# **[Redis 起步](http://www.cnblogs.com/shanyou/archive/2012/01/28/2330451.html)**

Rdis和JQuery一样是纯粹为应用而产生的，这里记录的是在CentOS 5.7上学习入门文章：

##### **1.Redis简介**

[Redis](http://code.google.com/p/redis/)是一个key-value存储系统。和Memcached类似，但是解决了断电后数据完全丢失的情况，而且她支持更多无化的value类型，除了和string外，还支持lists（链表）、sets（集合）和zsets（有序集合）几种数据类型。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。



##### **2.Redis的性能**

下面是官方的bench-mark数据：

* The test was done with 50 simultaneous clients performing 100000 requests.
* The value SET and GET is a 256 bytes string.
* The Linux box is running **Linux 2.6**, it’s **Xeon X3320 2.5Ghz**.
* Text executed using the loopback interface (127.0.0.1).

Results: **about 110000 SETs per second, about 81000 GETs per second.**

更多详细数据请见官方bench-mark page（<http://code.google.com/p/redis/wiki/Benchmarks>）

##### ****3.安装Redis****

Redis的代码遵循ANSI-C编写，可以在所有POSIX系统（如Linux, \*BSD, Mac OS X, Solaris等）上安装运行。而且Redis并不依赖任何非标准库，也没有编译参数必需添加。redis的安装出奇的简单，这可能也是他风靡的一个原因，让人很容易上手，不像某些东西，编译阶段就能让人完全绝望。

先去官网下载源码：

wget <http://redis.googlecode.com/files/redis-2.4.6.tar.gz>

解压：

tar –zxvf redis-2.4.6.tar.gz

编译

需要说明的事，redis的安装非常简单，已经有现成的Makefile文件，直接运行make命令即可。

make

make install

Redis 由四个可执行文件：**redis-benchmark**、**redis-cli**、**redis-server**、**redis-stat** 这四个文件，加上一个**redis.conf**就构成了整个redis的最终可用包。它们的作用如下：

* redis-server：Redis服务器的daemon启动程序
* redis-cli：Redis命令行操作工具。当然，你也可以用telnet根据其纯文本协议来操作
* redis-benchmark：Redis性能测试工具，测试Redis在你的系统及你的配置下的读写性能
* redis-stat：Redis状态检测工具，可以检测Redis当前状态参数及延迟状况

现在就可以启动redis了，redis只有一个启动参数，就是他的配置文件路径。

redis-server /etc/redis.conf

注意，默认复制过去的**redis.conf**文件的**daemonize**参数为**no**，所以redis不会在后台运行，这时要测试，我们需要重新开一个终端。修改为**yes**则为后台运行redis。另外配置文件中规定了pid文件，log文件和数据文件的地址，如果有需要先修改，默认log信息定向到stdout.

下面是redis.conf的主要配置参数的意义：

* daemonize：是否以后台daemon方式运行
* pidfile：pid文件位置
* port：监听的端口号
* timeout：请求超时时间
* loglevel：log信息级别
* logfile：log文件位置
* databases：开启数据库的数量
* save \* \*：保存快照的频率，第一个\*表示多长时间，第三个\*表示执行多少次写操作。在一定时间内执行一定数量的写操作时，自动保存快照。可设置多个条件。
* rdbcompression：是否使用压缩
* dbfilename：数据快照文件名（只是文件名，不包括目录）
* dir：数据快照的保存目录（这个是目录）
* appendonly：是否开启appendonlylog，开启的话每次写操作会记一条log，这会提高数据抗风险能力，但影响效率。
* appendfsync：appendonlylog如何同步到磁盘（三个选项，分别是每次写都强制调用fsync、每秒启用一次fsync、不调用fsync等待系统自己同步）

这时你可以打开一个终端进行测试了，配置文件中默认的监听端口是**6379**

我们可以开启一个Redis客户端进行测试

[root@SNDA-192-168-1-114 ~]# redis-cli   
Could not connect to Redis at 127.0.0.1:6379: Connection refused   
not connected> exit   
[root@SNDA-192-168-1-114 ~]# redis-server /etc/redis.conf   
[root@SNDA-192-168-1-114 ~]# redis-cli   
redis 127.0.0.1:6379> quit

##### ****4.redis数据结构****

redis 的作者antirez曾称其为一个数据结构服务器（**data structures server**），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保存，并提供给用户操作这几种结构的接口。我们可以想象我们在各种语言中的那些固有数据类型及其操作。

redis目前提供四种数据类型：**string**,**list**,**set**及**zset**(sorted set)和**Hash**。

* **string**是最简单的类型，你可以理解成与Memcached一模一个的类型，一个key对应一个value，其上支持的操作与Memcached的操作类似。但它的功能更丰富。
* **list**是一个链表结构，主要功能是push、pop、获取一个范围的所有值等等。操作中key理解为链表的名字。
* **set**是集合，和我们数学中的集合概念相似，对集合的操作有添加删除元素，有对多个集合求交并差等操作。操作中key理解为集合的名字。
* **zset**是set的一个升级版本，他在set的基础上增加了一个顺序属性，这一属性在添加修改元素的时候可以指定，每次指定后，zset会自动重新按新的值调整顺序。可以理解了有两列的mysql表，一列存value，一列存顺序。操作中key理解为zset的名字。
* **Hash**数据类型允许用户用Redis存储对象类型,Hash数据类型的一个重要优点是,当你存储的数据对象只有很少几个key值时,数据存储的内存消耗会很小.更多关于Hash数据类型的说明请见: <http://code.google.com/p/redis/wiki/Hashes>

在官网上给出了所有支持的接口列表，并副副附有详细的介绍，地址：

<http://code.google.com/p/redis/wiki/CommandReference>

另外，作者还提供了一个非常贴心的web命令行模拟页面，供初学者试用redis，地址：

<http://try.redis-db.com/>

##### ****5.redis数据存储****

redis的存储分为内存存储、磁盘存储和log文件三部分，配置文件中有三个参数对其进行配置。

**save seconds updates**，**save**配置，指出在多长时间内，有多少次更新操作，就将数据同步到数据文件。这个可以多个条件配合，比如默认配置文件中的设置，就设置了三个条件。

**appendonly yes**/**no** ，**appendonly**配置，指出是否在每次更新操作后进行日志记录，如果不开启，可能会在断电时导致一段时间内的数据丢失。因为redis本身同步数据文件是按上面的save条件来同步的，所以有的数据会在一段时间内只存在于内存中。

**appendfsync no**/**always**/**everysec** ，**appendfsync**配置，**no**表示等操作系统进行数据缓存同步到磁盘，**always**表示每次更新操作后手动调用**fsync**()将数据写到磁盘，**everysec**表示每秒同步一次。

##### ****6.redis主从配置****

redis支持**master-slave**的主从配置，配置方法是在从机的配置文件中指定**slaveof**参数为主机的ip和port即可

##### ****7.redis起步(链接整理)****

项目首页，下方是各种语言支持列表：

<http://code.google.com/p/redis/>

作者在wiki中给出了一个非常好的例子，以使我们可以快速上手，地址：

<http://code.google.com/p/redis/wiki/TwitterAlikeExample>

同时作者推荐的另一个教程，地址：

<http://labs.alcacoop.it/doku.php?id=articles:redis_land>

一个redis爱好者创建的相关问题讨论网站：

<http://www.rediscookbook.org/>

为什么使用 Redis及其产品定位

[http://www.infoq.com/cn/articles/tq-why-choose-redis](http://www.infoq.com/cn/articles/tq-why-choose-redis" \o "http://www.infoq.com/cn/articles/tq-why-choose-redis)

Redis内存使用优化与存储

[http://www.infoq.com/cn/articles/tq-redis-memory-usage-optimization-storage](http://www.infoq.com/cn/articles/tq-redis-memory-usage-optimization-storage" \o "http://www.infoq.com/cn/articles/tq-redis-memory-usage-optimization-storage)