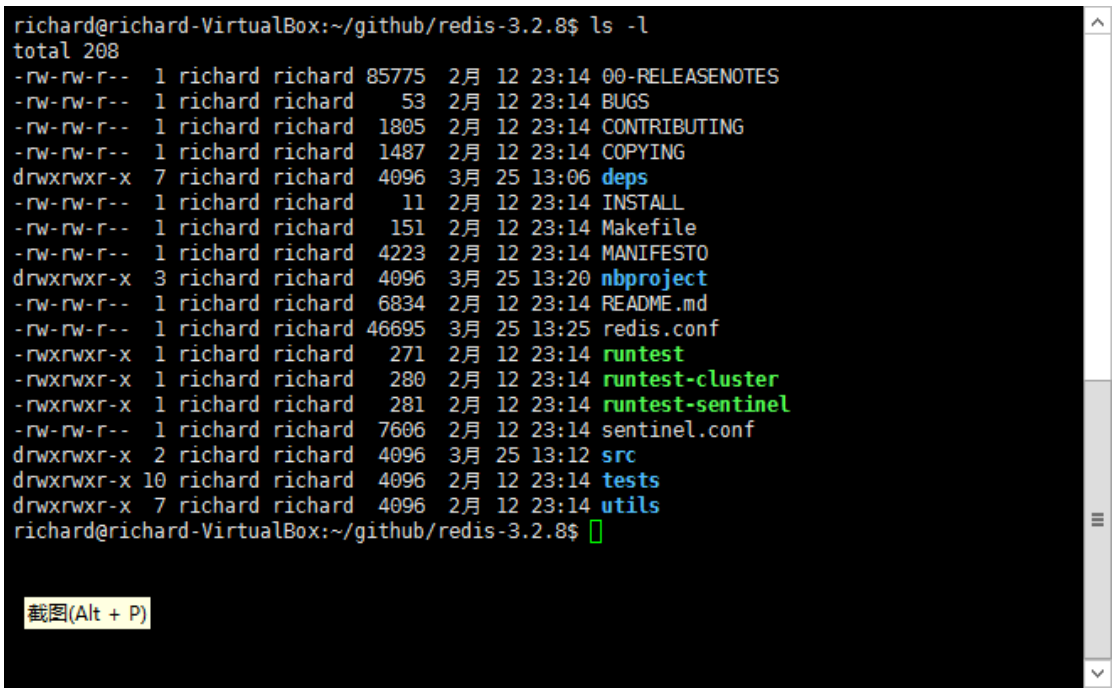

图 6 Redis 源码目录示意

make 完成之后，会在 src 目录下生成二进制文件，如下图：

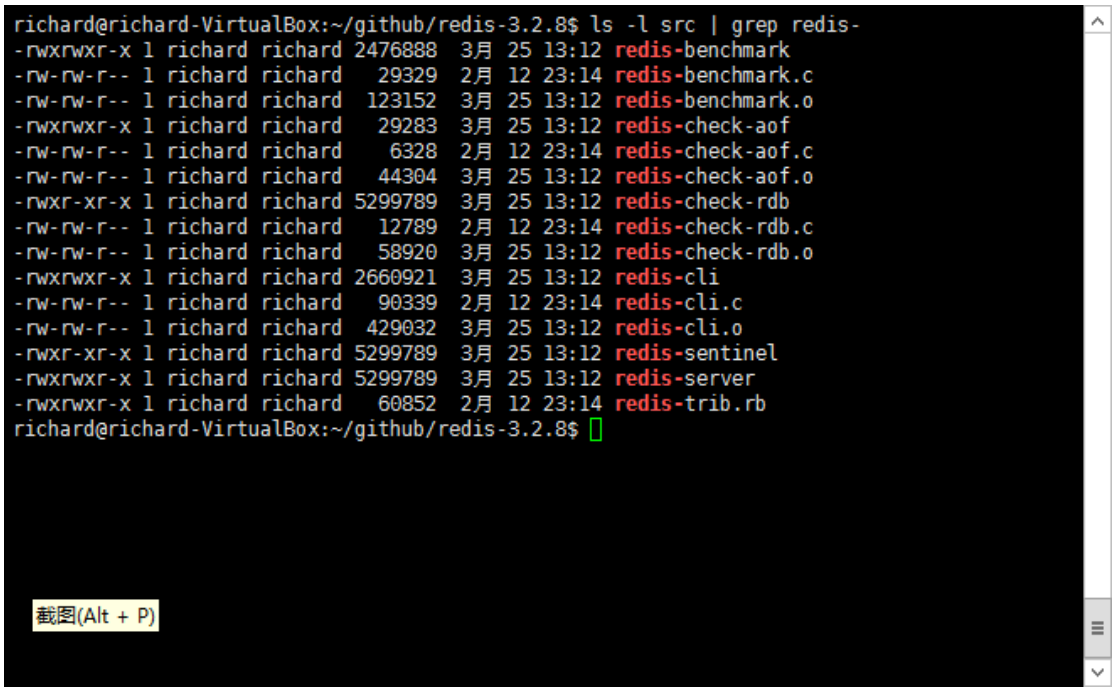

图 7 Redis 二进制文件生成示意

- redis-server：Redis 服务器端程序。
- redis-cli：Redis 客户端命令行程序。
- redis-sentinel：Redis 哨兵服务器程序。

- redis-benchmark：Redis 基准测试程序。
- redis-check-rdb：Redis 持久化 rdb 格式文件查看程序。
- redis-check-aof：Redis 持久化 aof 格式文件查看程序。

2.2. Windows 安装

Redis 官方不支持 Windows，后来微软在 GitHub 提交了一个 Windows 版的 Redis 分支，地址为：https://github.com/MSOpenTech/redis。可直接访问该站点，直接下载编译好的二进制安装文件，就可以在 Windows 上安装 Redis。

3. 使用介绍

3.1. 启动和停止

直接运行 redis-server 就可以启动 redis。

```
$    redis-server
```

图 8 redis 的启动打印输出

Redis 默认会使用 6379 端口。

停止 Redis 使用一下命令即可：

```
$    redis-cli -h 127.0.0.1 -p 6379 SHUTDOWN
```

这时候看到服务端的控制台已经退出程序，如下图：

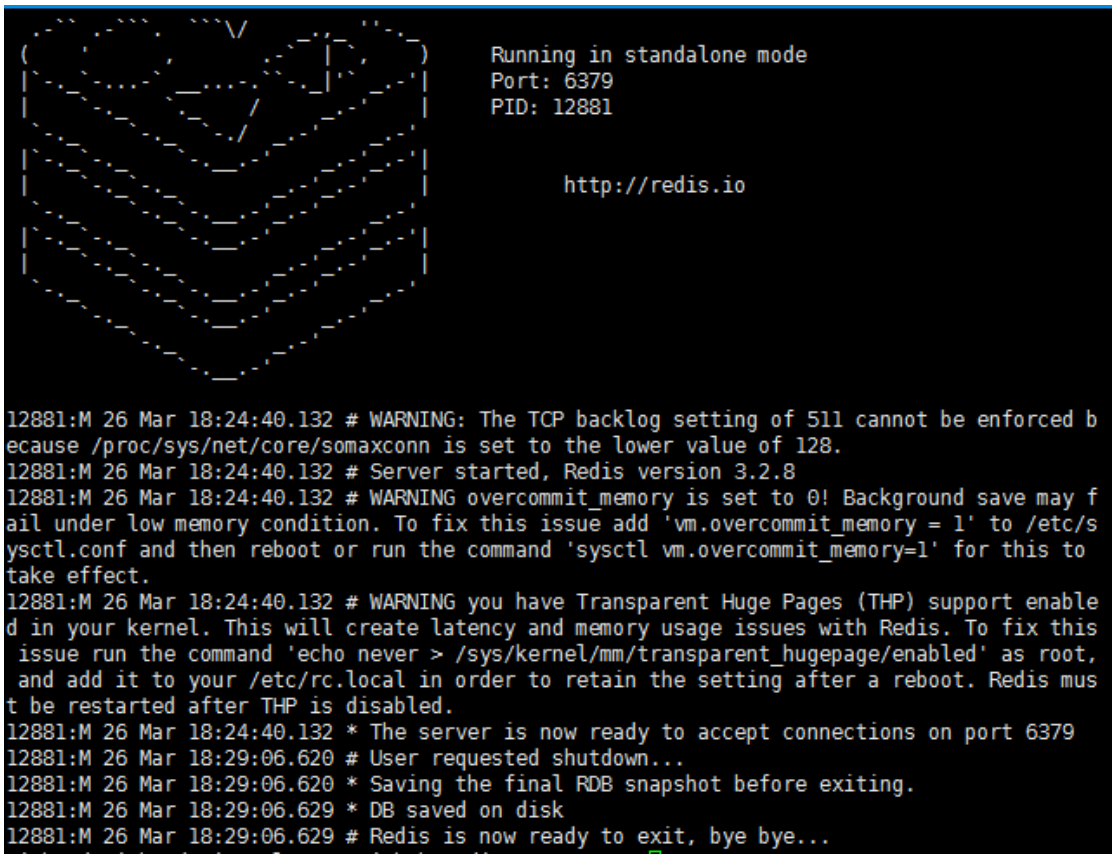

图 9 Redis 服务端退出

3.2. 命令行使用

3.2.1. 发送命令

Redis 客户端连接服务器的命令如下所示：

```
$ redis-cli -h ip -p port
```

其中 ip 是 Redis 服务器的 IP 地址，port 是 Redis 服务器的端口。通过 PING 命令可以测试连接是否正常，正常情况下，服务器会回复 PONG。

redis-cli 有两种发送命令方式，第一种是交互模式，如下图：

```
richard@richard-VirtualBox:~/github/redis-3.2.8/src$ ./redis-cli -h 127.0.0.1 -p 6379
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> quit
richard@richard-VirtualBox:~/github/redis-3.2.8/src$
```

图 10 redis-cli 的交互模式

交互模式下可以自由输入命令。第二种是单批命令模式，如下图：

```
richard@richard-VirtualBox:~/github/redis-3.2.8/src$ ./redis-cli -h 127.0.0.1 -p 6379 PING
PONG
richard@richard-VirtualBox:~/github/redis-3.2.8/src$
```

图 11 redis-cli 的单批命令模式

单批命令模式只能在连接命令之后附加命令，执行完断开连接并返回。

3.2.2. 命令返回值

- 状态回复

```
richard@richard-VirtualBox:~/github/redis-3.2.8/src$ ./redis-cli -h 127.0.0.1 -p 6379
127.0.0.1:6379> PING
PONG
127.0.0.1:6379> set test01 aa
OK
127.0.0.1:6379>
```

图 12 表示命令执行结果的状态回复

- 错误回复

```
127.0.0.1:6379> errorcommand
(error) ERR unknown command 'errorcommand'
127.0.0.1:6379>
```

图 13 命令不存在或命令格式有错的错误回复，一般以（error）开头

- 整数回复

```
127.0.0.1:6379> INCR foo
(integer) 1
127.0.0.1:6379>
```

图 14 命令执行结果是整数值的整数回复

- 字符串回复

```
127.0.0.1:6379> set msg "Hello Redis"
OK
127.0.0.1:6379> get msg
"Hello Redis"
127.0.0.1:6379>
```

图 15 命令结果返回单个字符串的字符串回复

- 多行字符串回复

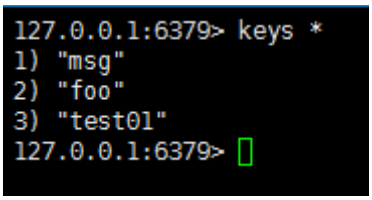

图 16 命令返回的是多个字符串的多行字符串回复

3.3. 配置 Redis

- 通过启动指定配置文件

Redis 服务端程序在启动的时候可以指定配置文件，这样可以修改默认的设置参数，命令如下所示;

```
$ redis-server /path/to/redis.conf
```

- 通过启动指定配置选项

也可以在启动的时候用配置选项来修改设置参数，如下所示；

```
$ redis-server /path/to/redis.conf –-loglevel warning
```

- 通过交互式命令配置

还可以在启动后，通过命令行程序的 CONFIG SET 命令动态修改配置参数，如下图所示：

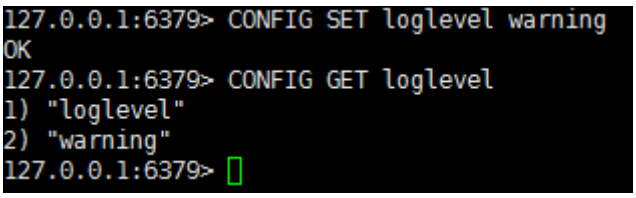

图 17 通过命令行动态修改 Redis 设置参数

由上图可知，利用 CONFIG GET 命令可以得到某个配置项的值。使用命令 CONFIG GET *，可以列出 CONFIG GET 命令支持的所有参数，如下图：

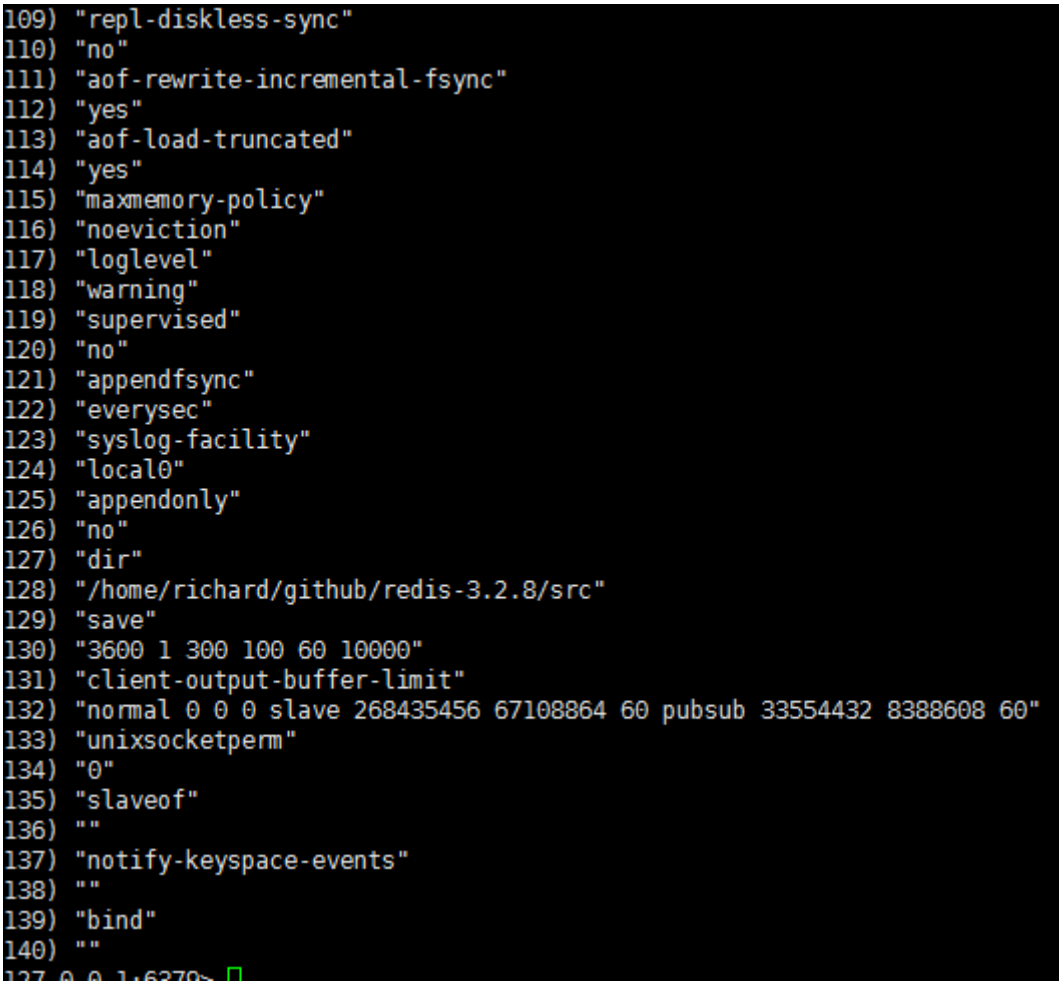

图 18 Redis 的配置参数

由上图可见，Redis 的配置参数多达上百个，可以参考官方文档来进行设置。

3.4. 多数据库支持

每个 Redis 实例包含了多个数据库，如下图所示：

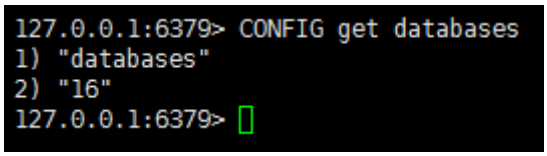

图 19 Redis 的数据库数

上图显示 Redis 实例包含 16 个数据库，每个数据库以编号命名，从 0 开始，可以通过 select 命令来切换数据库，如下图所示：

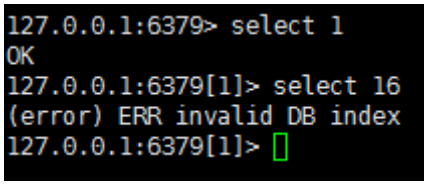

图 20 切换 Redis 数据库

由上图可见，select 不存在的数据库编号会返回错误回复。不同的数据库之间是不共享存储键命名空间，利用这种机制可以隔离不同的应用程序来访问 Redis 服务，如下图所示：

图 21 Redis 通过数据库来实现命名隔离

由上图可见，数据库 0 和数据库 1 的键值是不同的命名空间。

3.5. 数据类型及命令简介

3.5.1. Redis 键命名建议

由于 Redis 单个数据库内是没有命名隔离机制，所以建议命名如下：

object-type:object-id:object-field

object-type 代表业务实体类型，object-id 代表业务实体标识，object-field 代表业务实体属性，这三部分组合起来可以生成业务含义明确，命名不易冲突的键。

3.5.2. 键处理相关命令

- 获得符合规则的键名列表：**KEYS pattern**

  Pattern 支持以下通配符：

?	匹配一个字符
*	匹配任意多个字符
[ ]	匹配括号内任意一个字符，也可以用 X-Y 表示字符范围
\x	匹配字符 x，用于转义字符

- 判断一个键是否存在：**EXISTS key**

  如果键存在则返回 1，否则返回 0。

- 删除键：**DEL key1 [key2 …]**

  可以删除一个或多个键，返回值是删除的键的个数。

- 获得键值的数据类型：**TYPE key**

  TYPE 命令用来获得键值的数据类型，可能是 string（字符串）、hash（散列）、list（列表）、set（集合）、zset（有序集合）、none（键不存在）。

3.5.3. 字符串类型及命令

字符串类型是 Redis 的最基本的类型，它能存储任何形式的字符串，包括二进制数据，一个字符串类型的数据最大容量是 512MB。常见命令如下所示：

功能	命令示意
赋值	set key val
取值	get key
递增	incr num

✧ 应用场景举例：利用 incr 命令做文章访问量统计。

3.5.4. 散列类型及命令

散列类型是一种字典结构，存储了字段和字段值的映射，字段值只能是字符串，一个散列类型可以包含 2^{32}-1 个字段。散列相关的常见命令如下：

功能	命令示意
赋值	hset key field val
取值	hget key field
判断字段存在性	hexists key field
字段不存在时赋值	hsetnx key field val
增加字段值	hincrby key field incr
删除字段	hdel key field [field …]

✧ 应用场景举例：将标题和内容作为 key 和 val，缓存访问量较高的网页内容。

3.5.5. 列表类型及命令

列表类型是一个有序的字符串列表，两端都可以添加元素，还可以获得列表的一个片段，由于采用双向链表实现，所以增删元素较快，但是随机访问较慢，借助列表，Redis 可以实现队列功能。列表常见的命令如下所示：

功能	命令示意
左端增加元素	lpush key elem
右端增加元素	rpush key elem
左端弹出元素	lpop key
右端弹出元素	rpop key

获得列表元素个数	llen key

✧ 应用场景举例：缓存热门评论的有序列表。

3.5.6. 集合类型及命令

集合类型的特点是每个元素都是不同的，但是元素之间是没有顺序关系的。集合类型的相关命令如下所示：

功能	命令示意
增加元素	sadd key elem
删除元素	srem key elem
获得所有元素	smembers key
判断元素存在性	sismember key elem
集合求差	sdiff key [key …]
集合求交	sinter key [key …]
集合求并	sunion key [key …]

✧ 应用场景举例：缓存电商商品的标签。

3.5.7. 有序集合及命令

有序集合除了元素互不相同之外，还增加了元素的有序性，相关的命令如下所示：

功能	命令示意
增加元素	zadd key score elem
删除元素	zrem key elem
获得元素分数	zscore key elem
获得指定索引范围的元素	zrange key start stop

获得指定分数范围的元素	zrangebyscore key min max
增加某个元素的分数	zincrby key incr elem

✧ 应用场景举例：对在线视频根据点击量进行排序缓存。

3.5.8. 更多了解命令

由于 Redis 的命令比较多，网上已经有很多对命令详解的中文网站，如：

http://www.redis.cn/commands.html

http://redisdoc.com/

除此之外，如果不想搭建 Redis 环境，还想一试 Redis 命令的朋友，可以访问官方的 Redis 模拟学习环境，网站是：

http://try.redis.io/

这个网站还有配套交互式教程，体验非常友好，快试试吧！

4. 开发简介

Jedis 是 Redis 官方首选的 Java 客户端开发包。它的官方地址是：https://github.com/xetorthio/jedis。Jedis 包的 API 和 Redis 的命令可谓是一一对应，详细请查看官网的介绍。

5. 监控运维

Redis 的监控也有开源第三方工具，桌面版的工具有 Redis Desktop Manager，其官方网站是：https://redisdesktop.com/。界面如下图所示：

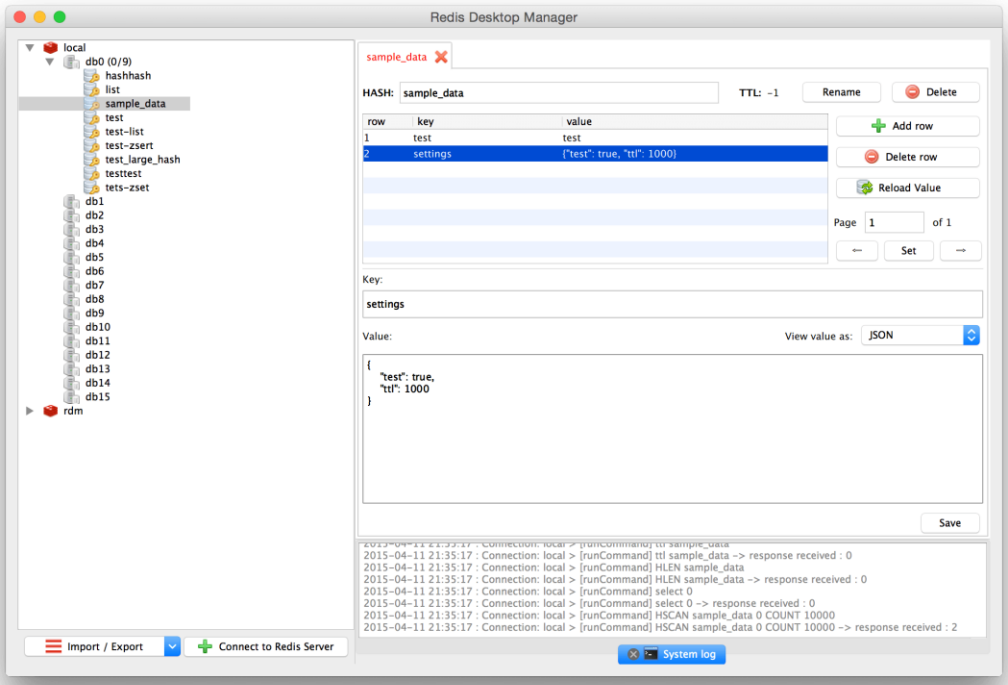

图 22 Redis Desktop Manager 界面

基于 Web 管理 Redis 的开源工具有 Redis Admin UI。其官方网站是：https://github.com/ServiceStackApps/RedisAdminUI/。该软件的界面如下所示：

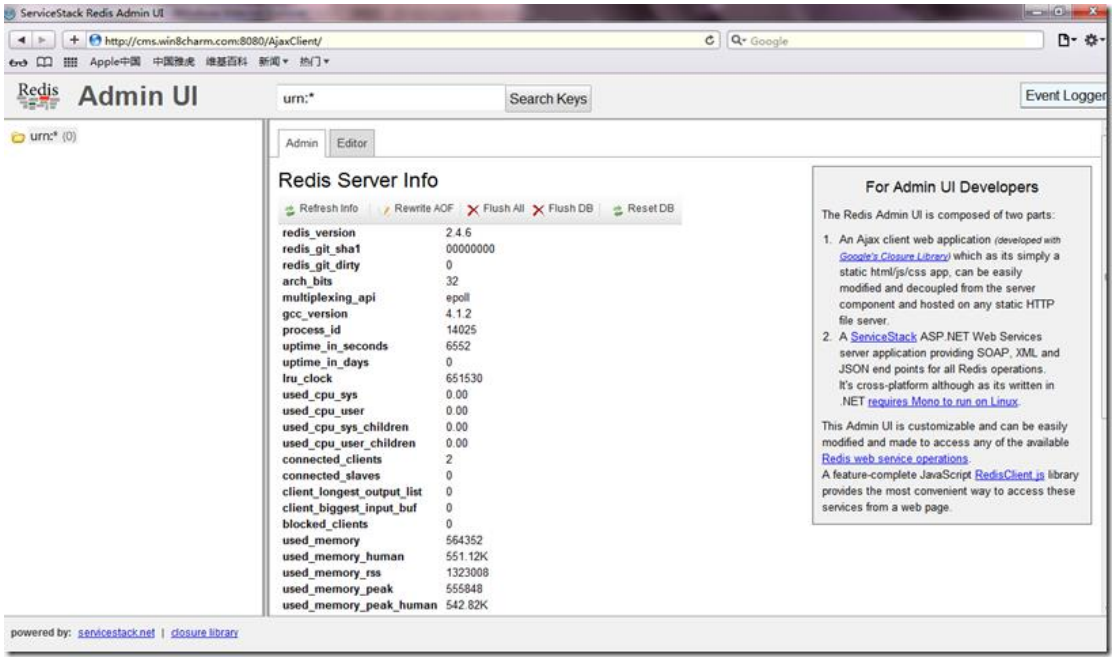

图 23 Redis Admin UI 界面

还可以直接在线体验该系统：http://redisreact.servicestack.net/#/。

还有一款基于 PHP 开发的 PHP Redis Admin。官方网站是：https://github.com/ErikDubbelboer/phpRedisAdmin。也有在线体验地址：https://dubbelboer.com/phpRedisAdmin/?overview。

6. 与 Memcached 的比较

6.1. Memcached 简介

Memcached 是一个高性能的分布式内存对象缓存系统，用于动态 Web 应用以减轻数据库负载。它通过在内存中缓存数据和对象来减少读取数据库的次数，从而提高动态、数据库驱动网站的速度。Memcached 基于一个存储键/值对的 Hash Map。其守护进程（daemon ）是用 C 写的，但是客户端可以用任何语言来编写，并通过 Memcached 协议与守护进程通信。它的特点包括：

- 协议简单
- 基于 libevent 的事件处理
- 内存存储
- 分布式

6.2. Redis 相比 Memcached 的特点

- Redis 相比 Memcached 来说，拥有更多的数据结构，支持更丰富的数据操作
- 使用简单的 key-value 存储的话，Memcached 的内存利用率更高，而如果 Redis 采用 hash 结构来做 key-value 存储，由于其组合式的压缩，其内存利用率会高于 Memcached。
- 由于 Redis 只使用单核，而 Memcached 可以使用多核，所以平均每一个

核上 Redis 在存储小数据时比 Memcached 性能更高。而在 100k 以上的数据中，Memcached 性能要高于 Redis。单 Redis 的单线程模型在源代码复杂度来说设计是简单了很多，易于学习和维护。

- Memcached 本身并不支持分布式，因此只能在客户端通过像一致性哈希这样的分布式算法来实现 Memcached 的分布式存储。相较于 Memcached 只能采用客户端实现分布式存储，Redis 更偏向于在服务器端构建分布式存储。

7.参考资料

https://redis.io/

http://redisdoc.com/

http://www.redis.cn/

《Redis 设计与实现》

《Redis 入门指南 第 2 版》

《Redis 开发与运维》

《Redis in action》