

Redis 哨兵模式之实践篇

Original Edwin 彬 星河之码 8/7

收录于话题

#Redis

8个

由经验而得的智慧，胜于书本而得的理论。

在上一篇文章----[Redis 哨兵模式之实现原理篇](#)中简单介绍了Redis的哨兵模式所涉及的一些理论。实践出真知，本文将介绍搭建哨兵模式的全过程，实现Redis的高可用。

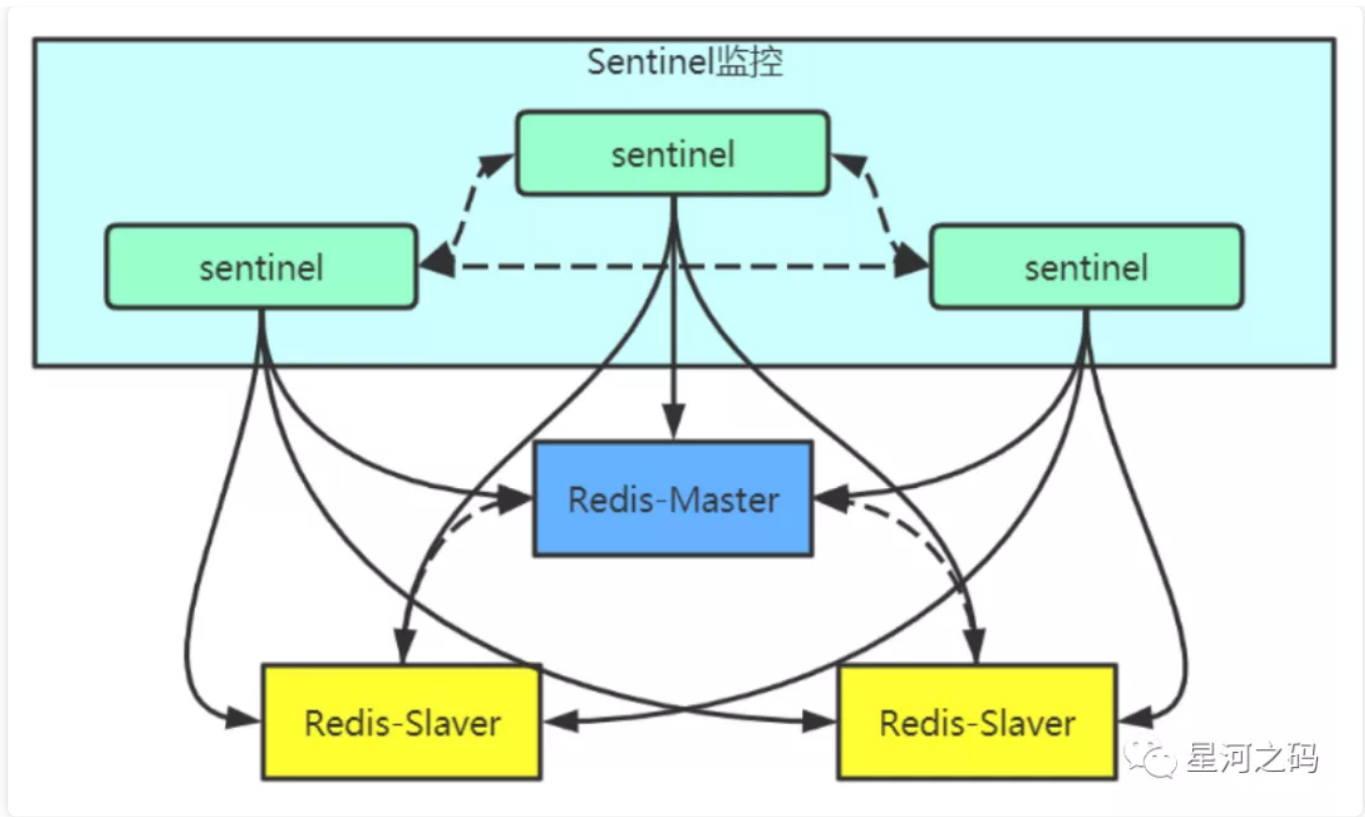
❖

部署方案

❖

在一台机器上采用伪分布式的方式部署。（生产环境应该是多台机器）

实现目标是一台机器上搭建 6 个节点，构成**一主两从三哨兵**集群模式。



服务器配置

类型	Ip	port
master	127.0.0.1	6379
slaver_1	127.0.0.1	6380
slaver_2	127.0.0.1	6381
sentinel_1	127.0.0.1	26390
sentinel_2	127.0.0.1	26391
sentinel_3	127.0.0.1	26392



环境搭建



下载解压

- 创建目录 `mkdir redisSentinel` 。在 `redisSentinel` 目录下创建一下目录，模拟不同的Redis实例

```
drwxr-xr-x 2 root root 4096 Aug 4 16:02 master-6379
drwxrwxr-x 7 root root 4096 Oct 27 2020 redis-6.0.9
-rw-r--r-- 1 root root 2261418 Oct 27 2020 redis-6.0.9.tar.gz
drwxr-xr-x 2 root root 4096 Aug 4 14:12 sentinel-26390
drwxr-xr-x 2 root root 4096 Aug 4 14:12 sentinel-26391
drwxr-xr-x 2 root root 4096 Aug 4 14:12 sentinel-26392
drwxr-xr-x 2 root root 4096 Aug 4 16:03 slaver-6380
drwxr-xr-x 2 root root 4096 Aug 4 16:04 slaver-6381
[centos@VM-0-15-centos redisSentinel]#
```

- 可直接到 Redis 官网下载最新稳定包，地址：<https://redis.io/download>。或者使用 命令下载安装包。

```
sudo wget http://download.redis.io/releases/redis-6.0.9.tar.gz
```

- 解压

将redis-6.0.9.tar.gz解压到 redisSentinel 目录中

```
tar -zxf redis-6.0.9.tar.gz -C redisSentinel
```

make 编译

- 进入解压文件目录使用make命令对解压的 redis 文件进行编译，如下

```
cd redis-6.0.9/
make
```

注意：如过出现以下错误，是因为没有安装 gcc 环境或者gcc 版本过低，所以在编译过程中会报错。

```
}
^
make[1]: *** [server.o] Error 1
make[1]: Leaving directory `/data/redisSentinel/redis-6.0.9'
make: *** [all] Error 2
```

自 redis 6.0.0 之后，编译 redis 需要支持 C11 特性，C11 特性在 4.9 中被引入。Centos 7 默认 gcc 版本为 4.8.5，所以需要升级gcc版本。

- 解决方式

```
yum -y install gcc gcc-c++ make tcl
yum -y install centos-release-scl
yum -y install devtoolset-9-gcc devtoolset-9-gcc-c++ devtoolset-9-binutils
scl enable devtoolset-9 bash
```

注意：以上命令中scl命令启用只是临时的，退出或重启后就会恢复到原来的gcc版本。如果要长期生效的话，执行如下命令

```
echo "source /opt/rh/devtoolset-9/enable" >> /etc/profile
```

升级gcc完成后,重新执行make，编译完成之后，会生成对应的src、conf等文件

```
make[1]: Leaving directory `/data/redisSentinel/redis-6.0.9/src'
[root@VM-0-15-centos redis-6.0.9]# ll
total 300
-rw-rw-r-- 1 root root 108806 Oct 27 2020 00-RELEASENOTES
-rw-rw-r-- 1 root root 51 Oct 27 2020 BUGS
-rw-rw-r-- 1 root root 2499 Oct 27 2020 CONTRIBUTING
-rw-rw-r-- 1 root root 1487 Oct 27 2020 COPYING
drwxrwxr-x 6 root root 4096 Aug 4 14:32 deps
-rw-rw-r-- 1 root root 11 Oct 27 2020 INSTALL
-rw-rw-r-- 1 root root 151 Oct 27 2020 Makefile
-rw-rw-r-- 1 root root 6888 Oct 27 2020 MANIFESTO
-rw-rw-r-- 1 root root 21099 Oct 27 2020 README.md
-rw-rw-r-- 1 root root 84841 Oct 27 2020 redis.conf
-rwxrwxr-x 1 root root 275 Oct 27 2020 runtest
-rwxrwxr-x 1 root root 280 Oct 27 2020 runtest-cluster
-rwxrwxr-x 1 root root 795 Oct 27 2020 runtest-moduleapi
-rwxrwxr-x 1 root root 281 Oct 27 2020 runtest-sentinel
-rw-rw-r-- 1 root root 10744 Oct 27 2020 sentinel.conf
drwxrwxr-x 3 root root 12288 Aug 4 14:38 src
drwxrwxr-x 11 root root 4096 Oct 27 2020 tests
-rw-rw-r-- 1 root root 3055 Oct 27 2020 TLS.md
drwxrwxr-x 9 root root 4096 Oct 27 2020 utils
[root@VM-0-15-centos redis-6.0.9]# cd src/
[root@VM-0-15-centos src]# ll
```

- 编译成功后，进入src文件夹，执行make install进行Redis安装。

```
cd src/
make install
```

```
-rw-r--r-- 1 root root 41680 Aug 4 14:37 zmalloc.o
[root@VM-0-15-centos src]# make install

Hint: It's a good idea to run 'make test' ;)

INSTALL install
INSTALL install
INSTALL install
INSTALL install
INSTALL install
[root@VM-0-15-centos src]# cd ../
[root@VM-0-15-centos redis-6.0.9]#
```

配置主从复制

将 redis.conf 复制三份到 master-6379, slaver-6380, master-6381 目录下

```
cp -r redis-6.0.9/redis.conf /data/redisSentinel/master-6379
cp -r redis-6.0.9/redis.conf /data/redisSentinel/slaver-6380
cp -r redis-6.0.9/redis.conf /data/redisSentinel/slaver-6381
```

修改配置

```
vim redis.conf
```

Master 与Slaver的配置主要在于端口号不同，并且在Slaver末尾添加上 `replicaof`

▪ Master : 127.0.0.1 6379

```
# master 端口
port 6379
# 默认绑定的是回环地址，默认不能被其他机器访问
# bind 127.0.0.1
# 后台执行 将`daemonize`由`no`改为`yes`
daemonize yes
#redis以守护进程方式运行时，系统默认把pid写入/var/run/redis.pid，可以通过pidfile指定pid文件
pidfile /var/run/redis_6379.pid
# 是否开启保护模式，由yes该为no
protected-mode no
```

Redis3.2版本后新增protected-mode配置，默认是yes，即开启。设置外部网络连接redis服务，

设置方式如下：

- 关闭protected-mode模式，此时外部网络可以直接访问
- 开启protected-mode保护模式，需配置bind ip或者设置访问密码

▪ Slaver: 127.0.0.1 6380

```
# master 端口
port 6380
# 默认绑定的是回环地址，默认不能被其他机器访问
# bind 127.0.0.1
# 后台执行 将`daemonize`由`no`改为`yes`
daemonize yes
#redis以守护进程方式运行时，系统默认把pid写入/var/run/redis.pid，可以通过pidfile指定pid文件
pidfile /var/run/redis_6380.pid
# 是否开启保护模式，由yes该为no
protected-mode no

replicaof 42.192.90.13 6379
```

- **Slaver: 127.0.0.1 6381**

```
# master 端口
port 6381
# 默认绑定的是回环地址，默认不能被其他机器访问
# bind 127.0.0.1
# 后台执行 将`daemonize`由`no`改为`yes`
daemonize yes
#redis以守护进程方式运行时，系统默认把pid写入/var/run/redis.pid，可以通过pidfile指定pid文件
pidfile /var/run/redis_6381.pid
# 是否开启保护模式，由yes该为no
protected-mode no

replicaof 127.0.0.1 6379
```

启动 Redis

- **通过 redis-server 启动主从节点。**


```
./redis-6.0.9/src/redis-server /data/redisSentinel/master-6379/redis.conf
./redis-6.0.9/src/redis-server /data/redisSentinel/slaver-6380/redis.conf
```

```
./redis-6.0.9/src/redis-server /data/redisSentinel/slaver-6381/redis.conf
```

▪ 检查集群状态

```
./redis-6.0.9/src/redis-cli -p 6479 info Replication
```

```
[root@VM-0-15-centos redisSentinel]# ./redis-6.0.9/src/redis-cli -p 6379 info Replication
# Replication
role:master
connected_slaves:2
slave0:ip=127.0.0.1,port=6380,state=online,offset=14,lag=1
slave1:ip=127.0.0.1,port=6381,state=online,offset=14,lag=1
master_replid:17fec3cc1c354af16aa603c9c7569ba703fed5c
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:14
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:14
[root@VM-0-15-centos redisSentinel]#
```

 星河之码

配置哨兵集群

将哨兵配置文件分别复制到 `sentinel26380 sentinel26381 sentinel26382`，需要注意的是每个文件的端口配置以及 `sentinel monitor mymaster 172.16.90.152 6479 2` 中最后的数字 2，哨兵集群汇总每个节点必须一致。

将哨兵配置文件 `sentinel.conf` 分别复制到 `sentinel-26390, sentinel-26391, sentinel-26392` 目录下

```
cp -r redis-6.0.9/sentinel.conf /data/redisSentinel/sentinel-26390
cp -r redis-6.0.9/sentinel.conf /data/redisSentinel/sentinel-26391
cp -r redis-6.0.9/sentinel.conf /data/redisSentinel/sentinel-26392
```

进入各个文件目录下修改配置

```
vim sentinel.conf
```

在配置过程中要注意 `sentinel monitor mymaster 127.0.0.1 6379 2` 中的 `quorum` 数量 2，`quorum` 的作用在上一篇文章 [Redis 哨兵模式之实现原理篇](#) 有说明。

▪ Sentinel_1:127.0.0.1 26390

以下部分配置为默认配置，没有特殊要求不需要修改

```
# 哨兵sentinel实例运行的端口 默认26379
port 26379

# 将`daemonize`由`no`改为`yes` 后台执行
daemonize yes

# 哨兵sentinel监控的redis主节点的 ip port
# master-name 可以自己命名的主节点名字 只能由字母A-z、数字0-9 、这三个字符".-_"组成。
# quorum 当这些quorum个数sentinel哨兵认为master主节点失联 那么这时 客观上认为主节点失联了
# sentinel monitor <master-name> <ip> <redis-port> <quorum>
sentinel monitor mymaster 127.0.0.1 6379 2

# 当在Redis实例中开启了requirepass foobared授权密码 则所有连接Redis实例的客户端都要提供密码
# 设置哨兵sentinel 连接主从的密码 注意必须为主从设置一样的验证密码
# sentinel auth-pass <master-name> <password>
#sentinel auth-pass mymaster MySUPER--secret-0123password
sentinel auth-pass mymaster 123456

# 指定多少毫秒之后 主节点没有应答哨兵sentinel 此时 哨兵主观上认为主节点下线 默认30秒，改成3秒
#sentinel down-after-milliseconds <master-name> <milliseconds>
sentinel down-after-milliseconds mymaster 3000

# 这个配置项指定了在发生failover主备切换时最多可以有多少个slave同时对新的master进行 同步，
#这个数字越小，完成failover所需的时间就越长，
#但是如果这个数字越大，就意味着越 多的slave因为replication而不可用。
#可以通过将这个值设为 1 来保证每次只有一个slave 处于不能处理命令请求的状态。
# sentinel parallel-syncs <master-name> <numslaves>
sentinel parallel-syncs mymaster 1

# 故障转移的超时时间 failover-timeout 可以用在以下这些方面：
#1. 同一个sentinel对同一个master两次failover之间的间隔时间。
#2. 当一个slave从一个错误的master那里同步数据开始计算时间。直到slave被纠正为向正确的master同步
#3.当想要取消一个正在进行的failover所需要的时间。
#4.当进行failover时，配置所有slaves指向新的master所需的最大时间。不过，即使过了这个超时，
#slaves依然会被正确配置为指向master，但是就不按parallel-syncs所配置的规则来了
# 默认三分钟
# sentinel failover-timeout <master-name> <milliseconds>
sentinel failover-timeout mymaster 180000
```



```
# 这个文件会自动生成（如果同一台服务器上启动，注意要修改为不同的端口）
pidfile /var/run/redis-sentinel-26390.pid
```

▪ Sentinel_1:127.0.0.1 26391

```
port 26391
daemonize yes
sentinel monitor mymaster 127.0.0.1 6379 2
sentinel auth-pass mymaster 123456
pidfile /var/run/redis-sentinel-26391.pid
```

▪ Sentinel_1:127.0.0.1 26392

```
port 26392
daemonize yes
sentinel monitor mymaster 127.0.0.1 6379 2
sentinel auth-pass mymaster 123456
pidfile /var/run/redis-sentinel-26392.pid
```

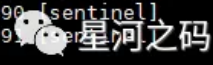
启动哨兵集群

```
./redis-6.0.9/src/redis-sentinel sentinel-26390/sentinel.conf
./redis-6.0.9/src/redis-sentinel sentinel-26391/sentinel.conf
./redis-6.0.9/src/redis-sentinel sentinel-26392/sentinel.conf
```

查看启动状态:

```
ps -ef | grep redis
```

```
[root@VM-0-15-centos redis-6.0.9]# ps -ef |grep redis
root      4560      1   0 16:05 ?        00:00:05 ./redis-6.0.9/src/redis-server 127.0.0.1:6379
root      4566      1   0 16:05 ?        00:00:05 ./redis-6.0.9/src/redis-server 127.0.0.1:6380
root      4576      1   0 16:05 ?        00:00:05 ./redis-6.0.9/src/redis-server 127.0.0.1:6381
root     23938      1   0 17:40 ?        00:00:03 ./redis-6.0.9/src/redis-sentinel *:26392 [sentinel]
root     24890      1   0 17:45 ?        00:00:02 ./redis-6.0.9/src/redis-sentinel *:26390 [sentinel]
root     24896      1   0 17:45 ?        00:00:02 ./redis-6.0.9/src/redis-sentinel *:26391 [sentinel]
root     28359 18909   0 18:02 pts/0    00:00:00 grep --color=auto redis
```



收录于话题 #Redis·8个

下一篇 · Redis 哨兵模式之实现原理篇

People who liked this content also liked

Redis 哨兵模式之执行过程解析篇

星河之码

重新认识redis(一)

一起来了解数据库啊

Redis (基础)

我是一名程序源