

原

Redis集群之主从复制，读写分离（下）（六）

2016年12月18日 11:51:34

cuipeng0916

阅读数 3695

CSDN Certification

版权声明：本文为博主原创文章，未经博主允许不得转载。<https://blog.csdn.net/cuipeng0916/article/details/53726092>

上一次呢我们讲到了redis的集群，还有redis的主从复制，读写分离的一些配置，那么接下来就接着上次还未完结的内容

上一次呢讲的是在正常的情况下redis服务在各个主机上的运行情况，那么接下来就是要介绍不正常的情况了。

假如说我们的redis的主库挂了或者是运行redis服务的服务器挂了，那么其余的redis从库是否会趁机上位还是忠于职守在slave的角色？那么接下来就为大家揭晓

首先启动redis服务，包括主库与从库

👍

3

💬

2

🔖

📱

⏪



```
[root@localhost bin]# redis-server redis_master.conf
[root@localhost bin]# redis-cli -p 6379
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> █
```

仅将文本发送到当前选项卡

已连接 192.168.2.66:22。

SSH2 xterm 165x100

```
-rwxr-xr-x 1 root root 5708988 Dec  4 17:40 redis-cli
-rw-r--r-- 1 root root  46711 Dec 17 11:44 redis_master.conf
lrwxrwxrwx 1 root root    12 Dec  4 17:40 redis-sentinel -> redis-server
-rwxr-xr-x 1 root root 7826926 Dec  4 17:40 redis-server
-rw-r--r-- 1 root root  46743 Dec 17 13:17 redis_slave_6380.conf
-rw-r--r-- 1 root root  46744 Dec 17 13:18 redis_slave_6381.conf
-rw-r--r-- 1 root root  19296 Dec 17 13:18 slave_6380.log
-rw-r--r-- 1 root root  11129 Dec 17 13:19 slave_6381.log
[root@localhost bin]# redis-server redis_slave_6380.conf
[root@localhost bin]# redis-cli -p 6380
127.0.0.1:6380> ping
PONG
127.0.0.1:6380> █
```

```
-rwxr-xr-x 1 root root 5708988 Dec  4 17:40 redis-cli
-rw-r--r-- 1 root root  46711 Dec 17 11:44 redis_master.conf
lrwxrwxrwx 1 root root    12 Dec  4 17:40 redis-sentinel -> redis-server
-rwxr-xr-x 1 root root 7826926 Dec  4 17:40 redis-server
-rw-r--r-- 1 root root  46743 Dec 17 13:17 redis_slave_6380.conf
-rw-r--r-- 1 root root  46744 Dec 17 13:18 redis_slave_6381.conf
-rw-r--r-- 1 root root  19296 Dec 17 13:18 slave_6380.log
-rw-r--r-- 1 root root  11129 Dec 17 13:19 slave_6381.log
[root@localhost bin]# redis-server redis_slave_6381.conf
[root@localhost bin]# redis-cli -p 6381
127.0.0.1:6381> ping
PONG
127.0.0.1:6381> █
```

<http://blog.csdn.net/cuipeng0916>

各个服务器上的redis服务均启动正常，那么接下来就是模拟redis主库宕机了

shutdown表示关闭redis服务

exit表示退出redis连接

```
127.0.0.1:6379> shutdown
not connected> exit
[root@localhost bin]# █
```

那么接下来就是查看各个redis从库的角色以及连接状态了

```
-----
127.0.0.1:6380> keys *
1) "k4"
2) "k3"
3) "k2"
4) "k1"
127.0.0.1:6380> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:down
master_last_io_seconds_ago:-1
master_sync_in_progress:0
slave_repl_offset:337
master_link_down_since_seconds:186
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
127.0.0.1:6380> █
```

```
127.0.0.1:6381> keys *
1) "k2"
2) "k4"
3) "k1"
4) "k3"
127.0.0.1:6381> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:down
master_last_io_seconds_ago:-1
master_sync_in_progress:0
slave_repl_offset:337
master_link_down_since_seconds:197
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
127.0.0.1:6381> █
```



3



2



我们可以看到，在从库中还是可以拿到数据的，说明redis主库挂了并不会影响redis从库的运行。但是看到master\_line\_status为down的时候，就知道这个时候的主库挂了，因为一开始的状态是up

那么如果redis主库的服务有重新启动了呢？redis从库会不会再次连接上主库？



```
[root@localhost bin]# redis-cli -p 6379
127.0.0.1:6379> keys *
1) "k2"
2) "k3"
3) "k4"
4) "k1"
127.0.0.1:6379> set k5 v5
OK
127.0.0.1:6379>

127.0.0.1:6380> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:up
master_last_io_seconds_ago:2
master_sync_in_progress:0
slave_repl_offset:81
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
127.0.0.1:6380> get k5
"v5"
127.0.0.1:6380>
```

```
127.0.0.1:6381> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:up
master_last_io_seconds_ago:2
master_sync_in_progress:0
slave_repl_offset:95
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
127.0.0.1:6381> get k5
"v5"
127.0.0.1:6381>
```

首先启动redis主库, 然后写入数据, 这个时候发现从库的master\_line\_status的连接状态都是up了, 并且可以取到在redis主库中写入的数据

那么这样子是不是很不好啊, 假如是电商网站, 然后突然间redis主库挂了, 那么这个时候就只有redis从库服务在运行了。但是redis从库是只读的是不是就无法写入数据了, 那么客户就无法下订单了。那么有没有什么方法, 就是在redis主服务挂了之后我再从redis从库服务中挑选出一个比较优秀的来接替主库的位置

### 方案一: 使用命令的方式

那么接下来呢我们就学习一个新的命令

```
slaveof no one
```

```
127.0.0.1:6380> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:down
master_last_io_seconds_ago:-1
master_sync_in_progress:0
slave_repl_offset:1
master_link_down_since_seconds:1482154107
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1067
repl_backlog_histlen:910
127.0.0.1:6380> SLAVEOF no one
OK
127.0.0.1:6380> info replication
# Replication
role:master
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1067
repl_backlog_histlen:910
```

可以看到，命令执行之后，就立刻趁机上位了，那么另外一台从库是不是也要换一个新的老大啊



3



2



```
127.0.0.1:6381> slaveof 127.0.0.1 6380
OK
127.0.0.1:6381> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6380
master_link_status:up
master_last_io_seconds_ago:1
master_sync_in_progress:0
slave_repl_offset:85
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
127.0.0.1:6381> █
```



3



2



但是这个时候原来的redis主库有杀回来了呢？这个时候是不是另外两个就是有两个redis主库的服务了，但是原来的从库并不会回到这个主库上去，而是后面那两台自立规则。那么这个时候是不是很不好啊，我们想要的是只有一个redis主库服务，那么有没有什么解决方法呢？

那么接下来就是终极的解决方案

## 方案二：哨兵模式

哨兵模式是反客为主的自动版，能够jiankong 后台主机是否故障，如果故障了根据投票数自动将从库转换为主库

首先恢复原来的一主N从的环境



```
[root@localhost bin]# redis-server redis_master.conf
[root@localhost bin]# redis-cli -p 6379
127.0.0.1:6379> shutdown
not connected> exit
[root@localhost bin]# redis-server redis_master.conf
[root@localhost bin]# redis-cli -p 6379
127.0.0.1:6379> info replication
# Replication
role:master
connected_slaves:2
slave0:ip=127.0.0.1,port=6381,state=online,offset=85,lag=0
slave1:ip=127.0.0.1,port=6380,state=online,offset=85,lag=1
master_repl_offset:85
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:84
127.0.0.1:6379>

127.0.0.1:6380> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:up
master_last_io_seconds_ago:8
master_sync_in_progress:0
slave_repl_offset:169
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:896
127.0.0.1:6380>

127.0.0.1:6381> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6379
master_link_status:up
master_last_io_seconds_ago:6
master_sync_in_progress:0
slave_repl_offset:183
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
127.0.0.1:6381>
```



3



2



接下来就是配置了

在redis安装目录下新建sentinel.conf文件, 名字绝不能错

```
[root@localhost bin]# touch sentinel.conf
[root@localhost bin]# vim sentinel.conf
```

编辑sentinel.conf文件, sentinel monitor 被监控数据库名字(自己起名字) 127.0.0.1 6379 1 【上面最后一个数字1, 表示主机挂掉后salve投票看让谁接替成为主机, 票数多少后成为主机】

```
1 sentinel monitor master 6379 127.0.0.1 6379 1
```

接下来就是启动哨兵进行监控了, 命令

```
redis-sentinel sentinel.conf
```



```
[root@localhost bin]# ll
total 27084
-rw-r--r-- 1 root root      116 Dec 19 21:47 dump.rdb
-rw-r--r-- 1 root root      116 Dec 19 21:42 dump_slave_6380.rdb
-rw-r--r-- 1 root root      116 Dec 19 21:41 dump_slave_6381.rdb
-rw-r--r-- 1 root root    34553 Dec 19 21:47 master_6379.log
-rwxr-xr-x 1 root root 5580279 Dec  4 17:40 redis-benchmark
-rwxr-xr-x 1 root root  22177 Dec  4 17:40 redis-check-aof
-rwxr-xr-x 1 root root 7826926 Dec  4 17:40 redis-check-rdb
-rwxr-xr-x 1 root root 5708988 Dec  4 17:40 redis-cli
-rw-r--r-- 1 root root   46711 Dec 17 11:44 redis_master.conf
lrwxrwxrwx 1 root root      12 Dec  4 17:40 redis-sentinel -> redis-server
-rwxr-xr-x 1 root root 7826926 Dec  4 17:40 redis-server
-rw-r--r-- 1 root root   46743 Dec 17 13:17 redis_slave_6380.conf
-rw-r--r-- 1 root root   46744 Dec 19 21:26 redis_slave_6381.conf
-rw-r--r-- 1 root root     46 Dec 19 21:51 sentinel.conf
-rw-r--r-- 1 root root  256098 Dec 19 21:55 slave_6380.log
-rw-r--r-- 1 root root  290701 Dec 19 21:55 slave_6381.log
[root@localhost bin]# redis-sentinel sentinel.conf
```



3

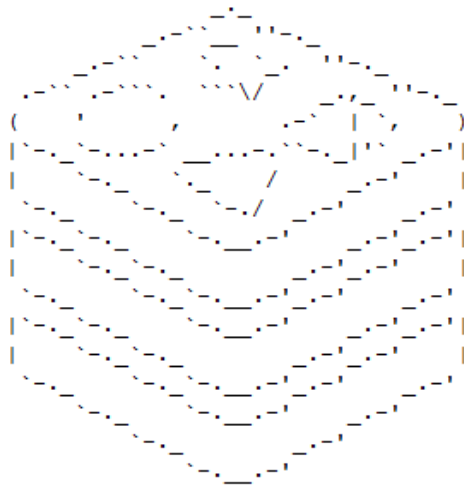


2





```
-----  
1311:X 19 Dec 21:57:03.622 * Increased maximum number of open files to 10032 (it was originally set to 1024).
```



```
Redis 3.2.5 (00000000/0) 64 bit
```

```
Running in sentinel mode
```

```
Port: 26379
```

```
PID: 1311
```

```
http://redis.io
```

```
1311:X 19 Dec 21:57:03.626 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to 128  
1311:X 19 Dec 21:57:03.689 # Sentinel ID is 6bd042deb80750ed071bcc489572dfa92ac8afd5  
1311:X 19 Dec 21:57:03.690 # +monitor master master_6379 127.0.0.1 6379 quorum 1  
1311:X 19 Dec 21:57:03.691 * +slave slave 127.0.0.1:6381 127.0.0.1 6381 @ master_6379 127.0.0.1 6379  
1311:X 19 Dec 21:57:03.696 * +slave slave 127.0.0.1:6380 127.0.0.1 6380 @ master 6379 127.0.0.1 6379
```

那么这个时候就是哨兵开始监控了

首先模拟主库宕机, 关闭主库, 并且观察哨兵输出的日志并且两个从库角色的变化



3



2



```
1311:X 19 Dec 21:57:03.626 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to 128
1311:X 19 Dec 21:57:03.689 # Sentinel ID is 6bd042deb80750ed071bcc489572dfa92ac8afd5
1311:X 19 Dec 21:57:03.690 # +monitor master master_6379 127.0.0.1 6379 quorum 1
1311:X 19 Dec 21:57:03.691 * +slave slave 127.0.0.1:6381 127.0.0.1 6381 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 21:57:03.696 * +slave slave 127.0.0.1:6380 127.0.0.1 6380 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.296 # +sdown master master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.296 # +odown master master_6379 127.0.0.1 6379 #quorum 1/1
1311:X 19 Dec 22:01:19.296 # +new-epoch 1
1311:X 19 Dec 22:01:19.296 # +try-failover master master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.341 # +vote-for-leader 6bd042deb80750ed071bcc489572dfa92ac8afd5 1
1311:X 19 Dec 22:01:19.341 # +elected-leader master master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.341 # +failover-state-select-slave master master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.403 # +selected-slave slave 127.0.0.1:6381 127.0.0.1 6381 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.403 * +failover-state-send-slaveof-noone slave 127.0.0.1:6381 127.0.0.1 6381 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.458 * +failover-state-wait-promotion slave 127.0.0.1:6381 127.0.0.1 6381 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.758 # +promoted-slave slave 127.0.0.1:6381 127.0.0.1 6381 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.758 # +failover-state-reconf-slaves master master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:19.811 * +slave-reconf-sent slave 127.0.0.1:6380 127.0.0.1 6380 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:20.672 * +slave-reconf-inprog slave 127.0.0.1:6380 127.0.0.1 6380 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:20.672 * +slave-reconf-done slave 127.0.0.1:6380 127.0.0.1 6380 @ master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:20.761 # +failover-end master master_6379 127.0.0.1 6379
1311:X 19 Dec 22:01:20.761 # +switch-master master_6379 127.0.0.1 6379 127.0.0.1 6381
1311:X 19 Dec 22:01:20.762 * +slave slave 127.0.0.1:6380 127.0.0.1 6380 @ master_6379 127.0.0.1 6381
1311:X 19 Dec 22:01:20.762 * +slave slave 127.0.0.1:6379 127.0.0.1 6379 @ master_6379 127.0.0.1 6381
1311:X 19 Dec 22:01:50.794 # +sdown slave 127.0.0.1:6379 127.0.0.1 6379 @ master_6379 127.0.0.1 6381
```



3



2



```
127.0.0.1:6380> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6381
master_link_status:up
master_last_io_seconds_ago:1
master_sync_in_progress:0
slave_repl_offset:10298
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:896
127.0.0.1:6380>

127.0.0.1:6381> info replication
# Replication
role:master
connected_slaves:1
slave0:ip=127.0.0.1,port=6380,state=online,offset=0,lag=0
master_repl_offset:1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:0
127.0.0.1:6381>
```

<http://blog.csdn.net/cuipeng0916>

（这里需要等待片刻才可看到结果）大家可以看到，端口号为6381的redis从库立马变成了主库，而且端口号为6380的redis从库就跟着端口号为6381混了。

那么问题来了，如果原来的老大回来了呢，6381会不会让位呢？

```
[root@localhost bin]# redis-server redis_master.conf
[root@localhost bin]# redis-cli -p 6379
127.0.0.1:6379> info replication
# Replication
role:slave
master_host:127.0.0.1
master_port:6381
master_link_status:up
master_last_io_seconds_ago:0
master_sync_in_progress:0
slave_repl_offset:30597
slave_priority:100
slave_read_only:1
connected_slaves:0
master_repl_offset:0
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
127.0.0.1:6379>
```

👍

3

💬

2

🔖

📱

<

（这里也许等待片刻）那么结果是原来的redis服务就变成从库了，连接着现在的端口号为6381的主库了。

哨兵模式的缺点：由于所有的写操作都是先在Master上操作，然后同步更新到Slave上，所以从Master同步到Slave机器有一定的延迟，当系统很繁忙的时候，延迟问会更加严重，Slave机器数量的增加也会使这个问题更加严重。

OK，到这里就是我们redis主从复制，读写分离的全部介绍了

如果大家觉得有哪些地方看不懂或者我的描述有问题的话请留言提醒我，谢谢



想对作者说点什么



有梦青年： 直接新建sentinel.conf文件不行啊,我是从编译的redis里复制的sentinel.conf文件然后改的 （1年前 #2楼）

👍



傻子点点： 总结很好，学习了。 （2年前 #1楼）

👍

## redis实战--redis主从复制实现读写分离（原理）

阅读数 3386

1、redis高并发跟整个系统的高并发之间的关系redis，你要搞高并发的话，不可避免... 博文 | 来自： 码农时代

<div>redis如何在cluster模式启动后实现读写分离？</div> <div>根据redis官方资料描述，通过slaveof的方式配置主从模式，可以实现读写分离，即主节点...<a href="#">论坛</a></div>	05-25	<div><div>👍</div><div>3</div></div>
<div>Redis-主从复制和读写分离</div> <div>一、Master&amp;amp;Slave是什么？<a href="#">也就是我们所说的主从复制，主机数据更...</a><a href="#">博文</a>   来自：<a href="#">liberty123...</a></div> <div>阅读数 820</div>		<div><div>💬</div><div>2</div></div>
<div>Redis 读写分离</div> <div>读写分离：对于读占比较高的场景，可以通过把一部分流量分摊导出从节点(salve)来...<a href="#">博文</a>   来自：<a href="#">fd2025的...</a></div> <div>阅读数 3067</div>		<div><div>🔖</div></div>
<div>12.Redis 读写分离</div> <div>1.Redis通过读写分离承载10w+的QPS1、redis不能支撑高并发的瓶颈在哪里？单机2...<a href="#">博文</a>   来自：<a href="#">enlyhua...</a></div> <div>阅读数 720</div>		<div><div>📱</div></div>
<div>8-redis主从复制：redis读写分离(主从复制架构)的原理</div> <div>主从复制架构原理redisreplication的核心机制（1）master节点采用异步方式复制数据...<a href="#">博文</a>   来自：<a href="#">weixin_4...</a></div> <div>阅读数 33</div>		<div><div>⏪</div></div>
<div>【Redis】的主从复制（读写分离）详解</div> <div>Redis的主从复制的优点有两处，分别是避免Redis单点故障构建读写分离架构，满足...<a href="#">博文</a>   来自：<a href="#">qq_3502...</a></div> <div>阅读数 250</div>		
<div>redis缓存架构-读写分离(三)</div> <div>1.redis如何通过读写分离来承载读请求QPS超过10万+?背景:redis，你要搞高并发的...<a href="#">博文</a>   来自：<a href="#">duanlei的...</a></div> <div>阅读数 18</div>		<div><div>👑</div><div>📺</div></div>
<div>• 在使用redis+twemproxy时，怎么做到读写分离？</div> <div>-<a href="#">问答</a></div>	0	<div><div>👤</div></div>
<div>6.redis sentinel实现高可用读写分离</div> <div>1.redissentinel故障转移的基本原理：多个sentinel发现并确认master有问题选举出一...<a href="#">博文</a>   来自：<a href="#">我去个地方</a></div> <div>阅读数 1835</div>		<div><div>⚠</div></div>
<div>redis主从库读写分离</div> <div>前言：随着web2.0的进一步发展，网民的生产力进一步提升，存储总量开始增加。此...<a href="#">博文</a>   来自：<a href="#">iechenyb...</a></div> <div>阅读数 770</div>		<div><div>⬆</div></div>

<div>redis<b>读写分离</b>机制与哨兵机制</div> <div>一、在/etc/redis/里面使用命令cpredis.conf/redis63791.conf,cpredis.conf/redis63792....</div>	阅读数 1781	博文   来自: 一叶孤舟...	3
<div>Redis配置<b>主从复制</b>，实现<b>读写分离</b></div> <div>前言：由于redis的高性能，在应用中对其依赖很高，有时候一台redis服务器性能不够...</div>	阅读数 530	博文   来自: 赵阳的博客	2
<div>Redis<b>集群之主从复制</b>，<b>读写分离</b>（上）（五）</div> <div>前言：随着web2.0的进一步发展，网民的生产力进一步提升，存储总量开始增加。此...</div>	阅读数 9353	博文   来自: cuipeng...	
<div>Redis配置主从架构，实现<b>读写分离</b></div> <div>Redis的主从架构，能帮助我们实现读多，写少的情况，下面配置Redis架构，很简单...</div>	阅读数 1万+	博文   来自: 打不死的...	<
<div>Redis教程（二） - <b>读写分离</b></div> <div>Redis教程（一）-安装场景我们为了适应大流量访问，提高Redis读写效率，不能让访...</div>	阅读数 145	博文   来自: EugeneH...	
<div>【Redis】Redis主从实现<b>读写分离</b></div> <div>前言 大家在工作中可能会遇到这样的需求，即Redis读写分离，目的是为了压力分散...</div>	阅读数 866	博文   来自: 程序员俱...	
<div>Redis主从结构，主库宕机挂了，怎么办</div> <div>前提：现在有主从结构，主库没有配置持久化，从库配置AOF。(主库用来备份和写服...</div>	阅读数 1万+	博文   来自: whycold...	VIP
<div>REDIS缓存集群介绍</div> <div>分享人：张强具体的安装步骤点击：REDIS集群配置（LINUX）1.背景介绍2.知识剖...</div>	阅读数 2228	博文   来自: 理想的世界	QR
<div>关于<b>读写分离</b>，读延迟的问题如何解决</div> <div>如果实现读写分离，但有些事务的逻辑里面，有可能后面的操作会对前面刚插入的一条记...</div>		论坛	0
<div>Redis<b>读写分离</b></div> <div>场景：Redis的主从架构，能帮助我们实现读多，写少的情况意义：读写分离主要是为...</div>	阅读数 190	博文   来自: LexKaing...	!
<div>redis<b>主从复制,读写分离</b>以及哨兵的配置</div> <div>1.Redis主从配置1.1：安装redis 具体教程可以看我安装redis-cluster博客中的内容,这...</div>	阅读数 3319	博文   来自: 这个名字...	^

<div><div>Redis主从复制,读写分离及sentinel监控的使用</div><div>在有redis基础的前提下，你肯定会发现redis真的很强大很实用，至少我是这么觉得的... 博文   来自： YZR的博客</div></div>	阅读数 1624	<div><div><div></div></div><div>3</div></div>
<div><div>redis sentinel 读写分离(六)</div><div>redissentinel读写分离(一)redissentinel读写分离(二)redissentinel读写分离(三)redissen... 博文   来自： weixin_3...</div></div>	阅读数 41	<div><div><div></div></div><div>2</div></div>
<div><div>Redis Sentinel 读写分离 Slave 连接池</div><div>更好阅读体验，请移步：[url]http://www.jack-yin.com/coding/spring-boot/2683.html[/u... 博文   来自： Jack_Yin...</div></div>	阅读数 210	<div><div><div></div></div></div>
<div><div>redis配置读写分离以及利用哨兵sentinel进行自动主从切换</div><div>redis利用哨兵（sentinel）进行主从切换，断断续续，自己终于通过配置验证了一下该... 博文   来自： LZJLZJL...</div></div>	阅读数 9337	<div><div><div></div></div></div>
<div><div>Redis之Sentinel主从复制，读写分离，主从切换</div><div>当数据量变得庞大的时候，读写分离还是很有必要的。同时避免一个redis服务宕机， ... 博文   来自： andyzhao...</div></div>	阅读数 625	
<div><div>四:redis主从读写分离，哨兵模式与spring的集成</div><div>本篇主要介绍redis实现主从读写分离，采用哨兵模式后如何与spring集成在项目中进... 博文   来自： zyshappy...</div></div>	阅读数 5398	
<div><div>redis主从复制和集群配置</div><div>配置Redis主从复制和集群配置详解1. 配置redis主从复制。1.1. 应用场景一般来说， ... 博文   来自： wyp5841...</div></div>	阅读数 936	<div><div><div></div></div></div>
<div><div>Redis的集群与主从复制（附Redis持久化）</div><div>Redis持久化1、Rdb方式Redis默认的方式，redis通过快照来将数据持久化到磁盘中... 博文   来自： shaonian...</div></div>	阅读数 1214	<div><div><div></div></div></div>
<div><div>redis集群—主从同步（复制）</div><div>Redis学习笔记之主从同步（重要）Redis进阶实践之十Redis主从复制的集群模式主... 博文   来自： 每天进步...</div></div>	阅读数 255	<div><div><div></div></div></div>
<div><div>Redis集群之主从模式</div><div>Redis的定位还是分布式缓存，关于分布式的特点和挑战这里不再介绍。一、Redis主... 博文   来自： xiaoxiaoy...</div></div>	阅读数 1418	<div><div><div></div></div></div>

<div>Redis集群主从复制</div> <div>Redis主机数据更新后根据配置和策略，自动同步到备机的master/slaver机制， Master... 博文   来自： 路人乙的...</div>	阅读数 264	
<div>redis主从复制和集群实现原理</div> <div>redis主从复制redis主从配置比较简单，基本就是在从节点配置文件加上:slaveof192.1... 博文   来自： 爱代码也...</div>	阅读数 2万+	3
<div>Redis Cluster集群主从方案（附Jedis Cluster教程）</div> <div>RedisCluster集群主从方案（附JedisCluster教程） 本文介绍一种通过Jedis和Cluster... 博文   来自： 永远的VC...</div>	阅读数 70	2
<div>redis学习四 ----- redis主从同步(读写分离)</div> <div>环境:系统:centos6.4redis:3.2.0机器ip:192.168.10.160 , 192.168.10.161主从同步原理:... 博文   来自： 九十的博客</div>	阅读数 4859	
<div>4.Redis主从复制集群搭建步骤</div> <div>本文主要介绍Redis主从复制的详细配置步骤。1.主-从模式；2.主-从-从模式。 博文   来自： security_...</div>	阅读数 600	
<div>Redis主从复制和集群配置</div> <div>Redis主从复制和集群配置说明</div>	08-21	下载
<div>RedisCluster读写分离改造</div> <div>RedisCluster模式启动的环境中，通过Redis中的每个连接，都可以访问clusternodes... 博文   来自： clamaa的...</div>	阅读数 2528	
<div>Redis集群 - 配置最简单的Redis主从读写分离</div> <div>要求：1、你已经知道redis是什么2、熟悉Linux基本操作(CentOS6.5)3、有单机redis... 博文   来自： The Hard...</div>	阅读数 2153	
<div>redis集群之读写分离</div> <div>http://www.toutiao.com/i6355296754555945474/?tt_from=mobile_qq&amp;utm_campaign... 博文   来自： u013322...</div>	阅读数 616	
<div>Redis集群主从复制,读写分离（上）</div> <div>前言：随着web2.0的进一步发展，网民的生产力进一步提升，存储总量开始增加。此... 博文   来自： 妄笑伊的...</div>	阅读数 451	
<div>Redis集群主从复制,读写分离（下）</div> <div>上一次呢我们讲到了Redis的集群，还有redis的主从复制，读写分离的一些配置，那... 博文   来自： 妄笑伊的...</div>	阅读数 600	



<div>Redis主从同步，读写分离设置</div> <div>本文介绍使用Redis的主从同步功能(master,slave)，使程序实现读写分离，避免io瓶颈... 博文   来自： 傲雪星枫</div>	阅读数 2977	<div>3</div> <div>4-5</div> <div>2</div> <div>5-26</div> <div></div>
<div>Redis集群之主从复制,读写分离(上)(五) - cuipeng的博客 - CSDN博客</div> <div>https://blog.csdn.net/cuipeng0916/article/details/53704629 前言:随着web2.0...博文 来自: 妄笑伊的博客 Redis集群之主从复制,读写分离(下)(六) 12-18...</div>		
<div>redis实战--redis主从复制实现读写分离(原理) - 码农时..._CSDN博客</div> <div>Redis集群之主从复制,读写分离(下)(六) 12-18 阅读数 3652 上一次呢我们...博文 来自: cuipeng的博客 在使用redis+twemproxy时,怎么做到读写分离? 03...</div>		
<div>mysql主从复制和读写分离实战</div> <div>目录mysql主从复制和读写分离实战mysql主从复制Master配置Slave1配置Slave2配置... 博文   来自： fsx25505...</div>	阅读数 4618	
<div>mysql主从复制与读写分离</div> <div>主从复制与读写分离如何一步步的 博文   来自： qinwunan...</div>	阅读数 6349	
<div>mysql集群技术：主从复制，读写分离</div> <div>服务器端读写分离的具体技术1数据库集群技术集群由3个概念2数据库复制replication... 博文   来自： kwinH的...</div>	阅读数 1581	
<div>Mysql数据库的主从复制读写分离搭建过程与原理介绍</div> <div>首先说明一下在搭建Mysql数据库的主从复制与读写分离的过程中十分容易出错，所以... 博文   来自： Major818...</div>	阅读数 2375	
<div>mysql5.7主从复制配置——读写分离实现</div> <div>为什么使用主从架构？ 1、实现服务器负载均衡； 2、通过复制实现数据的异地备份； ... 博文   来自： 幻羽公子...</div>	阅读数 2003	
<div>MySQL5.7.18基于事务的主从复制和读写分离</div> <div>在以前的mysql版本中，读写分离的实现一般都是基于日志的主从复制实现的，这样会... 博文   来自： lvshaoron...</div>	阅读数 1511	
<div>DirectX修复工具增强版</div> <div>最后更新： 2019-5-26 DirectX修复工具最新版： DirectX Repair V3.8 增强版NEW! 版... 博文   来自： VBcom的...</div>	阅读数 206万+	

<div>python图片处理类之~PIL.Image模块(ios android icon图标自动...</div> <div>1.从pyCharm提示下载PIL包 <a href="http://www.pythonware.com/products/pil/">http://www.pythonware.com/products/pil/</a> 2.解压后，进... 博文   来自: 专注于co...</div>	阅读数 11万+	
<div>centos7 开启网络连接</div> <div>1.当我们发现无法联网时，我们运行下面命令或者ping命令 ip addr 结果没有显示局域... 博文   来自: sfeng95...</div>	阅读数 3万+	<div>3</div>
<div>二值神经网络（Binary Neural Network，BNN）</div> <div>在我刚刚过去的研究生毕设中，我在ImageNet数据集上验证了图像特征二值化后仍然... 博文   来自: 雨石</div>	阅读数 3万+	<div>2</div>
<div>expat介绍文档翻译</div> <div>原文地址：<a href="http://www.xml.com/pub/a/1999/09/expat/index.html">http://www.xml.com/pub/a/1999/09/expat/index.html</a> 因为需要用，所以才... 博文   来自: ymj71506...</div>	阅读数 7万+	
<div>Unity自动打包工具</div> <div>Unity一键打包工具，一键生成几十个平台/渠道的安装包。 博文   来自: 夜风的BL...</div>	阅读数 1万+	
<div>RUtills -- 建立在Rserve之上的Java与R语言通信工具集</div> <div>注1：RUtills是我偶然发现的一个工具包，它建立在Rserve之上，可以很大程度上简化... 博文   来自: 竹叶青的...</div>	阅读数 6876	
<div>搭建图片服务器《二》-linux安装nginx</div> <div>nginx是个好东西，Nginx (engine x) 是一个高性能的HTTP和反向代理服务器，也是一... 博文   来自: maoyuan...</div>	阅读数 8万+	
<div>使用Unity实现卡拉OK歌词过渡效果</div> <div>好长时间之前做过的一个项目，其中设计到用Unity模拟卡拉OK歌词过渡的效果，如下... 博文   来自: 月儿圆</div>	阅读数 1686	
<div>链接（freemarker）</div> <div><a href="http://blog.csdn.net/liaomin416100569/article/details/78349072">http://blog.csdn.net/liaomin416100569/article/details/78349072</a> 博文   来自: biao2858...</div>	阅读数 498	
<div>Docx4j 简单操作文字图片（包含页眉页脚和主体内容）</div> <div>docx4j官方提供了一些例子，本文只是其中一部分应用的简单例子。需要注意的地方... 博文   来自: 偶尔记一下</div>	阅读数 1万+	
<div>微信支付V3微信公众号支付PHP教程(thinkPHP5公众号支付)/JSS...</div> <div>扫二维码关注，获取更多技术分享 本文承接之前发布的博客《微信支付V3微信公众... 博文   来自: Marswill</div>	阅读数 18万+	

<div>Keil（MDK-ARM）系列教程（一）_新建软件工程详细过程</div> <div>推荐 分享一个大神的人工智能教程。零基础！通俗易懂！风趣幽默！还带黄段子！希... 博文   来自： <a href="#">strongerH...</a></div>	阅读数 6万+	<div>👍 3</div>
<div>ThreadLocal的设计理念与作用</div> <div>Java中的ThreadLocal类允许我们创建只能被同一个线程读写的变量。因此，如果一... 博文   来自： <a href="#">u0118607...</a></div>	阅读数 9万+	<div>💬 2</div>
<div>如何用一行C++代码读写数据库</div> <div>这篇文章要表达的并非数据库相关的知识，而是如何使用DBIOWrapper。 DBIOW... 博文   来自： <a href="#">哈哈 哈 ...</a></div>	阅读数 9263	<div>🔖</div>
<div>强连通分量及缩点tarjan算法解析</div> <div>强连通分量： 简言之 就是找环（每条边只走一次，两两可达） 孤立的一个点也是一... 博文   来自： <a href="#">九野的博客</a></div>	阅读数 66万+	<div>📱</div>
<div>忘记Oracle数据库管理员密码的解决办法</div> <div>安装oracle 9i后，居然把刚刚更改的数据库管理员密码给忘了，又不重新安装，太麻... 博文   来自： <a href="#">llxsharp...</a></div>	阅读数 1万+	<div>⏪</div>
<div>Cocos2d-x 2.2.3 使用NDK配置编译环境</div> <div>Cocos2d-x 2.2.3 使用NDK配置编译环境2014年6月11日 Cocos2d-x 3.0以下的开发环... 博文   来自： <a href="#">巫山老妖</a></div>	阅读数 2万+	
<div>jquery/js实现一个网页同时调用多个倒计时(最新的)</div> <div>jquery/js实现一个网页同时调用多个倒计时(最新的) 最近需要网页添加多个倒计时. 查... 博文   来自： <a href="#">Websites</a></div>	阅读数 54万+	<div>👑</div>
<div>关于SpringBoot bean无法注入的问题（与文件包位置有关）</div> <div>问题场景描述整个项目通过Maven构建，大致结构如下： 核心Spring框架一个module... 博文   来自： <a href="#">开发随笔</a></div>	阅读数 27万+	<div>👤</div>
<div>VirtualBox COM获取对象失败</div> <div>错误详情 1. 先来看看错误详情 获取 VirtualBox COM 对象失败.应用程序将被中断.Fail... 博文   来自： <a href="#">多点折腾...</a></div>	阅读数 4万+	<div>⚠</div>
<div>Solaris11挂载光盘</div> <div>将ISO写入光盘 cdrecord sol-11_1-repo-full.iso 弹出光盘 root@solaris:/media# eject c... 博文   来自： <a href="#">数据之路-...</a></div>	阅读数 1745	<div>⬆</div>
<div>.NET和java的RSA互通，仅此而已</div> <div>RSA .net jva 互通 解决不能互通的问题 博文   来自： <a href="#">lubiaopan...</a></div>	阅读数 7万+	

<div>保护模式及其编程——8086系统寄存器和系统指令</div> <div>摘要：为了协助处理器完成初始化和控制系统操作，80x86提供了一个标志寄存器和...</div>	<div>阅读数 1233</div> <div>博文   来自： <a href="#">河西无名式</a></div>	<div>👍 3</div> <div>💬 2</div> <div>🔖</div> <div>📱</div> <div>&lt;</div>
<div>IE9 table显示问题,td出现错位的解决方法</div> <div>IE9 table显示问题 IE9 大数据行table的显示会出现列错位 原因是IE9的BUG 在jsp中...</div>	<div>阅读数 5461</div> <div>博文   来自： <a href="#">yzp_soft...</a></div>	
<div>仿饿了么</div> <div>http://blog.csdn.net/zxt0601/article/details/54235736</div>	<div>阅读数 322</div> <div>博文   来自： <a href="#">小马哥nic...</a></div>	
<div>Web前端面试指导(三十)：JavaScript有几种类型的值？你能画一...</div> <div>题目点评 数据类型是所有程序都会涉及到的，是计算机语言比较基础知识，这种问题...</div>	<div>阅读数 7718</div> <div>博文   来自： <a href="#">雄领IT的...</a></div>	
<div>git丢掉本地的修改，还原修改之前的程序</div> <div>今天在本地做了修改，后来又不想要这次修改的内容，想要还原到修改之前的状态， ...</div>	<div>阅读数 2437</div> <div>博文   来自： <a href="#">leedaning...</a></div>	
<div>origion绘制热图</div> <div>有时我们需要绘制热图，用x轴、y轴表示两维数据，用颜色表示第三维 第一步：需要...</div>	<div>阅读数 6588</div> <div>博文   来自： <a href="#">SunCherr...</a></div>	
<div>VS2015+Qt5+OpenCV3开发环境配置</div> <div>本文介绍如何使用VS2015作为编译开发环境，调用OpenCV3.31和Qt5.9.1写图像处理...</div>	<div>阅读数 1万+</div> <div>博文   来自： <a href="#">zhhp1001...</a></div>	<div></div> <div></div> <div></div> <div></div> <div></div>
<div>机器学习教程   Objective-C培训   交互设计视频教程   颜色模型   设计制作学习</div>		
<div>• redis实战--redis主从复制实现读写分离(部署搭建) - Ja..._CSDN博客</div> <div>前面已经介绍过了redis主从复制实现读写分离的原理;这一篇文章主要介绍下如何部署...Redis集群之主从复制,读写分离(下)(六) - cuipeng的博客 12-18 3187 上...</div>		<div>1</div>
<div>• Redis详解(六)---主从复制、读写分离 - 安静的小唐 - CSDN博客</div> <div>Redis集群之主从复制,读写分离(下)(六) 12-18 阅读数 3647 上一次呢我们...博文 来自: cuipeng的博客 redis实战--redis主从复制实现读写分离(原理) 07...</div>		<div>5-24</div>



3 YEARS

cuipeng0916

关注

原创7

粉丝6

喜欢4

评论5

等级：

博客 2

访问：1万+

积分：261

排名：36万+

勋章：

恒

最新文章

Redis集群之主从复制，读写分离（上）（五）

Redis事务介绍（四）

Redis持久化之AOF（三）

Redis持久化之RDB（二）

Redis入门 Hello World

个人分类

Linux1篇

归档

2016年12月7篇

热门文章

Redis集群之主从复制，读写分离（上）（五）

阅读数 9352

3

2

















Redis集群之主从复制，读写分离（下）（六）

阅读数 3693

Redis事务介绍（四）

阅读数 3077

Redis入门 Hello World

阅读数 1015

Redis持久化之RDB（二）

阅读数 625

最新评论

Redis集群之主从复制，读写分离...

LOVE\_\_\_\_JAVA：非常浅显易懂

Redis事务介绍（四）

qq\_43620428：赵海洋

Redis集群之主从复制，读写分离...

qq\_32814555：直接新建sentinel.conf文件不行啊，我是从编译的redis里复制的sentinel.conf文件然...

Redis集群之主从复制，读写分离...

u013034640：总结很好，学习了。

Redis集群之主从复制，读写分离...

u013034640：总结很好



程序人生



CSDN资讯

QQ 客服

kefu@csdn.net

客服论坛

400-660-0108

工作时间 8:30-22:00

关于我们 | 招聘 | 广告服务 | 网站地图

3

2



 百度提供站内搜索 京ICP备19004658号  
京公网安备11010502030143  
©1999-2019 北京创新乐知网络技术有限公司

网络110报警服务 经营性网站备案信息  
北京互联网违法和不良信息举报中心  
中国互联网举报中心 家长监护 版权申诉

  
3

  
2

