[1. 文档控制 2](#_Toc28708)

[1.1. 修改记录 2](#_Toc2376)

[1.2. 分发 2](#_Toc29232)

[1.3. 文档格式说明 3](#_Toc5169)

[2. 目录 4](#_Toc27380)

[1. Redis相关概念 5](#_Toc30595)

[1.1. Redis与Memcached的区别: 5](#_Toc2497)

[2. 集群安装部署 5](#_Toc11261)

[2.1. Redis安装 6](#_Toc30803)

[2.1.1. 安装redis 6](#_Toc95)

[2.1.2. 创建安装路径 7](#_Toc20099)

[2.1.3. 修改配置文件 7](#_Toc26858)

[2.1.4. redis的关闭启动 7](#_Toc9266)

[2.1.5. 监控redis 8](#_Toc18454)

[2.2. 安装tomcat-1\2 8](#_Toc15264)

[2.2.1. jdk安装 8](#_Toc25099)

[2.2.2. 二进制安装tomcat 8](#_Toc20386)

[2.2.3. 将tomcat调用redis的包放入tomcat/lib下 9](#_Toc30856)

[2.2.4. 修改context.xml配置文件支持调用redis 9](#_Toc11336)

[2.2.5. 修改server.xml文件以支持调取创建的测试页 9](#_Toc15202)

[2.2.6. 创建测试页 9](#_Toc11764)

[2.2.7. 启动tomcat 10](#_Toc22991)

[2.3. 安装keepavlied 实现redis主从高可用 10](#_Toc25078)

[2.3.1. 修改redis-2的配置文件 10](#_Toc8971)

[2.3.2. 启动并测试 10](#_Toc7455)

[2.3.3. 测试主从复制 11](#_Toc22809)

[2.3.4. 安装keepavlied 11](#_Toc16373)

[2.3.5. 相关脚本 13](#_Toc11959)

[2.4. 安装配置apache 16](#_Toc23907)

[2.4.1. 安装httpd 17](#_Toc29111)

[2.4.2. 配置httpd 17](#_Toc24610)

[2.5. 测试 17](#_Toc11612)

**《 apache+tomcat+redis sesson id主从复制》**

|  |  |
| --- | --- |
| 作者： | 广州云本开源软件有限公司 |
| 创建时间： | 2017年10月22日 |
| 最新更新时间： |  |
| 版本： | 1.0 |

# 文档控制

## 修改记录

| **日期** | **作者** | **版本** | **修改参考** |
| --- | --- | --- | --- |
|  |  |  |  |
| 2017-10-22 |  | 1．0 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 分发

| **编号** | **名称** | **地点** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 文档格式说明

本文档采用以下格式：

* 命令、代码等指令型内容：

execute command 1

# 目录

[1. 文档控制 3](#_Toc12369)

[1.1. 修改记录 3](#_Toc31092)

[1.2. 分发 3](#_Toc3687)

[1.3. 文档格式说明 4](#_Toc15592)

[2. 目录 5](#_Toc28169)

[1. Redis相关概念 7](#_Toc22187)

[1.1. 数据类型 8](#_Toc21221)

[1.2. 内存存储 8](#_Toc32066)

[1.3. 持久化 9](#_Toc7354)

[1.4. 主从同步 9](#_Toc12184)

[1.5. 性能 9](#_Toc31840)

[1.6. 提供API的语言 9](#_Toc14927)

[2. Redis设计规范 10](#_Toc13912)

[2.1. 软件版本介绍 10](#_Toc30893)

[2.2. 目前稳定版本 10](#_Toc18063)

[2.3. Redis与其他数据库和软件区别 10](#_Toc13754)

[2.4. Redis使用理由 11](#_Toc22387)

[2.4.1. 取最新的N个数据的操作 12](#_Toc30606)

[2.4.2. 排行榜应用，取TOP操作 12](#_Toc20117)

[2.4.3. 需要精准设定过期时间的应用 12](#_Toc3560)

[2.4.4. 计数器应用 12](#_Toc871)

[2.4.5. Uniq操作，获取某时间所有数据排重值 13](#_Toc18038)

[2.4.6. 实时系统，反垃圾系统 13](#_Toc6358)

[2.4.7. Pub/Sub构建实时消息系统 13](#_Toc31872)

[2.4.8. 构建队列系统 13](#_Toc30931)

[2.4.9. 缓存 13](#_Toc9943)

[2.5. Redis 持久化 13](#_Toc10504)

[2.5.1. Redis不同级别持久化 13](#_Toc24107)

[2.5.2. 不同级别的优缺点 14](#_Toc10828)

[2.5.3. 如何选择持久化方式 14](#_Toc26432)

[2.6. 复制 15](#_Toc24717)

[2.6.1. 架构图 15](#_Toc26520)

[2.6.2. Resid复制特性 15](#_Toc12114)

[2.6.3. Redis复制原理 16](#_Toc24708)

# Redis相关概念

* Redis是一个key-value存储系统，和Memacached类似，它支持存储value类型相对更多，包括string（字符串）、list（链表）、set（集合）和zset（有序集合）。这些数据类型都支持push/pop、add/remove及取交集和差集及更丰富的操作，而且这些操作是原子性的。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的。区别的是Redsi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。
* Key-value store注重对海量数据存取的性能、分布式、扩展性支持上。并不需要传统关系型数据库的一些特性，例如：schema、失误、完整sql查询支持等等。因此分布式环境下的性能相对传统的关系型数据库有较大的提升。

## Redis与Memcached的区别:

* 内存利用率：

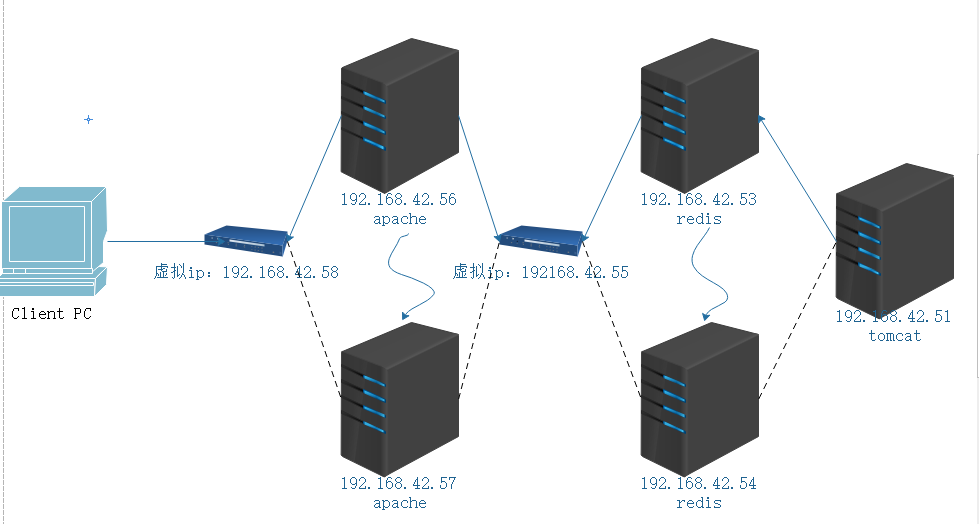
使用简单的key-value存储的话，Memcached的内存利用率更高，而如果Redis采用hash结构来做key-value存储，由于其组合式的压缩，其内存利用率会高于Memcached。

* 性能对比：

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，虽然Redis最近也在存储大数据的性能上进行优化，但是比起Memcached，还是稍有逊色。

# 集群安装部署

* Redis支持数据的持久化，可以将内存中的数据保持在磁盘中，重启的时候可以再次加载进行使用
* Redis支持数据的备份，即master-slave模式的数据备份。
* Redis不仅仅支持简单的key-Value类型的数据，同时还提供list，set，zset，hash等数据结构的存储。
* **架构图：**



## Redis安装

### 安装redis

1和2 安装方式一样，修改的配置文件不一样（下面看）

[root@reids-1 ~]# tar xf redis-3.2.5.tar.gz

[root@reids-1 ~]# cd redis-3.2.5

[root@reids-1 redis-3.2.5]# make

这里如果报错，tclsh8.5: not found 发现少了tcl 报错   
解决方法：下载，安装tcl

#wget http://downloads.sourceforge.net/tcl/tcl8.5.9-src.tar.gz

#cd /tcl8.5.9-src/unix

#./configure

#make

### 创建安装路径

[root@reids-1 ~]# mkdir -p /usr/local/redis/bin

[root@reids-1 ~]# mkdir -p /usr/local/redis/etc

[root@reids-1 ~]# mkdir -p /usr/local/redis/var

* **src目录下这些文件作用如下:**
* redis-server：Redis服务器的daemon启动程序
* redis-cli：Redis命令行操作工具.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

[root@reids-1 redis-3.2.5]# cd src/

[root@reids-1 src]# cp redis-benchmark redis-check-aof redis-cli redis-server /usr/local/redis/bin/

[root@reids-1 src]# cp ../redis.conf /usr/local/redis/etc

### 修改配置文件

[root@reids-1 ~]# vim /usr/local/redis/etc/redis.conf

128 daemonize yes

**是否把redis-server启动在后台，**

**默认是“否”。若改成yes，**

**会生成一个pid**

**pid文件**

61 bind 0.0.0.0

任意主机都可以访问

### redis的关闭启动

[root@reids-1 ~]# killall -9 redis-server

[root@reids-1 ~]# /usr/local/redis/bin/redis-server /usr/local/redis/etc/redis.conf

[root@reids-1 ~]# netstat -anpt |grep :6379

tcp 0 0 0.0.0.0:6379 0.0.0.0:\* LISTEN 29516/redis-server

### 监控redis

[root@reids-1 ~]# redis-cli -p 6379 monitor

OK

…

测试：访问192.168.4.55

发现测试页面发生变化， 变得是ip sessionid不变

## 安装tomcat-1\2

### jdk安装

[root@tomcat-1 ~]# tar xf jdk-7u65-linux-x64.gz

[root@tomcat-1 ~]# mv jdk1.7.0\_65 /usr/local/java

[root@tomcat-1 ~]# vim /etc/profile.d/java.sh

export JAVA\_HOME=/usr/local/java

export PATH=$PATH:$JAVA\_HOME/bin

[root@tomcat-1 ~]# source /etc/profile.d/java.sh

### 二进制安装tomcat

[root@tomcat-1 ~]# tar xf apache-tomcat-7.0.54.tar.gz

[root@tomcat-1 ~]# mv apache-tomcat-7.0.54 /usr/local/tomcat

### 将tomcat调用redis的包放入tomcat/lib下

[root@tomcat-1 ~]# cp tomcat-redis-session-manage-tomcat7.jar tomcat-juli.jar commons-logging-1.1.3.jar commons-pool2-2.2.jar jedis-2.5.2.jar /usr/local/tomcat/lib

### 修改context.xml配置文件支持调用redis

[root@tomcat-1 ~]# vim /usr/local/tomcat/conf/context.xml

<Context>

…

<Valve className="com.orangefunction.tomcat.redissessions.RedisSessionHandlerValve" />

<Manager className="com.orangefunction.tomcat.redissessions.RedisSessionManager"

redis服务器的虚拟ip地址

host="192.168.42.55"

port="6379"

database="0"

maxInactiveInterval="60" />

</Context>

### 修改server.xml文件以支持调取创建的测试页

[root@tomcat-1 ~]# vim /usr/local/tomcat/conf/server.xml

124 <Host name="localhost" appBase="webapps"

添加两行

125 unpackWARs="true" autoDeploy="true">

<Context docBase="/web/webapp" path="" reloadable="false" >

</Context>

### 创建测试页

[root@tomcat-1 ~]# mkdir -p /web/webapp

[root@tomcat-1 ~]# cd /web/webapp

[root@tomcat-1 webapp]# vim index.jsp

Session ID:<%= session.getId() %><BR>

tomcat2改成192.168.4.6

SessionPort:<%= request.getServerPort() %>

<% out.println("This tomcat server 192.168.4.5");%>

### 启动tomcat

[root@tomcat-1 ~]# /usr/local/tomcat/bin/startup.sh

[root@tomcat-1 ~]# netstat -anpt |grep :8080

tcp 0 0 :::8080 :::\* LISTEN 26474/java

## 安装keepavlied 实现redis主从高可用

### 修改redis-2的配置文件

[root@redis-2 src]# vim /usr/local/redis/etc/redis.conf

128 daemonize yes

61 bind 0.0.0.0

265 # slaveof <masterip> <masterport>

填写redis-1的ip和端口，同步redis-1的信息

slaveof 192.168.42.54 6379

### 启动并测试

[root@redis-2 ~]# /usr/local/redis/bin/redis-server /usr/local/redis/etc/redis.conf

[root@redis-2 ~]# netstat -anpt |grep redis

tcp 0 0 0.0.0.0:6379 0.0.0.0:\* LISTEN 29517/redis-server

tcp 0 0 192.168.42.53:42173 192.168.42.54:6379 ESTABLISHED 29517/redis-server

### 测试主从复制

[root@reids-1 ~]# /usr/local/redis/bin/redis-cli -h 192.168.4.7

192.168.42.53:6379> set name zhangsan

OK

[root@redis-2 ~]# /usr/local/redis/bin/redis-cli -h 192.168.4.8

192.168.42.54:6379> get name

"zhangsan"

### 安装keepavlied

* 两台安装方式一样，配置文件不一样

[root@redis1 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id redis1

}

vrrp\_script chk\_redis {

script "/etc/keepalived/scripts/redis\_check.sh"

interval 2

}

vrrp\_instance Redis {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

nopreempt

unicast\_src\_ip 192.168.42.53

unicast\_peer {

192.168.42.54

}

authentication {

auth\_type PASS

auth\_pass 112233

}

track\_script {

chk\_redis

}

virtual\_ipaddress {

192.168.42.55

}

notify\_master /etc/keepalived/scripts/redis\_master.sh

notify\_backup /etc/keepalived/scripts/redis\_backup.sh

notify\_fault /etc/keepalived/scripts/redis\_fault.sh

notify\_stop /etc/keepalived/scripts/redis\_stop.sh

}

[root@redis2 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id redis2

}

vrrp\_script chk\_redis {

script "/etc/keepalived/scripts/redis\_check.sh"

interval 2

}

vrrp\_instance Redis {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

# nopreempt

unicast\_src\_ip 192.168.42.54

unicast\_peer {

192.168.42.53

}

authentication {

auth\_type PASS

auth\_pass 112233

}

track\_script {

chk\_redis

}

virtual\_ipaddress {

192.168.42.55

}

notify\_master /etc/keepalived/scripts/redis\_master.sh

notify\_backup /etc/keepalived/scripts/redis\_backup.sh

notify\_fault /etc/keepalived/scripts/redis\_fault.sh

notify\_stop /etc/keepalived/scripts/redis\_stop.sh

}

### 相关脚本

[root@redis2 scripts]# ll /etc/keepalived/scripts/

总用量 24

-rwxr-xr-x 1 root root 891 10月 18 16:07 keepalived-redis-state.log

-rwxr-xr-x 1 755 root 457 10月 17 16:59 redis\_backup.sh

-rwxr-xr-x 1 755 root 140 10月 17 15:47 redis\_check.sh

-rwxr-xr-x 1 755 root 182 10月 17 16:41 redis\_fault.sh

-rwxr-xr-x 1 755 root 580 10月 18 09:19 redis\_master.sh

-rwxr-xr-x 1 755 root 162 10月 17 16:46 redis\_stop.sh

* **redis\_check.sh 两机相同**

[root@redis2 scripts]# cat redis\_check.sh

#!/bin/bash

ALIVE=`/usr/local/bin/redis-cli PING`

if [ "$ALIVE" == "PONG" ]

then

echo $ALIVE

exit 0

else

echo $ALIVE

exit 1

fi

* **redis\_master.sh**

[root@redis2 scripts]# cat redis\_master.sh

#!/bin/bash

#当进入master状态会呼叫notify\_master

redis\_cli="/usr/local/bin/redis-cli"

log\_file="/etc/keepalived/scripts/keepalived-redis-state.log"

echo "[master]" >> $log\_file

date >> $log\_file

echo "Being master...." >>$log\_file 2>&1

echo "Run SLAVEOF cmd....." >>$log\_file

######设置本地54为53的从机

$redis\_cli SLAVEOF 192.168.42.53 6379 >> $log\_file 2>&1q

#延迟10秒以后待数据同步完成后再取消同步状态

sleep 10

echo "Run SLAVEOF NO ONE cmd...." >>$log\_file

######设置为本地为Redis的主机

$redis\_cli SLAVEOF NO ONE >> $log\_file

[root@redis1 scripts]# cat redis\_master.sh

#!/bin/bash

#当进入master状态会呼叫notify\_master

redis\_cli="/usr/local/bin/redis-cli"

log\_file="/etc/keepalived/scripts/keepalived-redis-state.log"

echo "[master]" >> $log\_file

date >> $log\_file

echo "Being master...." >>$log\_file 2>&1

echo "Run SLAVEOF cmd....." >>$log\_file

######设置本机53为54的从机

$redis\_cli SLAVEOF 192.168.42.54 6379 >> $log\_file 2>&1q

#延迟10秒以后待数据同步完成后再取消同步状态

sleep 10

echo "Run SLAVEOF NO ONE cmd...." >>$log\_file

######设置为本地为Redis的主机

$redis\_cli SLAVEOF NO ONE >> $log\_file 2>&1

* **redis\_backup.sh**

[root@redis2 scripts]# cat redis\_backup.sh

#!/bin/bash

#当进入BACKUP状态会呼叫notify\_backup

redis\_cli="/usr/local/bin/redis-cli"

log\_file="/etc/keepalived/scripts/keepalived-redis-state.log"

echo "[backup]" >>$log\_file

date >> $log\_file

echo "Being slave......" >> $log\_file 2>&1

#延迟15秒待数据被对方同步完成之后再切换主从角色

sleep 15

echo "Run SLAVEOF cmd...." >>$log\_file

######设置本机54为53的从机

$redis\_cli SLAVEOF 192.168.42.53 6379 >>log\_file 2>&1

[root@redis1 scripts]# cat redis\_stop.sh

#!/bin/bash

#当程序终止时，则呼叫notify\_stop

log\_file=/etc/keepalived/scripts/keepalived-redis-state.log

echo "[stop]" >>$log\_file

date >>$log\_file

[root@redis1 scripts]# cat redis\_backup.sh

#!/bin/bash

#当进入BACKUP状态会呼叫notify\_backup

redis\_cli="/usr/local/bin/redis-cli"

log\_file="/etc/keepalived/scripts/keepalived-redis-state.log"

echo "[backup]" >>$log\_file

date >> $log\_file

echo "Being slave......" >> $log\_file 2>&1

#延迟15秒待数据被对方同步完成之后再切换主从角色

sleep 15

echo "Run SLAVEOF cmd...." >>$log\_file

######设置本机53为54的从机

$redis\_cli SLAVEOF 192.168.42.54 6379 >>log\_file 2>&1

* redis\_fault.sh redis\_stop.sh 脚本两机相同
* **redis\_fault.sh**

[root@redis2 scripts]# cat redis\_fault.sh

#!/bin/bash

#当发现异常情况时进入Fault状态呼叫notify\_fault

log\_file=/etc/keepalived/scripts/keepalived-redis-state.log

echo "[fault]" >>$log\_file

date >> $log\_file

* **redis\_stop.sh**

[root@redis2 scripts]# cat redis\_stop.sh

#!/bin/bash

#当程序终止时，则呼叫notify\_stop

log\_file=/etc/keepalived/scripts/keepalived-redis-state.log

echo "[stop]" >>$log\_file

date >>$log\_file

## 安装配置apache

### 安装httpd

Yum -y install httpd

### 配置httpd

Vim /etv/http/conf/httpd.conf

<Proxy balancer://myset>

BalancerMember http://192.168.42.51:8080 route=1

BalancerMember http://192.168.42.52:8080 route=2

# ProxySet lbmethod=bytraffic

</Proxy>

ProxyPass "/" "balancer://myset/"

ProxyPassReverse "/" "balancer://myset/"

## 测试

* 浏览器输入

firefox 192.168.42.58

可以看到tomcat 服务器ip在变化 sesson id没有发生改变