**Redis高可用部署**

**一、Redis Sentinel简介**

Redis Sentinel是redis自带的集群管理工具，主要功能有:

·监控（Monitoring）： Redis Sentinel实时监控主服务器和从服务器运行状态。

·提醒（Notification）：当被监控的某个 Redis 服务器出现问题时， Redis Sentinel 可以向系统管理员发送通知， 也可以通过 API 向其他程序发送通知。

·自动故障转移（Automatic failover）： 当一个主服务器不能正常工作时，Redis Sentinel 可以将一个从服务器升级为主服务器， 并对其他从服务器进行配置，让它们使用新的主服务器。当应用程序连接到 Redis 服务器时， Redis Sentinel会告之新的主服务器地址和端口。

Redis Sentinel 是一个分布式系统， 你可以在架构中运行多个 Sentinel 进程，这些进程通过相互通讯来判断一个主服务器是否断线，以及是否应该执行故障转移。

在配置Redis Sentinel时，至少需要有1个Master和1个Slave。当Master失效后,Redis Sentinel会报出失效警告，并通过自动故障转移将Slave提升为Master，并提供读写服务;当失效的Master恢复后，Redis Sentinel会自动识别，将Master自动转换为Slave并完成数据同步。

通过Redis Sentinel可以实现Redis零手工干预并且短时间内进行M-S切换，减少业务影响时间。

**二、硬件需求**

为了预防单节点故障，需要至少两台服务器，配置要求一致。

|  |  |  |
| --- | --- | --- |
| CPU | 内存 | 磁盘 |
| >2 CORES | >16 GB | >100 GB |

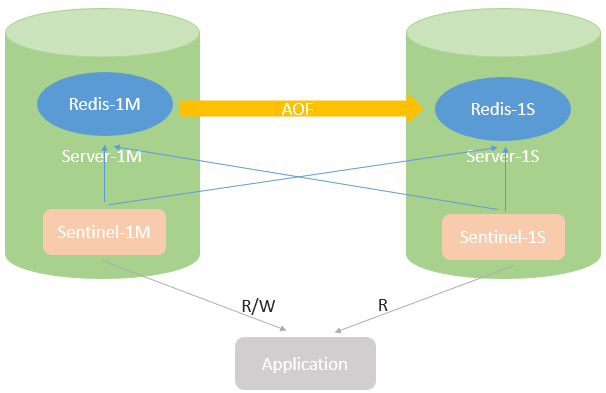
**三、拓扑结构**

在两个服务器中分别都部署Redis和Redis Sentinel。当Master中的Redis出现故障时（Redis进程终止、服务器僵死、服务器断电等），由Redis Sentinel将Master权限切换至Slave Redis中，并将只读模式更改为可读可写模式。应用程序通过Redis Sentinal确定当前Master Redis位置，进行重新连接。

根据业务模式，可以制定两种拓扑结构：单M-S结构和双M-S结构。如果有足够多的服务器，可以配置多M-S结构。

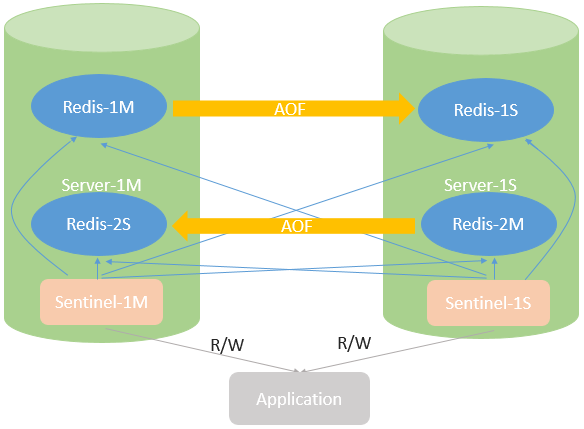
## 1、单M-S结构

单M-S结构特点是在Master服务器中配置Master Redis（Redis-1M）和Master Sentinel（Sentinel-1M）。Slave服务器中配置Slave Redis（Redis-1S）和Slave Sentinel（Sentinel-1S）。其中 Master Redis可以提供读写服务，但是Slave Redis只能提供只读服务。因此，在业务压力比较大的情况下，可以选择将只读业务放在Slave Redis中进行。

**[](http://photo.blog.sina.com.cn/showpic.html#blogid=75ad98f30101fwqj&url=http://album.sina.com.cn/pic/0029C0wjgy6EeWuOujOd4)**

## 2、双M-S结构

双M-S结构的特点是在每台服务器上配置一个Master Redis，同时部署一个Slave Redis。由两个Redis Sentinel同时对4个Redis进行监控。两个Master Redis可以同时对应用程序提供读写服务，即便其中一个服务器出现故障，另一个服务器也可以同时运行两个Master Redis提供读写服务。缺点是两个Master redis之间无法实现数据共享，不适合存在大量用户数据关联的应用使用。

**[](http://photo.blog.sina.com.cn/showpic.html#blogid=75ad98f30101fwqj&url=http://album.sina.com.cn/pic/0029C0wjgy6EeWx6TvYba)**

## 3、优劣对比

两个结构各有优缺点，分别适用于不同的应用场景：

单M-S结构适用于不同用户数据存在关联，但应用可以实现读写分离的业务模式。Master主要提供写操作，Slave主要提供读操作，充分利用硬件资源。

双（多）M-S结构适用于用户间不存在或者存在较少的数据关联的业务模式，读写效率是单M-S的两（多）倍，但要求故障时单台服务器能够承担两个Mater Redis的资源需求。

**四、 配置部署**

单M-S结构和双M-S结构配置相差无几，下面以双M-S结构配置为例。

## 1、Redis配置

1)Master Redis配置

在Server-1M上配置Redis-1M

# vi redis-1M.conf

## master redis-1M

## daemonize默认为no，修改为yes，启用后台运行

daemonize yes

# Redis 默认pid 文件位置redis.pid

#当运行多个 redis 服务时，需要指定不同的 pid 文件和端口

pidfile redis-1M.pid

##端口号

port 6379

##验证口令

requirepass \*\*\*\*\*\*\*\*\*\*\*\*\*

masterauth \*\*\*\*\*\*\*\*\*\*\*\*\*

#绑定可连接Redis的IP地址,不设置将处理所有请求

# bind 127.0.0.1

#客户端连接的超时时间,单位为秒,超时后会关闭连接（0为不设置）

timeout 0

#日志记录等级

loglevel notice

#设置数据库的个数

databases 16

#日志刷新策略（Master禁用）

#save 900 1

#save 300 10

#save 60 10000

#是否使用压缩镜像备份

rdbcompression yes

#镜像备份文件的文件名

dbfilename redis-1M\_dump.rdb

#镜像备份路径，默认值为 ./

dir /redis/backup

#设置该数据库为其他数据库的从数据库,主库无需设置

#slaveof

# slaveof

#指定与主数据库连接时需要的密码验证，主库无需设置

#masterauth

#masterauth

#如果 slave-serve-stale-data 设置成 'no'，slave会返回"SYNC with master in #progress"错误信息，但 INFO 和SLAVEOF命令除外。

slave-serve-stale-data yes

#客户端连接访问口令

# requirepass foobared

#限制同时连接的客户数量，防止过多的client导致内存耗尽。如果有足够内存可以不进行#设置

#maxclients 10000

#设置redis能够使用的最大内存。

# maxmemory

##启用增量（Master禁用）

appendonly no

#增量日志文件名，默认值为appendonly.aof

appendfilename appendonly.aof

#设置对 appendonly.aof 文件进行同步的频率

#always 表示每次有写操作都进行同步,everysec 表示对写操作进行累积,每秒同步一次。

#no表示等操作系统进行数据缓存同步到磁盘，都进行同步,everysec 表示对写操作进行累#积,每秒同步一次

appendfsync everysec

#是否重置Hash表

#设置成yes后redis将每100毫秒使用1毫秒CPU时间来对redis的hash表重新hash，##可降低内存的使用。当使用场景有较为严格的实时性需求,不能接受Redis时不时的对请##求有2毫秒的延迟的话，把这项配置为no。如果没有这么严格的实时性要求,可以设置为 #yes,能够尽可能快的释放内存。

activerehashing yes

##Slave开启只读模式

slave-read-only yes

在Server-1S上配置Redis-2M

# vi redis-2M.conf

## master redis-2M

# Redis 默认pid 文件位置redis.pid

#当运行多个 redis 服务时，需要指定不同的 pid 文件和端口

pidfile redis-2M.pid

#镜像备份文件的文件名

dbfilename redis-1M\_dump.rdb

#日志刷新策略（Slave启用）

save 900 1

save 300 10

save 60 10000

#-----------------其他参数与redis-1M保持一致-----------------

daemonize yes

#……

2)Slave Redis配置

在Server-1S上配置Redis-1S

# Redis 默认pid 文件位置redis.pid

#当运行多个 redis 服务时，需要指定不同的 pid 文件和端口

pidfile redis-1S.pid

##端口号

port 7379

#镜像备份文件的文件名

dbfilename redis-1S\_dump.rdb

#设置该数据库为其他数据库的从数据库,主库无需设置

Slaveof server-1M 6379

##启用增量（Master禁用）

appendonly yes

#-----------------其他参数与redis-1M保持一致-----------------

daemonize yes

#……

在Server-1M上配置Redis-2S

# Redis 默认pid 文件位置redis.pid

#当运行多个 redis 服务时，需要指定不同的 pid 文件和端口

pidfile redis-2S.pid

##端口号

port 7379

#镜像备份文件的文件名

dbfilename redis-2S\_dump.rdb

#设置该数据库为其他数据库的从数据库,主库无需设置

Slaveof server-1S 6379

#-----------------其他参数与redis-2M保持一致-----------------

daemonize yes

#……

## 2、Redis Sentinel配置

在Server-1M上配置Sentinel-1M

vi sentinel-1M.conf

#Master redis-1M

#hostname server-1M

#ip 192.168.84.129

#端口号

port 26379

sentinel monitor server-1M 192.168.84.129 6379 1

sentinel down-after-milliseconds server-1M 5000

sentinel failover-timeout server-1M 900000

sentinel can-failover server-1M yes

sentinel parallel-syncs server-1M 1

#Master redis-1S

#hostname server-1S

#ip 192.168.84.128

sentinel monitor server-1S 192.168.84.128 6379 1

sentinel down-after-milliseconds server-1S 5000

sentinel failover-timeout server-1S 900000

sentinel can-failover server-1S yes

sentinel parallel-syncs server-1S 1

在Server-1S上配置Sentinel-1S

#Master redis-1M

#hostname server-1M

#ip 192.168.84.128

#端口号

port 27379

sentinel monitor server-1M 192.168.84.129 6379 1

sentinel down-after-milliseconds server-1M 5000

sentinel failover-timeout server-1M 900000

sentinel can-failover server-1M yes

sentinel parallel-syncs server-1M 1

#Master redis-1S

#hostname server-1S

#ip 192.168.84.128

sentinel monitor server-1S 192.168.84.128 6379 1

sentinel down-after-milliseconds server-1S 5000

sentinel failover-timeout server-1S 900000

sentinel can-failover server-1S yes

sentinel parallel-syncs server-1S 1

## 3、启动服务

Server-1M启动redis

redis-server redis-1M.conf

redis-server redis-2M.conf

Server-1S启动redis

redis-server redis-1S.conf

redis-server redis-2Sd.conf

检测同步

Master日志检查：

[28162] 10 Nov 23:32:11.810 \* DB loaded from append only file: 0.000 seconds

[28162] 10 Nov 23:32:11.810 \* The server is now ready to accept connections on port 6379

[28162] 10 Nov 23:32:35.360 \* Slave ask for synchronization

[28162] 10 Nov 23:32:35.360 \* Starting BGSAVE for SYNC

[28162] 10 Nov 23:32:35.361 \* Background saving started by pid 28170

[28170] 10 Nov 23:32:35.373 \* DB saved on disk

[28170] 10 Nov 23:32:35.375 \* RDB: 0 MB of memory used by copy-on-write

[28162] 10 Nov 23:32:35.437 \* Background saving terminated with success

[28162] 10 Nov 23:32:35.438 \* Synchronization with slave succeeded

Slave日志检查：

[28091] 10 Nov 23:27:15.908 \* DB loaded from append only file: 0.001 seconds

[28091] 10 Nov 23:27:15.908 \* The server is now ready to accept connections on port 6389

[28091] 10 Nov 23:27:15.944 \* Connecting to MASTER...

[28091] 10 Nov 23:27:15.947 \* MASTER <-> SLAVE sync started

[28091] 10 Nov 23:27:15.951 \* Non blocking connect for SYNC fired the event.

[28091] 10 Nov 23:27:15.952 \* Master replied to PING, replication can continue...

[28091] 10 Nov 23:27:15.990 \* MASTER <-> SLAVE sync: receiving 50 bytes from master

[28091] 10 Nov 23:27:15.990 \* MASTER <-> SLAVE sync: Loading DB in memory

[28091] 10 Nov 23:27:15.991 \* MASTER <-> SLAVE sync: Finished with success

[28091] 10 Nov 23:27:15.993 \* Background append only file rewriting started by pid 28094

[28094] 10 Nov 23:27:15.998 \* SYNC append only file rewrite performed

[28094] 10 Nov 23:27:15.998 \* AOF rewrite: 0 MB of memory used by copy-on-write

[28091] 10 Nov 23:27:16.047 \* Background AOF rewrite terminated with success

[28091] 10 Nov 23:27:16.047 \* Parent diff successfully flushed to the rewritten AOF (0 bytes)

[28091] 10 Nov 23:27:16.048 \* Background AOF rewrite finished successfully

启动sentinel服务

redis-server sentinel-1M.conf --sentinel

[28156] 10 Nov 23:33:22.337 \* +slave slave 192.168.84.129:6389 192.168.84.129 6389 @server-1S 192.168.84.128 6379

[28156] 10 Nov 23:39:37.071 # +sdown sentinel 192.168.84.128:26389 192.168.84.128 26389 @server-1S 192.168.84.128 6379

[28156] 10 Nov 23:39:37.072 # +sdown sentinel 192.168.84.128:26389 192.168.84.128 26389 @server-1M 192.168.84.129 6379

[28156] 10 Nov 23:40:05.438 # -sdown sentinel 192.168.84.128:26389 192.168.84.128 26389 @server-1S 192.168.84.128 6379

[28156] 10 Nov 23:40:05.438 # -sdown sentinel 192.168.84.128:26389 192.168.84.128 26389 @server-1M 192.168.84.129 6379

[28156] 10 Nov 23:40:10.262 \* -dup-sentinel masterserver-1S 192.168.84.128 6379 #duplicate of 192.168.84.128:26389 or e453a45a5e1421519dcbe00a199f203579b27b0f

[28156] 10 Nov 23:40:10.263 \* +sentinel sentinel 192.168.84.128:26389 192.168.84.128 26389 @server-1S 192.168.84.128 6379

[28156] 10 Nov 23:40:10.263 \* -dup-sentinel masterserver-1M 192.168.84.129 6379 #duplicate of 192.168.84.128:26389 or e453a45a5e1421519dcbe00a199f203579b27b0f

[28156] 10 Nov 23:40:10.263 \* +sentinel sentinel 192.168.84.128:26389 192.168.84.128 26389 @server-1M 192.168.84.129 6379

redis-server sentinel-1S.conf --sentinel

[16383] 10 Nov 23:40:05.499 \* +slave slave 192.168.84.129:6389 192.168.84.129 6389 @server-1S 192.168.84.128 6379

[16383] 10 Nov 23:40:05.500 \* +slave slave 192.168.84.128:6389 192.168.84.128 6389 @server-1M 192.168.84.129 6379

[16383] 10 Nov 23:40:06.098 \* +sentinel sentinel 192.168.84.129:26379 192.168.84.129 26379 @server-1S 192.168.84.128 6379

[16383] 10 Nov 23:40:08.404 \* +sentinel sentinel 192.168.84.129:26379 192.168.84.129 26379 @server-1M 192.168.84.129 6379

## 4、故障模拟检测

模拟redis-1M（Master 192.168.84.129 6379）故障，sentinel自动检测状态，然后切换至redis-1S（Slave 192.168.84.128 6389）,并将redis-1M转移为redis-1S的slave，状态为+sdown。

[28156] 11 Nov 00:42:48.431 # +sdown masterserver-1M 192.168.84.129 6379

[28156] 11 Nov 00:42:48.431 # +odown masterserver-1M 192.168.84.129 6379 #quorum 1/1

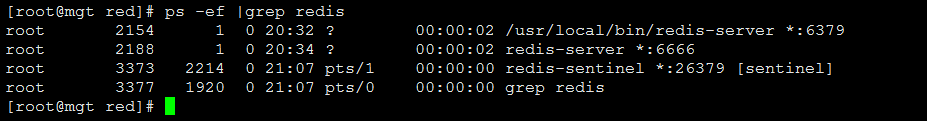
[28156] 11 Nov 00:42:56.570 # +failover-detected masterserver-1M 192.168.84.129 6379

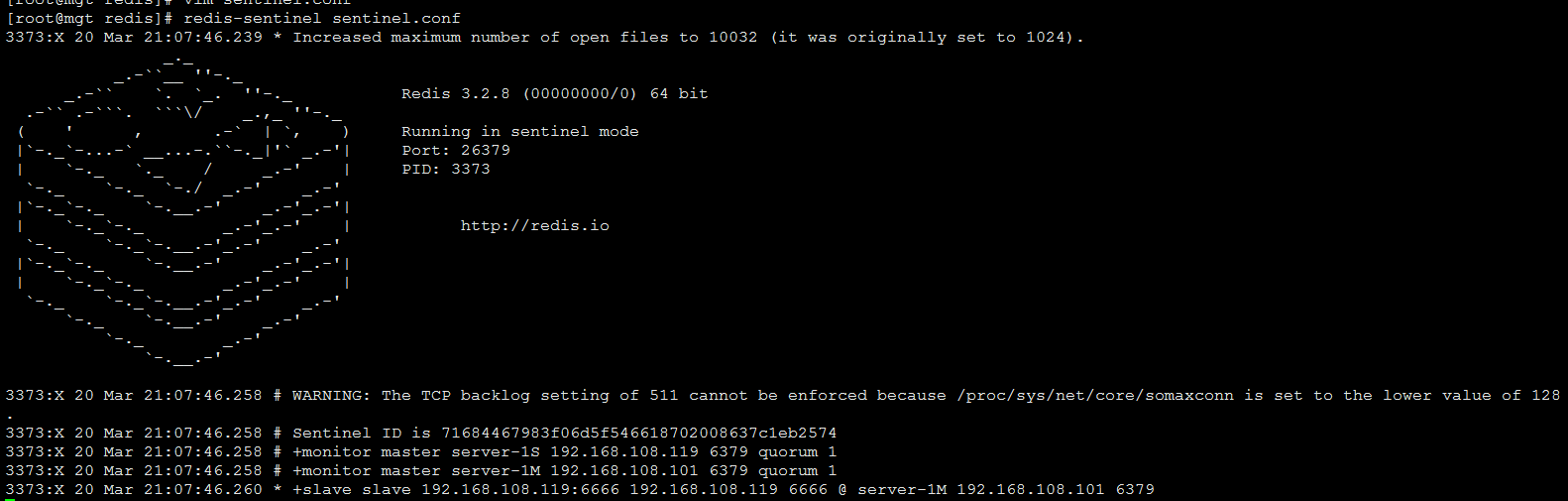
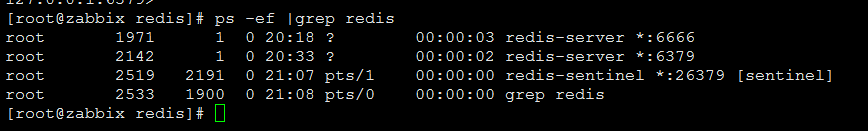
[28156] 11 Nov 00:42:56.670 # +failover-end masterserver-1M 192.168.84.129 6379

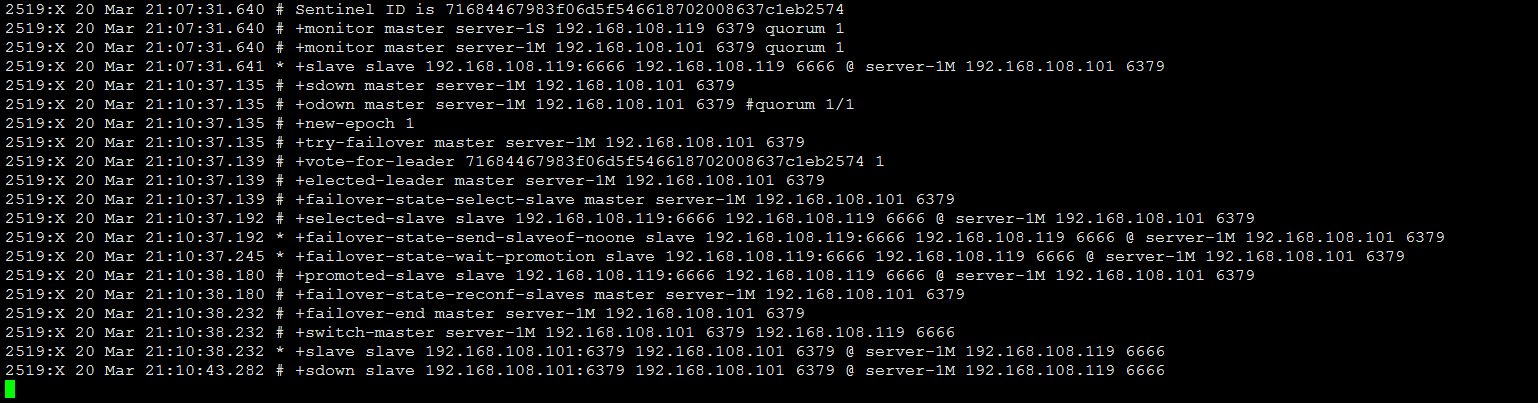
[28156] 11 Nov 00:42:56.670 # +switch-masterserver-1M 192.168.84.129 6379 192.168.84.128 6389

[28156] 11 Nov 00:42:57.055 \* +sentinel sentinel 192.168.84.128:26389 192.168.84.128 26389 @server-1M 192.168.84.128 6389

[28156] 11 Nov 00:43:01.688 # +sdown slave 192.168.84.129:6379 192.168.84.129 6379 @server-1M 192.168.84.128 6389







**五、备份恢复**

## 1、备份策略

Redis提供两种相对有效的备份方法：RDB和AOF。

RDB是在某个时间点将内存中的所有数据的快照保存到磁盘上，在数据恢复时，可以恢复备份时间以前的所有数据，但无法恢复备份时间点后面的数据。

AOF是以协议文本的方式，将所有对数据库进行过写入的命令（及其参数）记录到 AOF 文件，以此达到记录数据库状态的目的。优点是基本可以实现数据无丢失（缓存的数据有可能丢失），缺点是随着数据量的持续增加，AOF文件也会越来越大。

在保证数据安全的情况下，尽量避免因备份数据消耗过多的Redis资源，采用如下备份策略：

Master端：不采用任何备份机制

Slave端：采用AOF（严格数据要求时可同时开启RDB），每天将AOF文件备份至备份服务器。

为了最大限度减少Master端的资源干扰，将备份相关全部迁移至Slave端完成。同时这样也有缺点，当Master挂掉后，应用服务切换至Slave端，此时的Slave端的负载将会很大。目前Redis不支持RDB和AOF参数动态修改，需要重启Redis生效，希望能在新的版本中实现更高效的修改方式。

## 2、灾难恢复

当Master端Redis服务崩溃（包含主机断电、进程消失等），Redis sentinel将Slave切换为读写状态，提供生产服务。通过故障诊断修复Master，启动后会自动加入Sentinel并从Slave端完成数据同步，但不会切换。

· 当Master和Slave同时崩溃（如机房断电），启动服务器后，将备份服务器最新的AOF备份拷贝至Master端，启动Master。一切完成后再启动Slave。

**六、 运维监控**

目前针对redis的监控比较少见，主要有redis-live、dredis等。但这些工具对redis性能消耗比较严重，实时监控比较困难。

redis的监控可以简单分为安全监控和性能监控。安全监控可以通过redis sentinel的输出日志判断Master和Slave的状态；性能监控需要收集redis的性能参数进行评估。因此二者并不冲突，通过shell脚本可以实现轻量级的监控，缺点是没有可视化的实时图表。

## 1、安全监控

Redis sentinel的输出日志简洁而且内容很丰富，包含redis的实时状态、故障切换时间以及sentinel自身的状态，并且针对故障消除也有详细的记录。通过对sentinel日志挖掘，找出故障时刻进行及时报警，并通知管理员。

由于sentinel默认不启用日志记录，可以通过以下方法记录日志：

vi sentinel-1M.sh

LOG=/redis/redis-2.6.16/log/sentinel-1M.log

redis-server /redis/redis-2.6.16/sentinel-1M.conf --sentinel >>$LOG &

sentinel\_monitor

#---------------------------------------------------------------------------------------

#-- Redis sentinel log monitor --

#-- --

#-- VERSION : 1.0 Completed at 2013-11-12 --

#-- SUPPORT : redis 2.6 or later --

#-- FUNCTION:Sentinel log monitor --

#-- AUTHOR : Icecream --

#---------------------------------------------------------------------------------------

脚本内容请联系作者。

如有报错信息，通过Email通知管理员，并将日志信息写入监控日志。

通过crontab定期调用sentinel\_monitor.sh进行日志监控：

##每分钟执行一次

\* \* \* \* \* /redis/redis-2.6.16/log/sentinel\_monitor.sh >/dev/null 2>&1

或

##每五分钟执行一次

\*/5 \* \* \* \* /redis/redis-2.6.16/log/sentinel\_monitor.sh >/dev/null 2>&1

监控日志输出样例：

-------------------------------------------------------------------------------

2013-11-11 17:46:01 Sentinel Monitor

-------------------------------------------------------------------------------

IP :192.168.84.129

HOSTNAME :ice11g1

SENTINEL :sentinel-1M

ERRORS :

[30220] 11 Nov 17:39:32.557 # +sdown slave 192.168.84.129:6379 192.168.84.129 6379 @ ice11g1 192.168.84.128 6389

[30220] 11 Nov 17:45:23.388 \* +demote-old-slave slave 192.168.84.129:6379 192.168.84.129 6379 @ ice11g1 192.168.84.128 6389

[30220] 11 Nov 17:45:23.587 # -sdown slave 192.168.84.129:6379 192.168.84.129 6379 @ ice11g1 192.168.84.128 6389

[30220] 11 Nov 17:45:33.426 \* +slave slave 192.168.84.129:6379 192.168.84.129 6379 @ ice11g1 192.168.84.128 6389

-------------------------------------------------------------------------------

-------------------------------------------------------------------------------

2013-11-11 17:47:01 Sentinel Monitor

-------------------------------------------------------------------------------

IP :192.168.84.129

HOSTNAME :ice11g1

SENTINEL :sentinel-1M

ERRORS :

OK,no error in sentinel-1M.log

## 2、性能监控

redis\_monitor

#---------------------------------------------------------------------------------------

#-- Redis  monitor --

#-- --

#-- VERSION : 1.0 Completed at 2013-11-12 --

#-- SUPPORT : redis 2.6 or later --

#-- FUNCTION:redis monitor --

#-- AUTHOR : Icecream --

#---------------------------------------------------------------------------------------

脚本内容请联系作者！

脚本输出样例：

[2013-11-15 14:52:31] 9 105.54M 116338688 1.05 135.59M

[2013-11-15 14:52:36] 9 105.58M 116338688 1.05 135.59M

[2013-11-15 14:52:41] 9 105.58M 116338688 1.05 135.59M

[2013-11-15 14:52:46] 9 105.58M 116338688 1.05 135.59M

[2013-11-15 14:52:51] 9 105.58M 116338688 1.05 135.59M

[2013-11-15 14:52:56] 9 105.58M 116338688 1.05 135.59M

[2013-11-15 14:53:01] 9 105.54M 116338688 1.05 135.59M

[2013-11-15 14:53:06] 9 105.54M 116338688 1.05 135.59M

[2013-11-15 14:53:11] 10 105.56M 116338688 1.05 135.59M

[2013-11-15 14:53:16] 10 105.60M 116338688 1.05 135.59M

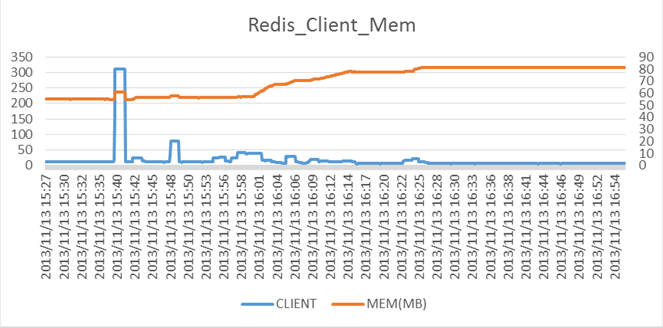
[2013-11-15 14:53:21] 10 105.56M 116338688 1.05 135.59M

[2013-11-15 14:53:26] 10 105.63M 116338688 1.05 135.59M

[2013-11-15 14:53:31] 10 105.60M 116338688 1.05 135.59M

[2013-11-15 14:53:36] 10 105.56M 116338688 1.05 135.59M

通过输出日志可以手工绘制曲线图：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=75ad98f30101fwqj&url=http://album.sina.com.cn/pic/0029C0wjgy6EeWR7AB3e1)