链滴

Redis 学习笔记 - 入门篇

作者：shengfvan

原文链接：https://ld246.com/article/1492937080015

来源网站：链滴

Redis 是什么？
Redis(REmote DIctionary Server 远程字典服务器)是一个开源的使用 ANSI C 语言编写的 、支网络、可基于内存亦可持久化的日志型、字典型结构的数据库(非关系型)。
解释一下定义，所谓的字典型，也就是我们常见的 Key-Value 结构。允许其他应用通过 TCP 协读写字典中的内容。Redis 数据库的所有数据都存储在内存中，因此读写速度异常迅速，每秒大约能行 10 万次左右的键值读写，对应的也有持久化到硬盘的功能。
Redis 的特性/优点：
1、存储结构
采用字典数据类型(Key-Value).
Redis 的目前支持的数据类型共有 5 种：

字符串类型(Strings)
散列类型(Hashes)
列表类型(Lists)
集合类型(Sets)
有序集合类型(Ordered Sets)

2、内存存储与持久化
Redis 数据库的所有数据都存储在内存中，由于内存相对于硬盘读写速度的差异，Redis 的读写度相对也有了非常明显的优势。断电或程序退出时，将所有数据持久化到硬盘中。
3、功能丰富
Redis 虽然是作为数据库开发的，但是更多的是用于缓存或队列。

用作缓存
Redis 可以设置 TTL(Time To Live 生存时间)，过期自动删除。也可以限制数据占用最大内存，超出根据设定规则删除不需要的 Key-Value。

用作队列
支持阻塞式读取，也支持发布/订阅消息模式。

4、简单稳定

结构简单
Key-Value 形式。

交互简单
采用命令读写数据,如

<blockquote>
#添加了一个 Key 为 hello，Value 为 sheng 的键值对
set hello sheng
#读取该键值对
get sheng
</blockquote>
<pre><code class="highlight-chroma">对比关系型数据库的SQL语句，确实方便许多。
</code></pre>

操作具有原子性
所有 Redis 操作都是原子操作，这确保如果两个客户端并发访问，Redis 服务器能接收更新的值。</l
>

下载和安装
Redis 官方是不支持 windows 的，但是 Microsoft Open Tech group 在 GitHub 上开发了一个
Win 64 位系统上可以运行的分支版本,项目地址是：<a href="https://ld246.com/forward?goto=ht
ps%3A%2F%2Fgithub.com%2FMSOpenTech%2Fredis" target="_blank" rel="nofollow ugc">ht
ps://github.com/MSOpenTech/redis。
笔者是在 win7 下学习的，具体的安装过程请参考 <a href="https://ld246.com/forward?goto
http%3A%2F%2Fblog.csdn.net%2Frenfufei%2Farticle%2Fdetails%2F38474435" target="_blank
 rel="nofollow ugc">http://blog.csdn.net/renfufei/article/details/38474435，下载 .zip
件，解压后有如下可执行文件，直接可用。
<blockquote>
redis-benchmark.exe #基准测试
redis-check-aof.exe #aof 文件修复工具
redis-check-dump.exe #rdb 文件检查工具
redis-cli.exe #客户端
redis-server.exe #服务器
redis.windows.conf #配置文件
</blockquote>
Linux 下请参考：<a href="https://ld246.com/forward?goto=https%3A%2F%2Fredis.io%2
download" target="_blank" rel="nofollow ugc">https://redis.io/download
入门
安装成功后，接下来就开始尝试 Redis 的使用吧。
1、启动和停止
A、直接到解压的目录下双击 redis-server.exe 启动
B、笔者建议在 CMD 上采用指令操作。
<blockquote>
cd d:/soft/redis3.2
redis-server
</blockquote>
这样 redis 服务器就启动了，redis 还提供了 redis-cli.exe 命令行客户端。如果服务启动了，打
此客户端上就会显示现在的服务器地址和端口。
redis 提供了测试连接的命令:ping，如果连接成功，则会返回 pong
<blockquote>
127.0.0.1:6279>ping
pong
</blockquote>
redis 的默认端口号是 6379，要修改端口号命令：
<blockquote>
redis-server --port 8080
</blockquote>
端口修改后，客户端访问命令：
<blockquote>
redis-cli -p 8080
</blockquote>
redis 默认是没有访问密码的，要设置 redis 密码访问，命令如下：
<blockquote>
#将 redis 的访问密码设为 123456
redis-server --requirepass 123456
</blockquote>
修改密码后，客户端访问命令：
<blockquote>

redis-cli -a 123456
</blockquote>
如果密码错误，客户端上还是显示正确的服务器地址和端口，但是访问时就会收限制：
<blockquote>
redis-cli -a 123456
(error)NOAUTH Authentication required.
</blockquote>
此外，还有另外一种方式。我们可以发现.zip 文件解压出来的文件中有一个 redis.windows.conf 文件，启动 redis 时，也可以采用 redis.windows.conf 中的配置来启动，端口，访问密码可以在文中修改，然后通过如下命令启动：
<blockquote>
redis-server redis.windows.conf
</blockquote>
当然指定了配置文件后，我们也还是可以通过命令在当次服务启动时覆盖配置文件的配置，但不修改配置文件的值，如：
<blockquote>
#redis 的日志等级由低到高是 debug>verbose>notice>warning ，默认的日志等级是 notice，
redis-server redis.windows.conf --loglevel warning
</blockquote>
停止 redis 服务，只要在命令客户端中输入如下命令：
<blockquote>
127.0.0.1:6279>shutdown
</blockquote>
此后我们的操作都通过 redis-cli.exe 命令行客户端来执行。
2、配置
上面提到了 redis 启动时可以在命令中指定配置，也可以使用配置文件中的配置。
在命令客户端中，还有一种命令可以修改 redis 配置，且无需重启 redis 服务器。命令如下：</
>
<blockquote>
127.0.0.1:6279>config set loglevel warning
</blockquote>
查询配置信息的命令如下：
<blockquote>
127.0.0.1:6279>config get configname
</blockquote>
3、多数据库
redis 实际上默认提供了 16 个数据库来存储字典结构的数据，编号由 0-15，默认使用 0.redis 支持自定义数据库名字，也不支持为每个数据库设置不同的密码。这些数据库并不是完全隔离的，如 flushall 命令可以清空所有数据库的数据。
因此不建议一个 redis 数据库实例用于不同的应用。
如下命令可以选择数据库：
<blockquote>
127.0.0.1:6279>select 1
127.0.0.1:6279>get hello
(nil)
127.0.0.1:6279>select 0
127.0.0.1:6279>get hello
"sheng"
</blockquote>
4、基础指令
1、查询当前库中所有 key。key 支持 glob 风格的通配符匹配。
<blockquote>
<pre><code class="highlight-chroma"><span class="highlight

cl">?:匹配一个字符
*:匹配任意个(包括个)字符
[]:匹配区间内的任字符
\x:转译字符，如要配?则使用\?
</code></pre>
</blockquote>
<blockquote>
127.0.0.1:6279>keys pattern
</blockquote>
2、判断 key 是否存在，存在返回 1，不存在返回 0.
<blockquote>
127.0.0.1:6279>exists key
</blockquote>
3、删除 key
<blockquote>
127.0.0.1:6279>del key
</blockquote>
4、获得 key 的类型，返回的值可能为：string(字符串)，list(列表)，hash(散列)，set(集合)，zse(有序集合)
<blockquote>
127.0.0.1:6279>type key
</blockquote>
5、字符串类型
字符串类型是 redis 中最基本的类型，能存储任何形式的字符串，包括二进制，json 甚至图片。个字符串类型的最大容量是 512M。
A、赋值和取值【set key/get key】
<blockquote>
#添加一个字符串类型键值对
</blockquote>
<blockquote>
127.0.0.1:6279>set key val
</blockquote>
<blockquote>
ok
</blockquote>
<blockquote>
#根据 key 读取键值
</blockquote>
<blockquote>
127.0.0.1:6279>get key
</blockquote>
<blockquote>
val
</blockquote>
B、数字递增/递减【incr key/decr key】
如果字符串是整数形式，redis 提供了指定键值递增/递减的命令：
<blockquote>
#如果 key 不存在，则默认值为 0,然后执行递增操作，返回的值为 1
</blockquote>
<blockquote>
127.0.0.1:6279>incr key

</blockquote>
<blockquote>
(integer)1
</blockquote>
<blockquote>
#如果值不是整数，则提示错误
</blockquote>
<blockquote>
127.0.0.1:6279>set key test
</blockquote>
<blockquote>
127.0.0.1:6279>incr key
</blockquote>
<blockquote>
(error) ERR value is not an integer or out of range
</blockquote>
这里值得一提的是，因为 redis 的原子性，如果有多个应用同时对一个 key 执行递增操作，则会自自增一次，两个应用得到的分别是 1 和 2.
C、整数加法/减法【incrby key num/ decr key num】
<blockquote>
127.0.0.1:6279>set num 1
</blockquote>
<blockquote>
127.0.0.1:6279>incrby num 3
</blockquote>
<blockquote>
(integer) 4
</blockquote>
D、浮点数加法/减法【incrbyfloat key float/decrbyfloat key float】
<blockquote>
127.0.0.1:6279>set num 1
</blockquote>
<blockquote>
127.0.0.1:6279>incrbyfloat num 2.22
</blockquote>
<blockquote>
"3.22"
</blockquote>
E、字符串拼接【append key val】
<blockquote>
127.0.0.1:6279>set string a
</blockquote>
<blockquote>
127.0.0.1:6279>append string ppend
</blockquote>
<blockquote>
"append"
</blockquote>
<blockquote>
#拼接多个单词或需要空格隔开的，需要加上引号
</blockquote>
<blockquote>
127.0.0.1:6279>append string " with many words"
</blockquote>

<blockquote>
"append with many words"
</blockquote>
F、获取字符串长度【strlen key】
<blockquote>
127.0.0.1:6279>set string sheng
</blockquote>
<blockquote>
(integer) 5
</blockquote>
<blockquote>
127.0.0.1:6279>set string 盛
</blockquote>
<blockquote>
#redis 中中文用 utf-8 编码，一个中文的长度是 3 字节
</blockquote>
<blockquote>
(integer) 2
</blockquote>
对于中文读写，可能会遇到显示的是如下的 16 进制字符串：
<blockquote>
127.0.0.1:6279>set key 你好
</blockquote>
<blockquote>
127.0.0.1:6279>get key
</blockquote>
<blockquote>
"\xc4\xe3\xba\xc3"
</blockquote>
遇到这种情况，可使用如下命令，解决中文字符的问题：
<blockquote>
redis-cli --raw
</blockquote>
G、批量读取/批量赋值【mget key1 key2 .../mset key1 val1 key2 val2...】
<blockquote>
127.0.0.1:6279>mset one 1 two 2 three 3
</blockquote>
<blockquote>
ok
</blockquote>
<blockquote>
127.0.0.1:6279>mget one two
</blockquote>
<blockquote>
1
</blockquote>
<blockquote>
2
</blockquote>
H、位操作
redis 中提供了 4 个命令用于二进制位操作。redis 存储的是 ASCII 二进制码。
<blockquote>
#获取指定二进制位的值(0/1)，如果指定的位数超过 key 值的二进制总长度，则返回 0
</blockquote>

<blockquote>
getbit key offset
</blockquote>
<blockquote>
#设置指定二进制位的值，返回修改位的旧值
</blockquote>
<blockquote>
setbit key offset val
</blockquote>
<blockquote>
#获取指定字节范围的 1 的数量
</blockquote>
<blockquote>
bitcount key begin end
</blockquote>
<blockquote>
#二进制与、或、非、异或操作
</blockquote>
<blockquote>
bitop and/or/not/xor result key1 key2
</blockquote>
下面我们来看几个例子：
<blockquote>
#a 的 ASCII 码为 97，二进制码为 01100001，b 的 ASCII 码为 98，二进制码为 01100010</p

</blockquote>
<blockquote>
127.0.0.1:6279>mset key1 a key2 b
</blockquote>
<blockquote>
127.0.0.1:6279>getbit key1
</blockquote>
<blockquote>
(integer) 3
</blockquote>
<blockquote>
#修改 a 的左起第七位，将 0 改为 1，即将 a 改为 c
</blockquote>
<blockquote>
127.0.0.1:6279>setbit key1 6 1
</blockquote>
<blockquote>
(integer) 0
</blockquote>
<blockquote>
127.0.0.1:6279>get key1
</blockquote>
<blockquote>
"c"
</blockquote>
<blockquote>
127.0.0.1:6279>bitcount key1
</blockquote>
<blockquote>

(integer) 4
</blockquote>
<blockquote>
#对 c 和 b 的二进制进行与操作，结果(二进制结果为 01100010)存在 key0 中
</blockquote>
<blockquote>
127.0.0.1:6279>bitop and key0 key1 key2
</blockquote>
<blockquote>
127.0.0.1:6279>get key0
</blockquote>
<blockquote>
"b"
</blockquote>
6、散列操作
散列适合用于存储对象，一个 key 下可以设有多个字段和对应的字段值。需要注意的是散列的字只能是字符串类型。
一个散列类型的 key 最多可以有 2E32-1 个字段。
A、赋值与取值【hset key field value/hget get field】
<blockquote>
#当 key 中没有字段时，赋值返回值为 1，如果存在，则更新字段值，返回 0
</blockquote>
<blockquote>
127.0.0.1:6279>hset key car color white
</blockquote>
<blockquote>
(integer) 1
</blockquote>
<blockquote>
127.0.0.1:6279>hset key car color black
</blockquote>
<blockquote>
(integer) 0
</blockquote>
B、批量赋值与取值【hmset key field1 val1 field1 val2/hmget key field1 field2/hgetall key
<blockquote>
127.0.0.1:6279>hmset key car color white brand ford
</blockquote>
<blockquote>
ok
</blockquote>
<blockquote>
127.0.0.1:6279>hget key car color brand
</blockquote>
<blockquote>
"color"
</blockquote>
<blockquote>
"black"
</blockquote>
<blockquote>
"brand"
</blockquote>

<blockquote>
"ford"
#查询散列健的所有字段值
127.0.0.1:6279>hgetall car
</blockquote>
<blockquote>
"color"
</blockquote>
<blockquote>
"black"
</blockquote>
<blockquote>
"brand"
</blockquote>
<blockquote>
"ford"
</blockquote>
持续更新中.....