分布式缓存服务

常见问题

文档版本 01

发布日期 2025-02-10

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 实例类型/版本

1.1 DCS Redis 4.0 支持的新特性说明

与Redis 3.0版本相比，Redis 4.0及以上版本，除了开源Redis增加的特性之外，创建耗时也相应缩短。

实例由虚机方式改成了物理机容器化部署，创建实例只需要8~10秒时间完成。

Redis 4.0版本更新的特性，主要涉及三个方面：

1. 新命令的增加，如MEMORY、SWAPDB。
2. Lazyfree机制，延迟删除大key，降低删除操作对系统资源的占用影响。
3. 内存性能优化，即主动碎片整理。

MEMORY 命令

在Redis 3.0及之前，只能通过info memory命令了解有限的几个内存统计信息。Redis 4.0引入新的命令memory，让您能够更深入了解Redis的内存使用情况。

单机、主备、读写分离实例执行memory help命令：

```
127.0.0.1:6379[8]> memory help
1) "MEMORY DOCTOR                        - Outputs memory problems report"
2) "MEMORY USAGE <key> [SAMPLES <count>] - Estimate memory usage of key"
3) "MEMORY STATS                         - Show memory usage details"
4) "MEMORY PURGE                         - Ask the allocator to release memory"
5) "MEMORY MALLOC-STATS                  - Show allocator internal stats"
```

集群实例执行memory help命令：

```
127.0.0.1:6379[8]> memory help
1) MEMORY <subcommand> arg arg ... arg. Subcommands are:
2) DOCTOR - Return memory problems reports.
3) MALLOC-STATS -- Return internal statistics report from the memory allocator.
4) PURGE -- Attempt to purge dirty pages for reclamation by the allocator.
5) STATS -- Return information about the memory usage of the server.
6) USAGE <key> [SAMPLES <count>] -- Return memory in bytes used by <key> and its value. Nested values are sampled up to <count
> times (default: 5).
```

usage命令

输入**memory usage *[key]***，如果当前key存在，则返回key的value实际使用内存估算值；如果key不存在，则返回nil。不同Redis版本计算使用内存时可能会有差异，返回结果可能会有不同。


```
127.0.0.1:6379[8]> set dcs "DCS is an online, distributed, in-memory cache service compatible with Redis, and Memcached."
OK
127.0.0.1:6379[8]> memory usage dcs
(integer) 141
127.0.0.1:6379[8]>
```

说明

1. usage统计value内存占用，以及key自身的内存占用，不包含key的Expire内存占用。不同Redis版本，统计结果可能有差异。

   ```
   //以下内容基于Redis 5.0.2版本验证
   192.168.0.66:6379> set a "Hello, world!"
   OK
   192.168.0.66:6379> memory usage a
   (integer) 58
   192.168.0.66:6379> set abc "Hello, world!"
   OK
   192.168.0.66:6379> memory usage abc
   (integer) 60   //key名称长度变化后，内存占用也有变化，说明usage统计包含了key自身的占用
   192.168.0.66:6379> expire abc 1000000
   (integer) 1
   192.168.0.66:6379> memory usage abc
   (integer) 60    //加了过期时间后，内存占用没有改变，说明usage统计不包含expire内存占用
   192.168.0.66:6379>
   ```

2. 对hash、list、set、sorted set等数据类型，usage命令会抽样统计，提供内存占用的估算值。

   使用方式：**memory usage** *keyset* **samples** *1000*

   其中keyset表示一个集合数据类型的key，1000表示抽样个数。

stats命令

返回当前实例内存使用细节。

使用方法：**memory stats**

```
127.0.0.1:6379[8]> memory stats
 1) "peak.allocated"
 2) (integer) 2412408
 3) "total.allocated"
 4) (integer) 2084720
 5) "startup.allocated"
 6) (integer) 824928
 7) "replication.backlog"
... ...
```

以下给出部分数据返回项的具体含义。

表 1-1 memory stats

数据返回项	说明
peak.allocated	Redis实例运行过程中，allocator分配的内存峰值。同info memory的used_memory_peak
total.allocated	allocator当前分配的内存字节数。同info memory的used_memory
startup.allocated	Redis启动占用的内存字节数
replication.backlog	Redis复制积压缓冲区（replication backlog）内存使用字节数，通过repl-backlog-size参数设置，默认1M

数据返回项	说明
clients.slaves	在master侧，所有slave clients消耗的内存字节数
clients.normal	Redis所有常规客户端消耗内存字节数
overhead.total	Redis额外的总开销内存字节数； 即分配器分配的总内存total.allocated，减去数据实际存储使用内存。
keys.count	Redis实例中key的数量
keys.bytes-per-key	每个key平均占用字节数。注意，overhead也会均摊到每个key上，因此不能以此值来表示业务实际的key平均长度。
dataset.bytes	表示Redis数据占用的内存容量。即分配的内存总量，减去总的额外开销内存量。
dataset.percentage	表示Redis数据占用内存占总内存分配的百分比
peak.percentage	当前内存使用量与峰值时的占比
fragmentation	表示Redis的内存碎片率

doctor命令

使用方法： **memory doctor**

used_memory（total.allocated）小于5M，doctor认为内存使用量过小，不做进一步诊断。当满足以下某一点，Redis会给出诊断结果和建议：

1. peak分配内存大于当前total_allocated的1.5倍，即peak.allocated/total.allocated > 1.5，说明内存碎片率高，RSS远大于used_memory
2. High fragmentation/fragmentation大于1.4，说明内存碎片率高
3. 每个Normal Client平均使用内存大于200KB，说明pipeline可能使用不当，或Pub/Sub客户端处理消息不及时
4. 每个Slave Client平均使用内存大于10MB，说明master的写入流量过高

purge命令

使用方法： **memory purge**

用途：通过调用jemalloc内部命令，进行内存释放。释放对象包括Redis进程占用但未有效使用的内存，即常说的内存碎片。

说明

memory purge只适用于使用jemalloc作为allocator的Redis实例。

Lazy free 机制

解决的痛点/问题

Redis是单线程程序，当运行一个耗时较大的请求时，会导致所有请求排队等待，在请求处理完成前，Redis不能响应其他请求，因此容易引发性能问题。而Redis删除大的集合键时，就属于一种比较耗时的请求。

原理

Redis 4.0提供的一种惰性删除或者说延迟释放机制，主要用于解决删除大key对Redis进程的阻塞，从而避免带来性能与可用性问题。

删除key时，Redis异步延时释放key的内存，把key释放操作放在bio(Background I/O)单独的子线程处理中。

使用方法

1. 主动删除
   - **unlink**

     unlink与del命令目的一样，删除某个key。unlink在删除集合类键时，如果集合键的元素个数大于64个，会把内存释放操作，给单独的bio(Background I/O)线程来执行。因此unlink删除操作能在非常短的时间内完成包含上百万个元素的大key删除。
   - **flushall/flushdb**

     通过对flushall/flushdb添加ASYNC异步清理选项，Redis在清理整个实例或单个DB时，操作都是异步的。
2. 过期key删除、大key驱逐删除

   被动删除有四种场景，每种场景对应一个配置参数，默认都是关闭：

```
lazyfree-lazy-eviction no //针对redis内存使用达到maxmemory，并设置有淘汰策略时，是否采用lazy free机制
lazyfree-lazy-expire no //针对设置有TTL的键，过期后，被redis清理删除时是否采用lazy free机制
lazyfree-lazy-server-del no //针对有些指令在处理已存在的键时，会带有一个隐式的DEL键的操作
slave-lazy-flush no //针对slave进行全量数据同步，slave在加载master的RDB文件前，会运行flushall来清理自己的数据场景
```

其他新增命令

1. **swapdb**

   用途：交换同一Redis实例内2个db的数据。

   用法：**swapdb** *dbindex1 dbindex2*
2. **zlexcount**

   用途：在有序集合中，返回符合条件的元素个数。

   用法：**zlexcount** *key min max*

内存使用和性能改进

1. 使用更少的内存来存储相同数量的数据
2. 可以对使用的内存进行碎片整理，并逐渐回收

1.2 DCS Redis 5.0 支持的新特性说明

DCS的Redis 5.0版本继承了Redis 4.0版本的所有功能增强以及新的命令，同时还兼容开源Redis 5.0版本的新增特性。

Stream 数据结构

Stream是Redis 5.0引入的一种新数据类型，它是一个全新的支持多播的可持久化消息队列。

Redis Stream的结构示意图如图1-1所示，它是一个可持久化的数据结构，用一个消息链表，将所有加入进来的消息都串起来。

Stream数据结构具有以下特性：

1. Stream中可以有多个消费者组。
2. 每个消费组都含有一个Last_delivered_id，指向消费组当前已消费的最后一个元素（消息）。
3. 每个消费组可以含有多个消费者对象，消费者共享消费组中的Last_delivered_id，相同消费组内的消费者存在竞争关系，即一个元素只能被其中一个消费者进行消费。
4. 消费者对象内还维持了一个Pending_ids，Pending_ids记录已发送给客户端，但是还没完成ACK（消费确认）的元素id。
5. Stream与Redis其他数据结构的比较，见表1-2。

图 1-1 Stream 数据结构示意图

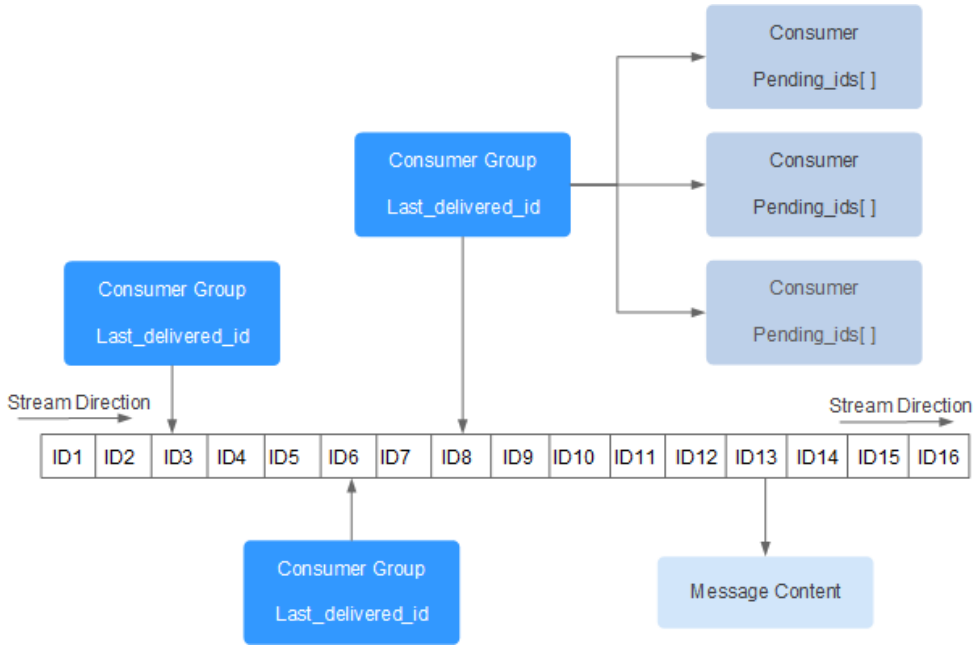


表 1-2 Stream 与 Redis 现有数据结构比较

比较项	Stream	List、Pub/Sub、Zset
复杂度	获取元素高效，复杂度为O(logN)	List获取元素的复杂度为O(N)
offset	支持offset，每个消息元素有唯一id。不会因为新元素加入或者其他元素淘汰而改变id。	List没有offset概念，如果有元素被逐出，无法确定最新的元素
持久化	支持消息元素持久化，可以保存到AOF和RDB中。	Pub/Sub不支持持久化消息。
消费分组	支持消费分组	Pub/Sub不支持消费分组

比较项	Stream	List、Pub/Sub、Zset
消息确认	支持ACK（消费确认）	Pub/Sub不支持
性能	Stream性能与消费者数量无明显关系	Pub/Sub性能与客户端数量正相关
逐出	允许按时间线逐出历史数据，支持block，给予radix tree和listpack，内存开销少。	Zset不能重复添加相同元素，不支持逐出和block，内存开销大。
删除元素	不能从中间删除消息元素。	Zset支持删除任意元素

Stream相关命令介绍

接下来按照使用流程中出现的顺序介绍**Stream相关命令**。详细命令见**表1-3**

1. 首先使用**XADD**添加流元素，即创建Stream，添加流元素时可指定消息数量最大保存范围。
2. 然后通过**XGROUP**创建消费者组。
3. 消费者使用**XREADGROUP**指令进行消费。
4. 客户端消费完毕后使用**XACK**命令确认消息已消费成功。

图 1-2 Stream 相关命令介绍

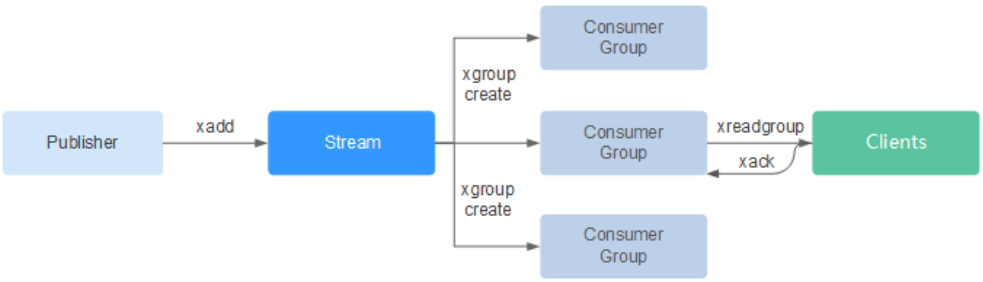


表 1-3 Stream 的详细命令

命令	说明	语法
XACK	从流的消费者组的*待处理条目列表*（简称PEL）中删除一条或多条消息。	XACK key group ID [ID ...]
XADD	将指定的流条目追加到指定key的流中。 如果key不存在，作为运行这个命令的副作用，将使用流的条目自动创建key。	XADD key ID field string [field string ...]

命令	说明	语法
XCLAIM	在流的消费者组上下文中，此命令改变待处理消息的所有权， 因此新的所有者是在命令参数中指定的消费者。	XCLAIM key group consumer min-idle-time ID [ID ...] [IDLE ms] [TIME ms-unix-time] [RETRYCOUNT count] [FORCE] [JUSTID]
XDEL	从指定流中移除指定的条目，并返回成功删除的条目的数量，在传递的ID不存在的情况下， 返回的数量可能与传递的ID数量不同。	XDEL key ID [ID ...]
XGROUP	该命令用于管理流数据结构关联的消费者组。使用XGROUP你可以： ● 创建与流关联的新消费者组。 ● 销毁一个消费者组。 ● 从消费者组中移除指定的消费者。 ● 将消费者组的*最后交付ID*设置为其他内容。	XGROUP [CREATE key groupname id-or-$] [SETID key id-or-$] [DESTROY key groupname] [DELCONSUMER key groupname consumername]
XINFO	检索关于流和关联的消费者组的不同的信息。	XINFO [CONSUMERS key groupname] key key [HELP]
XLEN	返回流中的条目数。如果指定的key不存在，则此命令返回0，就好像该流为空。	XLEN key
XPENDING	通过消费者组从流中获取数据。检查待处理消息列表的接口，用于观察和了解消费者组中哪些客户端是活跃的，哪些消息在等待消费，或者查看是否有空闲的消息。	XPENDING key group [start end count] [consumer]
XRANGE	返回流中满足给定ID范围的条目。	XRANGE key start end [COUNT count]
XREAD	从一个或者多个流中读取数据，仅返回ID大于调用者报告的最后接收ID的条目。	XREAD [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]
XREADGROUP	XREAD命令的特殊版本，指定消费者组进行读取。	XREADGROUP GROUP group consumer [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]
XREVRANGE	与XRANGE相同，但显著的区别是以相反的顺序返回条目，并以相反的顺序获取开始-结束参数	XREVRANGE key end start [COUNT count]

命令	说明	语法
XTRIM	XTRIM将流裁剪为指定数量的项目，如有需要，将驱逐旧的项目（ID较小的项目）。	XTRIM key MAXLEN [~] count

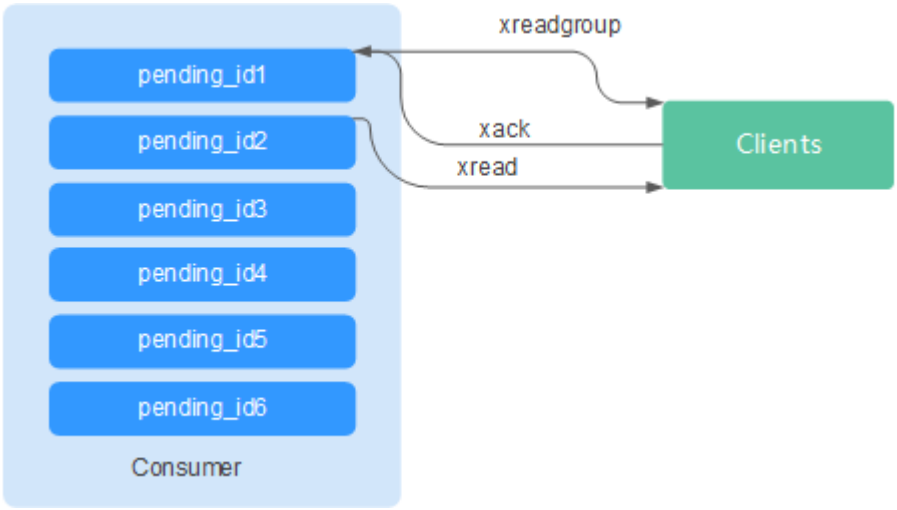
消息（流元素）消费确认

Stream与相比Pub/Sub，不仅增加消费分组模式，还支持消息消费确认。

当一条消息被某个消费者调用XREADGROUP命令读取或调用XCLAIM命令接管的时候， 服务器尚不确定它是否至少被处理了一次。 因此，**一旦消费者成功处理完一条消息，它应该调用XACK知会Stream，这样这个消息就不会被再次处理， 同时关于此消息的PEL（pending_ids）条目也会被清除，从Redis服务器释放内存**。

某些情况下，因为网络问题等，客户端消费完毕后没有调用XACK，这时候PEL内会保留对应的元素ID。待客户端重新连上后，XREADGROUP的起始消息ID建议设置为0-0，表示读取所有的PEL消息及自last_id之后的消息。同时，消费者消费消息时需要能够支持消息重复传递。

图 1-3 ACK 机制解读



内存使用优化

Redis 5.0在上一版本基础上，在内存使用上做了进一步优化。

- 主动碎片整理

  当key被频繁修改，value长度不断变化时，Redis会为key分配新的内存空间。由于Redis追求高性能，实现了自己的内存分配器来管理内存，因此并不会将原有内存释放给OS，从而导致出现内存碎片。当used_memory_rss/used_memory高于1.5，一般认为内存碎片占比过高，内存利用率低。

  因此，合理规划和使用缓存数据，规范数据写入，有助于减少内存碎片的产生。

  Redis 3.0及以下：可以通过定期重启服务解决内存碎片问题。建议实际缓存数据不超过配置可用内存的50%。

  Redis 4.0：支持主动整理内存碎片，服务在运行期间进行自动内存碎片清理。同时Redis 4.0支持通过memory purge命令手动清理内存碎片。

Redis 5.0：增强版主动碎片整理，配合Jemalloc版本更新，更快更智能，延时更低。

- HyperLogLog算法优化

  HyperLogLog是一种基数计数方法，使用少量的内存空间完成海量数据的计数统计，在Redis 5.0中，HyperLogLog算法得到改进，优化了计数统计时的内存使用效率。

  举个例子：B树计数效率非常高，但是内存消耗也比较多。而HyperLogLog可节省大量存储空间。当B树需要1M内存统计，HyperLogLog只需要1kb。

- 内存信息统计报告能力增强

  INFO命令返回信息更加详实。

命令新增和优化

1. **客户端管理增强**
   - Redis-cli支持集群管理

     在Redis 4.0以及之前版本，需要安装redis-trib模块，管理集群。

     Redis 5.0对Redis-cli做了优化，集成了集群的所有管理功能。具体使用可以通过命令**redis-cli --cluster help**查看帮助信息。
   - 优化客户端在频繁连接与中断场景下的性能

     当您的应用需要使用短连接时，这个优化价值凸显。
2. **有序集合使用更简单**

   有序集合新增两个命令：ZPOPMIN和ZPOPMAX。
   - ZPOPMIN key [count]

     删除并返回有序集合key中的最多count个具有最低得分的成员。如果返回多个成员，也会按照得分高低（value值比较），从低到高排列。
   - ZPOPMAX key [count]

     删除并返回有序集合key中的最多count个具有最高得分的成员。如果返回多个成员，也会按照得分高低（value值比较），从高到低排列。
3. **help增加更多子命令说明**

   支持help直接查看快速使用攻略，你不再需要每次登录redis.io去查找。例如，命令行输入stream使用攻略：xinfo help

```
127.0.0.1:6379> xinfo help
1) XINFO <subcommand> arg arg ... arg. Subcommands are:
2) CONSUMERS <key> <groupname>  -- Show consumer groups of group <groupname>.
3) GROUPS <key>                 -- Show the stream consumer groups.
4) STREAM <key>                 -- Show information about the stream.
5) HELP                         -- Print this help.
127.0.0.1:6379>
```

4. **Redis-cli命令输入提示**

   Redis-cli在输入完整的命令后，会展示参数提醒，帮助用户记忆命令语法格式。

   如下图所示，输入**zadd**命令，Redis-cli使用浅颜色字体显示zadd的语法。

```
# Cluster
cluster_enabled:0

# Keyspace
db0:keys=1,expires=0,avg_ttl=0
198.19.59.199:6379> zadd key [NX|XX] [CH] [INCR] score member [score member ...]
```


RDB 支持存储 LFU、LRU

Redis 5.0开始，RDB快照文件中增加存储key逐出策略**LRU**和**LFU**：

- FIFO：先进先出。最早存储的数据，优先被淘汰。
- LRU：最近最少使用。长期未使用的数据，优先被淘汰。
- LFU：最不经常使用。在一段时间内，使用次数最少的数据，优先被淘汰。

说明

Redis 5.0的RDB文件格式有变化，向下兼容。因此如果使用快照的方式迁移，可以从Redis低版本迁移到Redis 5.0，但不能从Redis 5.0迁移到低版本。

1.3 DCS Redis 6.0 支持的新特性说明

DCS的Redis 6.0版本继承了Redis 5.0版本的所有功能增强以及新的命令，同时还兼容开源Redis 6.0版本的新增特性。

RESP3 协议

在Redis 6.0中，推出了下一代Redis协议-RESP3，相比于RESP2协议，增加了一部分新的数据类型。

- **Null**：空值，替代RESP2中的*-1、$-1
- **Array**：有序集合
- **Simple string**：节省空间的安全字符串（非二进制）
- **Blob string**：二进制格式的安全字符串
- **Simple error**：节省空间的安全错误码/错误信息（非二进制）
- **Blob Error**：二进制格式的安全错误码/错误信息
- **Boolean**：True/False，布尔类型
- **Number**：有符号的64位整数
- **Big Number**：大数字类型
- **Double**：浮点数
- **Verbatim string**：二进制格式的安全字符串，带文本格式
- **Map**：无序的键值对
- **Set**：无序的不重复元素集合
- **Attribute**：属性键值对，类似于Map
- **PUSH**：带外数据，类似于Array，用于Redis服务端主动向客户端推送数据
- **Hello**：hello命令返回的响应类型，用于客户端、服务端建立连接时使用

说明

如需使用RESP3协议，需要保证客户端SDK支持RESP3协议，否则在建立连接时，与服务端通过hello通信协商使用的协议依旧是RESP2协议。

客户端缓存

Redis 6.0中通过TRACKING模块实现了主动通知客户端刷新缓存的机制，根据协议类型，实现方式如下：

RESP3

- 普通模式
- 广播模式

RESP2

- 转发模式

开启客户端缓存通知的格式如下：

```
CLIENT TRACKING ON|OFF [REDIRECT client-id] [PREFIX prefix] [BCAST] [OPTIN][OPTOUT] [NOLOOP]
```

在RESP3协议中，主要是借助了PUSH类型的消息来实现服务端的主动推送通知。在普通模式中，Redis会记住每个客户端请求的key，当该key所对应的value发生变化时，将会发送失效消息（invalidation message）通知对应的客户端集合，但对于每个客户端仅会通知一次，即使后续该key所对应的value有其他操作改动，除非客户端在接收到失效消息后，再次通过读取该key的方式开启通知。开启普通模式的track功能命令如下：

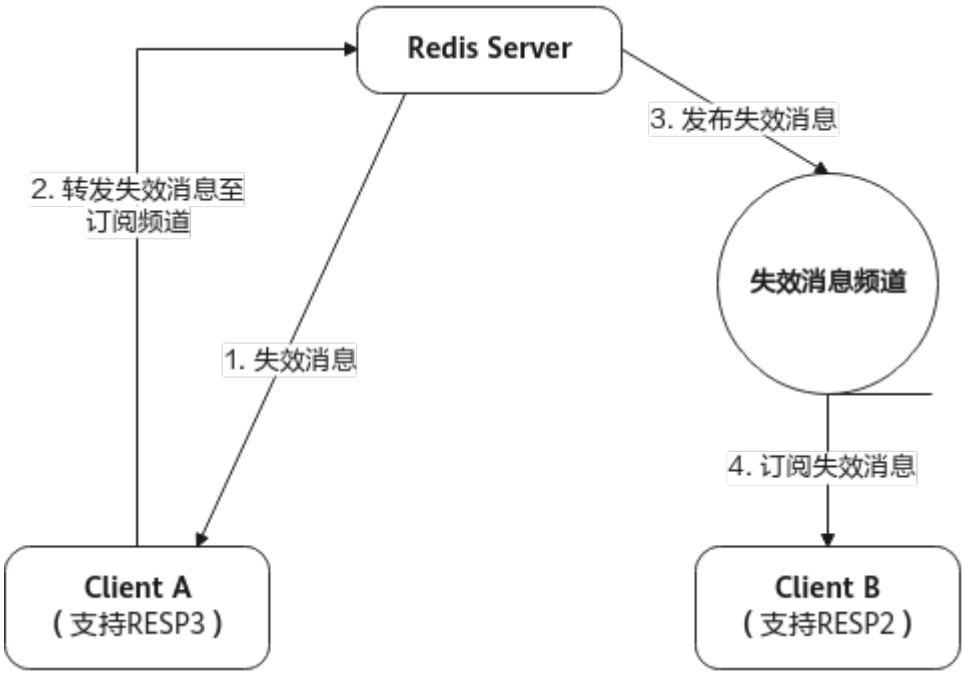
```
CLIENT TRACKING ON
```

对于广播模式，则根据所track的key prefix来决定在符合key prefix的key所对应的value有所变化时，通知给所有的客户端，如key prefix所匹配的key数量较多，或改动较多，将会导致服务端发送大量的失效广播消息，消耗网络带宽。开启广播模式的track功能命令如下:

```
CLIENT TRACKING ON BCAST PREFIX key-prefix
```

如客户端SDK不支持RESP3协议，只能采用RESP2协议的转发模式来实现客户端缓存主动更新通知，需要准备一个专门支持RESP3协议的客户端来作为中转节点，转发来自Redis的失效消息（invalidation message）至特定的订阅频道。工作原理如下:

图 1-4 工作原理



支持 SSL

Redis 6.0开始支持SSL/TLS方式的加密连接及加密传输，可通过在服务控制台上开启SSL服务，生成实例的SSL/TLS证书及密钥，在使用客户端连接时，指定该证书/密钥即可，连接示例如下：

```
redis-cli --tls --cert /etc/redis/ssl/redis.crt --key /etc/redis/ssl/redis.key --cacert /etc/redis/ssl/redis.crt
```

详情请参见：**SSL设置**。

RDB 加载速度优化

在Redis 6.0中，针对RDB文件的实际组成部分，做了对应的加载优化，相比于之前的加载方式，可以获得大概20%~30%的速度提升 。

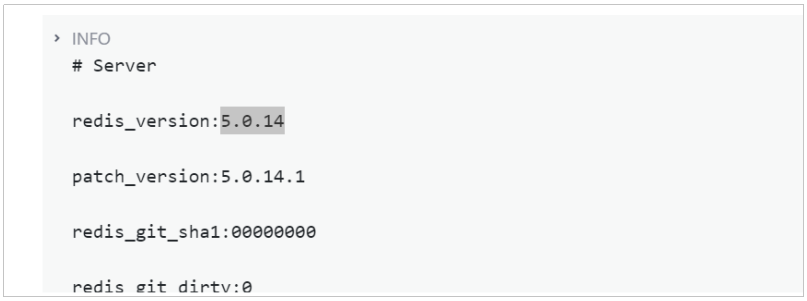
INFO 命令优化

针对INFO命令的处理做了相关优化，尤其针对大量客户端连接场景，性能消耗及时延上有较大改进。

1.4 如何查询 Redis 实例的原生版本

连接需要查询的实例，执行info命令：

图 1-5 查询实例信息



1.5 Redis 的 Arm 和 x86 实例有什么差异?

当前华为云DCS Redis实例的CPU架构支持Arm计算和x86计算。Arm Redis和x86的Redis两者功能是一样的。对客户端使用来说完全一样，没有额外的适配工作量。

可能在部分复杂的命令，大key且命令复杂度超过O(N)场景下，x86 Redis单CPU能力会强于Arm Redis。

综合来看，Arm Redis和x86 Redis能力和性能相当，都能满足业务需求，Arm Redis价格略低，请根据需要进行选择。

说明

部分Region已停售“Arm计算”类型，推荐使用“x86计算”类型。

2 实例特性

2.1 DCS 实例的 CPU 规格是怎么样的

Redis基础版：

使用DCS Redis基础版实例的用户无需关心CPU规格的指标，仅需关心QPS，带宽，内存大小等核心指标。

Redis基础版的实例基于开源Redis构造，开源Redis使用单个主线程处理命令，只能利用一个核的CPU，因此，只需认为单个Redis节点仅使用1核CPU即可。提升基础版Redis实例的内存大小，CPU规格不变。

Redis基础版由于社区版单线程处理模型的限制，**如需增加实例CPU处理性能，请使用集群类型的Redis实例，通过增加分片的方式，来增加整个集群的处理性能。**集群实例每个节点默认分配1核CPU进行处理。

Redis企业版：

DCS Redis企业版为多线程版本，企业版Redis线程数请参考**表2-1**。企业版Redis的CPU最大使用率=线程数x100%，例如线程数为3的企业版Redis，最大CPU使用率为300%。

表 2-1 企业版 Redis 线程数

实例内存规格	线程数（企业版高性能型主备实例）	线程数（企业版存储型主备实例）
8 GB	3	3
16 GB	4	6
32 GB	4	7
64 GB	4	-

2.2 如何理解分片数与副本数?

什么是分片

分片也叫**条带**，指Redis集群的一个管理组，对应一个redis-server进程。一个Redis集群由若干条带组成，每个条带负责若干个slot（槽），数据分布式存储在slot中。Redis集群通过条带化分区，实现超大容量存储以及并发连接数提升。

每个集群实例由多个分片组成，每个分片默认为一个双副本的主备实例。分片数等于实例中主节点的个数。

什么是副本

副本指缓存实例的**节点**，包含主节点和备节点。单副本表示实例没有备节点，双副本表示实例有备节点（一个主节点，一个备节点）。例如主备实例的副本数设置为3时，表示该实例有1个主节点，2个备节点。

不同实例类型的副本和分片数

- **单机实例：**单机实例只有1个节点，1个Redis进程，当Redis进程故障后，DCS为实例重新拉起一个新的Redis进程。
- **主备/读写分离实例：**分片数为1，包含一个主节点，一个或多个备节点。当主节点出现故障时，会进行主备倒换，恢复业务。
- **集群实例：**集群实例由多个分片组成，每个分片默认是一个双副本的主备实例。例如一个3分片，2副本的集群实例，则每个分片都有2个节点（1个主节点，1个备节点）。

实例类型	分片数	副本数	负载均衡	占用IP数
单机	单分片	单副本，不支持多副本	-	1个
主备	单分片	默认双副本，支持配置为2-10副本 企业版主备实例仅支持2副本	不支持	占用IP个数=副本数
读写分离	单分片	默认双副本，支持2-6副本	支持	1个
Proxy集群	多分片	双副本，不支持其他副本数	支持	1个
Cluster集群	多分片	默认双副本，支持配置为1-5副本	不支持	占用IP个数=副本数*分片数

2.3 Redis 实例支持的单个 Key 和 Value 数据大小是否有限制?

- Key的大小上限为512M。

  建议key的大小不超过1KB，这样既节约存储空间，也利于Redis进行检索。
- String类型的value值上限为512M。
- 集合、链表、哈希等key类型，单个元素的value上限为512M。

  事实上，集合、链表、哈希都可以看成由String类型的key按照一定的映射关系组合而成。

同时，请注意避免对大Value进行长时间高并发写入，这样会影响网络传输效率，也会增加redis-server的内部处理耗时，从而导致请求时延较大。

2.4 Redis 集群可以读取每个节点的 IP 地址吗?

Redis 3.0版本的集群实例（Proxy版本）的使用方式与单机、主备实例相同，无需知晓后端地址。

Redis 4.0及以上版本的集群实例（Cluster版本）可以使用**cluster nodes**命令获取。

redis-cli -h *{redis_address}* -p *{redis_port}* -a *{redis_password}* cluster nodes

在命令返回的结果中，获取所有master节点的IP端口，如下图所示。

```
[root@ecs-[redacted]54-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a [redacted]3 cluster nodes
fb75f0743af4695a3d241ff7790b2f508e4985ff 192.168.0.140:6379@16379 myself,master - 0 1562144170000 3 connected
d112bae791b2bbd9602fe32963536b8a0db9eb79 192.168.0.61:6379@16379 master - 0 1562144171524 1 connected 0-5460
73e2f8fe196166f9ad1283361867d24c136413f0 192.168.0.194:6379@16379 master - 0 1562144170000 2 connected 5461-10
40d72299fde6045de0f79ee4b97910b505acbc6a 192.168.0.231:6379@16379 slave 73e2f8fe196166f9ad1283361867d24c13641
be6c07faa64d724323e0d7cedc3f38346dcbd212 192.168.0.80:6379@16379 slave fb75f0743af4695a3d241ff7790b2f508e4985f
c16b9acaeed7dd0721f129596cd43bd499c0e396 192.168.0.169:6379@16379 slave d112bae791b2bbd9602fe32963536b8a0db9eb
```

2.5 DCS Redis 集群实例是否支持原生集群?

当前DCS Redis 3.0版本支持Proxy集群，Redis 4.0和5.0版本支持原生集群（Cluster集群）和Proxy集群，Redis 6.0基础版实例支持原生集群（Cluster集群）。

2.6 Redis 实例的数据逐出策略是什么?

逐出指将数据从缓存中删除，以腾出更多的存储空间容纳新的缓存数据，详情请参见官网**逐出策略**。Redis实例支持在配置运行参数中**查看或修改Redis实例使用的逐出策略**。

Redis 实例支持的逐出策略

在达到内存上限（maxmemory）时，Redis支持选择以下8种数据逐出策略：

- noeviction：在这种策略下，如果缓存达到了配置的上限，实例将不再处理客户端任何增加缓存数据的请求，比如写命令，实例直接返回错误给客户端。缓存达到上限后，实例只处理删除和少数几个例外请求。

- allkeys-lru：根据LRU（Least recently used，最近最少使用）算法尝试回收最少使用的键，使得新添加的数据有空间存放。
- volatile-lru：根据LRU（Least recently used，最近最少使用）算法尝试回收最少使用的键，但仅限于具有“expire”字段集的键，使得新添加的数据有空间存放。
- allkeys-random：回收随机的键使得新添加的数据有空间存放。
- volatile-random：回收随机的键使得新添加的数据有空间存放，但仅限于具有“expire”字段集的键。
- volatile-ttl：回收具有“expire”字段集中的键，且优先回收存活时间（TTL）较短的键，使得新添加的数据有空间存放。
- allkeys-lfu：从所有键中驱逐最不常用的键。
- volatile-lfu：从具有“expire”字段集的所有键中驱逐最不常用的键。

说明

- 当没有键满足回收前提条件时，数据逐出策略volatile-lru、volatile-random、volatile-ttl与noeviction策略相同，具体见上文noeviction介绍。
- 2020年7月之前创建的Redis实例，逐出策略默认为noeviction。2020年7月及之后创建的实例，逐出策略默认为volatile-lru。

查看或修改 Redis 实例使用的逐出策略

Redis实例支持通过修改maxmemory-policy参数配置，查看及修改实例的数据逐出的策略。

2.7 Redis 实例是否支持读写分离?

Redis实例支持读写分离的情况如下表所示：

实例类型	是否支持读写分离
读写分离实例	支持。 **说明** 读写分离功能，推荐使用**读写分离实例**，无需在客户端做任何配置。
Redis Cluster集群实例	支持从客户端实现读写分离，需要在客户端做配置，参考**配置说明**。
Redis 4.0/5.0/6.0基础版主备实例	支持从客户端实现读写分离，需要在客户端增加用户读写请求判断。
其他版本及实例类型	不支持。

配置说明

- **Redis Cluster集群实例**，使用**cluster nodes**查询所有主备节点，客户端连接备节点，并在节点上做配置，开启备节点只读访问，从而实现读写分离。

  查询集群节点命令如下：

  ```
  redis-cli -h {redis_address} -p {redis_port} -a {redis_password} cluster nodes
  ```

  从节点配置只读模式，请参考**READONLY命令**。
- **Redis 4.0/5.0/6.0基础版版本主备实例**，在控制台的实例详情信息页面，域名区分可读写域名和只读域名，分别对应主节点和备节点，在客户端增加用户读写请求判断，如果是写请求，则将请求发送给读写域名，如果是读请求，则将请求发送给只读域名。
- **读写分离实例**，默认为从服务端侧实现的读写分离，通过Proxy节点识别用户读写请求，如果是写请求，则转发给主节点，如果是读请求，则转发给备节点，不需要用户在客户端做任何配置。

2.8 Redis 实例是否支持多 DB 方式?

Redis实例支持多DB方式的情况如下：

- Redis单机、读写分离和主备缓存实例支持多数据库（多DB），默认256个，DB编号为0-255。默认使用的是DB0。多数据库主要用于数据隔离，每个数据库的大小不是平均分配，可能会出现一个数据库将实例的内存完全占用的情况。
- Redis Proxy集群默认只有一个DB。
  - 如需创建多DB的Proxy集群实例请参考**如何创建多DB的Proxy集群实例**。
  - 创建单DB的Proxy集群实例后，如需开启多DB的操作请参考**Proxy集群使用多DB限制**。

    说明

    Redis 3.0 proxy不支持开启多DB。
- Redis Cluster集群实例不支持多DB，只有一个DB，即DB0。

DB的个数不支持修改，每个DB的大小也不支持自定义。

2.9 DCS 是否支持外部扩展模块、插件或者 Module?

DCS Redis不支持加载外部扩展模块、插件和Module。DCS后续也没有Module的规划。

2.10 Redis 实例支持数据持久化吗？开启持久化有什么影响?

是否支持持久化

单机：不支持持久化。

主备、读写分离和集群（单副本集群除外）：支持持久化。

Redis 实例支持的持久化方式

- Redis实例默认仅支持AOF的方式进行持久化，同时支持客户自行开关数据持久化配置。创建的实例（单机或单副本集群除外）默认开启AOF持久化。
- Redis实例默认不支持RDB持久化，因此也无法支持客户自行配置save参数。如果需要进行RDB持久化，可以使用主备或者集群实例的备份恢复功能，备份恢复时，Redis 4.0及以上版本实例，可以支持选择生成RDB持久化文件并且自动转储到OBS中。

持久化的磁盘是什么类型

Redis 4.0及以上版本的实例，持久化的磁盘是SSD类型。

开启/关闭 AOF 持久化的影响

开启AOF持久化后，由于Redis-Server进程需要在AOF文件中记录对应的操作信息，用来进行数据持久化。开启持久化可能存在的影响：

- 当出现底层计算节点磁盘硬件故障或者IO故障时，可能会造成时延冲高或者主备倒换等情况发生。
- Redis-Server进程会定期进行AOF重写操作，重写期间可能会造成短暂的时延冲高，AOF重写规则请参考**AOF文件在什么情况下会被重写?**。

如果在缓存场景下使用DCS实例进行应用加速，建议可以关闭持久化参数以获得更高的性能和稳定性。

关闭持久化需根据实际业务慎重操作，关闭持久化后在极端故障场景（例如主备节点同时故障等）下可能出现缓存数据丢失的问题。

如何开启/关闭 Redis 持久化

在实例的配置参数中将appendonly参数设置为no即可关闭AOF持久化，设置为yes即开启AOF持久化。（单机实例不支持持久化）

配置参数的操作请参考**修改DCS实例配置参数**。

Redis 实例是否支持开启主节点不持久化，仅从节点持久化

DCS Redis 4.0/5.0/6.0基础版的主备和集群实例、企业版高性能型主备实例，可以通过将实例参数appendonly设置为only-replica，开启实例仅从节点持久化。

其他版本和实例类型暂不支持该特性。

说明

- appendonly参数默认只有yes和no两个选项，如需设置为only-replica，需要联系运维人员放通该参数的白名单。
- 仅从节点持久化相对于主从节点同时持久化，因主节点减少了AOF写入和重写的影响，性能会有所提升，但可靠性会有所降低，请根据实际业务情况选择。

2.11 Redis 实例支持存储的数据个数有限制吗?

Redis对数据存储的个数没有限制。Redis所存储的数据只需要在实例内存范围内即可。

3 安全性

3.1 如何配置安全组

由于Redis 3.0/Memcached和Redis 4.0/Redis 5.0/Redis 6.0实例的部署模式不一样，DCS在控制访问缓存实例的方式也不一样，差别如下：

- Redis 3.0/Memcached/Redis 6.0企业版：通过配置安全组访问规则控制，不支持白名单功能。安全组配置操作请参考本章节操作。
- Redis 4.0/Redis 5.0/Redis 6.0 基础版：不支持安全组，只支持通过白名单控制。白名单配置操作，请参考**管理实例白名单**。

本节主要介绍**VPC内访问**DCS Redis 3.0/Memcached/Redis 6.0企业版实例，和通过**公网访问**DCS Redis 3.0实例时如何配置安全组。

VPC 内访问 Redis 3.0/Memcached/Redis 6.0 企业版实例

为避免跨VPC访问导致时延增大影响DCS缓存实例性能，建议客户端部署在与DCS缓存实例处于相同虚拟私有云（VPC）和相同子网的弹性云服务器（ECS）上。

除了建议ECS、DCS缓存实例处于相同VPC之外，还需要他们的安全组分别配置了正确的规则，客户端才能访问DCS缓存实例。

- 如果ECS、DCS缓存实例配置相同的安全组，安全组创建后，默认包含组内网络访问不受限制的规则。
- 如果ECS、DCS缓存实例配置了不同安全组，可参考如下配置方式：

  说明

  - 假设ECS、DCS缓存实例分别配置了安全组：sg-ECS、sg-DCS。
  - 以Redis 3.0访问端口6379为例，其它实例请以实际情况为准。
  - 以下规则，远端可使用安全组，也可以使用具体的IP地址。

  a. 配置ECS所在安全组。

  ECS所在安全组需要增加如下出方向规则，以保证客户端能正常访问DCS缓存实例。如果出方向规则不受限，则不用添加。

b. 配置DCS缓存实例所在安全组。

DCS实例所在安全组需要增加如下入方向规则，以保证能被客户端访问。

须知

缓存实例的入方向规则中，源地址建议使用指定IP地址，慎用“0.0.0.0/0”，避免绑定相同安全组的弹性云服务器遭受Redis漏洞攻击。

通过公网访问 Redis 3.0 实例

DCS缓存实例安全组配置了正确的规则，客户端才能访问DCS缓存实例。

假设DCS缓存实例安全组为sg-DCS，则需要配置如下**入方向**规则：

协议为TCP，源IP为0.0.0.0/0，或者指定客户端地址。SSL加密功能开启时，端口配置为36379；SSL加密功能关闭时，端口配置为6379。如下图所示。

图 3-1 安全组规则（以端口配置为 36379 为例）

迁移任务安全组说明

- 使用DCS在线迁移时，创建迁移任务需要选择安全组，迁移任务所选安全组的“出方向规则”需要放通源端Redis和目标端Redis的IP和端口（默认情况下安全组出方向是全部放通的，则无需再单独放通）。

- 使用DCS的备份迁移时，会使用到“default”安全组，需确认“default”安全组的“出方向规则”全部放通（默认情况下安全组出方向是全部放通的，则无需再单独放通）。

图 3-2 迁移任务安全组“出方向规则”

3.2 Redis 4.0/5.0/6.0 基础版实例为什么不支持安全组?

目前Redis 4.0/5.0/6.0基础版实例是基于VPC Endpoint，不支持安全组。

如果需要指定的IP地址才能访问Redis 4.0、Redis 5.0和Redis 6.0基础版实例，您需要将指定的IP地址加入到实例白名单中。

如果实例没有添加任何白名单或停用白名单功能，所有与实例所在VPC互通的IP地址都可以访问该实例。

创建白名单分组

步骤1 登录分布式缓存服务管理控制台。

步骤2 在管理控制台左上角单击 ，选择实例所在的区域。

步骤3 单击左侧菜单栏的“缓存管理”，进入实例信息页面。

步骤4 单击需要创建白名单的DCS缓存实例名称，进入该实例的基本信息页面。

步骤5 选择“实例配置 > 白名单配置”进入白名单配置页面，单击“创建白名单分组”。

步骤6 在弹出的“创建白名单分组”页面，设置“分组名”和“IP地址/地址段”。

表 3-1 创建白名单参数说明

参数名称	参数说明	示例
分组名	实例的白名单分组名称。每个实例支持创建4组白名单。 分组名的命名规范： • 需以字母开头。 • 长度范围在4到64位之间。 • 只能包含字母、数字、中划线、下划线。	DCS-test

参数名称	参数说明	示例
IP地址/地址段	添加允许访问实例的IP地址/地址段。每个实例最多可以添加100个IP地址/地址段。多个IP地址/地址段之间用“,”分隔。 不支持的IP和地址段：0.0.0.0和0.0.0.0/0	10.10.10.1,10.10.10.10,192.168.0.0/16

步骤7 设置完成后，单击“确定”。

创建成功后，白名单功能立即生效，只有该分组白名单中的IP地址才允许访问实例。对于已有的长连接，需重新连接后生效。

说明

- 修改白名单：在白名单列表，单击“编辑”，可以修改该白名单分组下的IP地址/地址段。
- 删除白名单：在白名单列表，单击“删除”，可以删除该白名单分组。
- 停用白名单：单击白名单列表左上角的“停用白名单”。停用白名单后，所有与实例VPC相通的IP都能访问该实例，单击“启用白名单”，可以重启白名单。

----结束

3.3 Redis 的安全加固方面有哪些建议?

在众多开源缓存技术中，Redis无疑是目前功能最为强大，应用最多的缓存技术之一，但是原生Redis版本在安全方面非常薄弱，很多地方不满足安全要求，如果暴露在公网上，极易受到恶意攻击，导致数据泄露和丢失。

针对DCS的Redis实例，您在使用过程中，可参考如下建议：

- 网络连接配置
  a. 敏感数据加密后存储在Redis实例，且实例不开启公网访问。
     对于敏感数据，尽量加密后存储。如无特殊需要，尽量不使用公网访问。
  b. 对安全组设置有限的、必须的允许访问规则。
     安全组与VPC均是用于网络安全访问控制的配置，以端口最少放开原则配置安全组规则，降低网络入侵风险。
  c. 客户端应用所在ECS设置防火墙。
     客户端应用所在的服务器建议配置防火墙过滤规则。
  d. 设置实例访问密码。
  e. 配置实例白名单。
- Redis-cli使用
  a. 隐藏密码
     安全问题：通过在redis-cli指定-a参数，密码会被ps出来，属于敏感信息。
     解决方案：修改Redis源码，在main方法进入后，立即隐藏掉密码，避免被ps出来。
  b. 禁用脚本通过sudo方式执行

安全问题： redis-cli访问参数带密码敏感信息，会被ps出来，也容易被系统记录操作日志。

解决方案：改为通过API方式（Python可以使用redis-py）来安全访问，禁止通过sudo方式切换到dbuser账号使用redis-cli。

3.4 Redis 实例是否支持 SSL 加密传输?

Redis 6.0基础版实例SSL默认关闭，如需开启SSL加密传输，请参考。

当前DCS Redis 3.0实例在开启公网访问时，支持基于Stunnel的客户端与服务端TLS加密传输（参考文档：**Stunnel安装和配置**）。在开启公网访问时，指定的CA会为每个实例生成唯一的服务证书。客户端可以使用从服务控制台上下载的CA根证书，并在连接实例时提供该证书，对实例服务端进行认证并达到加密传输的目的。

Redis 4.0/5.0不支持SSL加密传输，仅支持明文传输。

3.5 如何修改 Redis 3.0 实例的 SSL 开关?

在公网开启时，SSL加密默认开启。

如果需要关闭SSL加密，建议按照以下操作执行：

1. 打开修改公网访问页面。

2. 在“修改公网访问”页面，关闭“SSL加密”，单击“确定”。

3. 在实例详情页面的连接信息区域，显示“SSL加密”为已关闭。关闭SSL加密操作完成。

3.6 DCS 实例是否支持跨可用区部署?

Redis主备/集群/读写分离实例、Memcached主备实例支持跨可用区（AZ）部署。

- 当主备或者集群或者读写分离实例进行跨可用区部署时，如果其中一个可用区故障，另一个可用区的节点不受影响。备节点会自动升级为主节点，对外提供服务，从而提供更高的容灾能力。
- 实例跨可用区部署时，主备节点之间同步效率与同AZ部署相比基本无差异。

3.7 连接实例必须使用密码吗？如何获取密码?

- Redis实例支持密码模式和免密模式。Redis本身支持不设置密码，客户端可以直接连接Redis缓存服务并使用，但出于安全考虑，建议尽量选用密码模式，通过密码来鉴权验证，提升安全性。若选用密码模式，您需要在创建实例时自定义密码。
- Memcached实例支持密码模式和免密模式，用户可以根据自身应用特点选用支持文本和二进制协议的任何Memcached客户端。若选用密码模式，您需要在创建实例时自定义密码。
- 如需修改Redis访问方式、修改或重置密码，请参考**密码管理**。

3.8 哨兵原理

Sentinel 概览

Redis Sentinel为Redis实现高可用。实际使用中，您可以使用Sentinel帮助Redis在无需人工干预的情况下抵御某些类型的故障，Redis Sentinel还能够完成其他辅助任务，如监控、通知和客户端配置。详细介绍可参考**Redis官网**。

Sentinel 原理

Redis Sentinel是一个分布式系统，Sentinel的设计基础在于多个Sentinel进程协同工作，这样做的好处有：

1. 只有当多个哨兵一致同意某主节点不可用，才执行故障检测，这能够降低误报的可能性。
2. 即使有些Sentinel进程故障，Sentinel系统也能正常工作，从而抵御故障。

从更大范围来看，Sentinel加上Redis主从节点以及连接到Sentinel和Redis的客户端，整体也构成一个更大的分布式系统。

Sentinel 功能

- 监控：Sentinel不间断地检查主从节点是否都在正常工作。
- 通知：如果Redis中某节点故障，Sentinel可以通过API通知系统管理员或其他计算机程序。
- 自动故障切换：如果主节点异常，Sentinel启动故障切换，将一个从节点升主，其他从节点从新的主节点进行复制，并通知使用该Redis的应用程序使用新地址进行连接。

- 客户端配置来源：Sentinel充当客户端服务发现的权威来源。客户端连接到Sentinel，请求当前负责特定业务的Redis主节点地址。如果发生故障切换，Sentinels将下发新地址。

DCS 如何使用 Sentinel

Sentinel对用户不可见，仅在服务内部中使用。

关于DCS是否支持哨兵模式接入，请参考：**DCS是否支持哨兵模式接入？**

3.9 DCS 是否支持哨兵模式接入？

Redis 4.0及以上版本：

单机、主备、Cluster集群实例不支持哨兵模式接入。

读写分离支持哨兵模式和集群模式接入。

Proxy集群实例默认支持集群模式接入，开启cluster-sentinel-enabled参数后，支持哨兵模式接入。开启cluster-sentinel-enabled参数（配置为yes）的操作，请参考**修改DCS实例配置参数**。

Redis 3.0实例不支持哨兵模式接入。

4 客户端和网络连接

4.1 DCS 实例支持公网访问吗?

- Redis 3.0实例

  目前**只有Redis 3.0版本密码模式的实例支持公网访问**，且可选择是否通过SSL方式来访问DCS缓存实例。建议使用前先下载CA证书，并使用CA证书来验证DCS缓存实例的证书，以确保缓存数据的安全。具体可参考**公网连接Redis缓存实例**。

- Redis 4.0及以上版本实例

  Redis 4.0及以上版本的实例，暂不支持直接绑定公网，开启密码访问模式的单机、主备、读写分离、Proxy集群实例支持通过ELB实现公网访问，开启公网访问的操作，请参考**开启Redis 4.0/5.0/6.0公网访问并获取公网访问地址**。

  Cluster集群实例不支持公网访问。

- Memcached实例

  暂不支持公网访问，您必须通过同一虚拟私有云下的弹性云服务器来访问缓存实例，以确保缓存数据的安全。如果您在应用开发调试阶段，可以通过ssh代理方式，实现本地环境访问实例。具体操作参考**使用SSH隧道代理机制实现公网访问DCS实例**。

4.2 Redis 连接失败问题排查和解决

概述

本章节主要描述Redis连接过程出现的问题，以及解决方法。

问题分类

当您发现与Redis实例连接出现异常时，可以根据本文的内容，从以下几个方面进行排查。

- **Redis和ECS之间的连接问题**
- **公网连接Redis 3.0**
- **密码问题**

- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**
- **性能问题导致连接超时**

Redis 和 ECS 之间的连接问题

客户端所在的ECS必须和Redis实例在同一个VPC内，并且需要确保ECS和Redis之间可以正常连接。

- 如果是Redis 3.0或企业版实例，Redis和ECS的安全组没有配置正确，连接失败。

  解决方法：配置ECS和Redis实例所在安全组规则，允许Redis实例被访问。具体配置，可以参考**配置安全组**。
- 如果是Redis 4.0/5.0/6.0基础版实例，开启了白名单功能，连接失败。

  如果实例开启了白名单，在使用客户端连接时，需要**确保客户端IP是否在白名单内**，如果不在白名单，会出现连接失败。具体配置操作，可以参考**配置白名单**。客户端IP如果有变化，需要将变化后的IP加入白名单。
- Redis实例和ECS不在同一个Region。

  解决方法：不支持跨Region访问，可以在ECS所在的Region创建Redis实例，创建时注意选择与ECS相同VPC，创建之后，使用**数据迁移**进行迁移，将原有Redis实例数据迁移到新实例中。
- Redis实例和ECS不在同一个VPC。

  不同的VPC，网络是不相通的，不在同一VPC下的ECS是无法访问Redis实例。可以通过创建VPC对等连接，将两个VPC的网络打通，实现跨VPC访问Redis实例。

  关于创建和使用VPC对等连接，请参考**VPC对等连接说明**。

公网连接 Redis 3.0

在进行公网访问时，请先仔细阅读**公网连接**章节，**检查实例是否满足公网访问的要求**。

- 连接时提示：Error: Connection reset by peer或者出现：远程主机强迫关闭一个现有的连接。
  - 原因1：安全组没有配置正确。

    解决方法：需要允许Redis实例被访问，具体配置操作和公网连接操作，请按照**公网连接**章节中的操作进行。
  - 原因2：查看Redis所在vpc子网是否被ACL关联，同时这个ACL出方向被限制了。若是，放开限制。
  - 原因3：开启了SSL加密传输，连接时没有安装配置Stunnel，直接使用了界面提示的IP地址进行连接。

    解决方法：开启SSL加密时，必须安装配置Stunnel客户端，具体操作，请按照**公网连接Redis实例（开启SSL加密）**执行。其中，请注意，**在连接Redis实例命令中，IP地址需要配置为Stunnel客户端地址和端口，不要使用控制台展示的Redis实例公网连接地址和端口**。
- 已经开启了公网访问的Redis实例，公网访问被关闭了，无法使用公网访问。

  原因：该Redis实例绑定的弹性公网IP被解绑，导致Redis实例公网被关闭。

  解决方法：在控制台重新开启实例的公网访问，绑定弹性公网IP，并重新连接。

密码问题

密码输入错误时，端口可以连接上，但鉴权认证会失败。如果忘记了密码，可以**重置密码**。

实例配置问题

连接Redis时存在拒绝连接，可登录分布式缓存服务控制台，进入实例详情页面，调整实例参数maxclients的配置，具体操作可参考**修改配置参数**。

客户端连接问题

- 在使用Redis-cli连接Cluster集群时，连接失败。

  解决方法：请检查连接命令是否加上**-c**，在连接Cluster集群节点时务必使用正确连接命令。

  - Cluster集群连接命令：

    ./redis-cli -h *{dcs_instance_address}* -p 6379 -a {password} -c

  - 单机、主备、Proxy集群连接命令：

    ./redis-cli -h *{dcs_instance_address}* -p 6379 -a {password}

  具体连接操作，请参考**Redis-cli连接**。

- 出现Read timed out或Could not get a resource from the pool。

  解决方法：

  - 排查是否使用了keys命令，keys命令会消耗大量资源，造成Redis阻塞。建议使用scan命令替代，且避免频繁执行。
  - 排查实例是否是Redis 3.0，Redis 3.0底层用的是sata盘，当Redis数据持久化即AOF时，会触发偶现的磁盘性能问题，导致连接异常，可更换Redis实例为4.0及以上版本，其底层是ssd盘，磁盘性能更高，或若不需要持久化可关闭AOF。

- 出现unexpected end of stream错误，导致业务异常。

  解决方法：

  - Jedis连接池调优，建议参考**Jedis参数配置建议**进行配置连接池参数。
  - 排查是否大key较多，建议根据**优化大key**排查优化。

- 连接断开。

  解决方法：

  - 调整应用超时时间。
  - 优化业务，避免出现慢查询。
  - 建议使用scan命令替代keys命令。

- Jedis连接池问题，请参考**使用Jedis连接池报错如何处理？** 。

带宽超限导致连接问题

当实例已使用带宽达到实例规格最大带宽，可能会导致部分Redis连接超时现象。

您可以查看监控指标“流控次数”，统计周期内被流控的次数，确认带宽是否已经达到上限。

然后，检查实例是否有大Key和热Key，如果存在大Key或者单个Key负载过大，容易造成对于单个Key的操作占用带宽资源过高。大Key和热Key操作，请参考**分析实例大Key和热Key**。

性能问题导致连接超时

使用了keys等消耗资源的命令，导致CPU使用率超高；或者实例没有设置过期时间、没有清除已过期的Key，导致存储的数据过多，一直在内存中，内存使用率过高等，这些都容易出现访问缓慢、连接不上等情况。

- 建议客户改成scan命令或者禁用keys命令。
- 查看监控指标，并配置对应的告警。监控项和配置告警步骤，可查看**必须配置的监控告警**。

  例如，可以通过监控指标“内存利用率”和“已用内存”查看实例内存使用情况、“活跃的客户端数量”查看实例连接数是否达到上限等。
- 检查实例是否存在大Key和热Key。

  DCS控制台提供了大Key和热Key的分析功能，具体使用，请参考**分析Redis实例的大Key和热Key**。

4.3 DCS 实例是否支持跨 VPC 访问?

跨VPC访问，即客户端和实例不在同一个VPC。

对于未开启公网访问的实例，一般情况下，不同VPC间网络不互通，不在同一VPC下的弹性云服务器无法访问DCS缓存实例。

可以通过创建VPC对等连接，将两个VPC的网络打通，实现跨VPC访问DCS缓存实例。

用户通过VPC对等访问DCS缓存实例时，除了满足VPC对等网跨VPC访问的约束之外，还存在如下约束：

- 当创建实例时使用了172.16.0.0/12~24网段时，客户端不能在192.168.1.0/24、192.168.2.0/24、192.168.3.0/24网段。
- 当创建实例时使用了192.168.0.0/16~24网段时，客户端不能在172.31.1.0/24、172.31.2.0/24、172.31.3.0/24网段。
- 当创建实例时使用了10.0.0.0/8~24网段时，客户端不能在172.31.1.0/24、172.31.2.0/24、172.31.3.0/24网段。

关于创建和使用VPC对等连接，请参考**VPC对等连接说明**。

4.4 Redis 公网访问所需弹性 IP 是否收费?

公网访问Redis 3.0实例所需的弹性公网IP需要收取一定的费用。

在开启公网访问开关前，您首先需要创建一个弹性公网IP。具体收费规则，请参考**弹性公网IP收费说明**。

4.5 Redis 连接时报错：“(error) NOAUTH Authentication required”。

报错信息是指实例设置了免密访问。连接时不输入密码，即可避免上述错误。

4.6 客户 Http 的 Server 端关闭导致 Redis 访问失败

原因分析：客户端使用长连接，或者连接池，用完后关闭与DCS实例的连接，再次使用时，出现报错。

解决方案：使用长连接或连接池，用完后不要关闭连接；如果发现连接中断，请重新建连。

4.7 客户端出现概率性超时错误

针对低概率超时错误，是Redis使用的正常现象。Redis使用受到网络传输、客户端设置超时时间等因素影响，可能出现单个请求超时问题。

建议客户业务编码时，具备重试操作，提升业务的可靠性，避免低概率的单次请求失败时业务失败。

当出现了连接超时问题时，可以优先检查Redis是否开启了AOF持久化功能，并根据业务需求，决定是否开启AOF持久化（**开启/关闭AOF持久化的影响**）。关闭AOF持久化可以提升客户端连接的稳定性，减少出现阻塞，连接不上的情况。

如果出现超时错误概率频繁，请联系技术服务人员。

4.8 使用 Jedis 连接池报错如何处理?

在使用Jedis连接池JedisPool模式下，比较常见的报错如下：

```
redis.clients.jedis.exceptions.JedisConnectionException: Could not get a resource from the pool
```

首先确认DCS缓存实例是正常运行中状态，然后按以下步骤进行排查。

步骤1 检查网络。

1. 核对IP地址配置。

   检查jedis客户端配置的IP地址是否与DCS缓存实例的连接地址或IP地址一致，如果是通过公网访问，则检查是否与DCS缓存实例绑定的弹性IP地址一致，不一致则修改一致后重试。

2. 测试网络。

   在客户端使用ping和Telnet小工具测试网络。

   - 如果ping不通：
     - VPC内访问时，要求客户端与DCS缓存实例的VPC相同，并且正确**配置安全组或白名单**。
     - 公网SSL方式访问Redis 3.0时，要求DCS缓存实例**安全组放开了36379端口访问**。
     - 公网直接访问Redis 3.0（非SSL方式）时，要求DCS缓存实例**安全组放开了6379端口访问**。
   - 如果IP地址可以ping通，telnet对应的端口不通，则尝试重启实例，如重启后仍未恢复，请联系技术支持。

步骤2 检查连接数是否超限。

查看已建立的网络连接数是否超过JedisPool配置的上限。如果连接数接近配置的上限值，则建议重启服务观察。如果明显没有接近，排除连接数超限可能。

Unix/Linux系统使用**:**

netstat -an | grep 6379 | grep ESTABLISHED | wc -l

Windows系统使用:

netstat -an | find "6379" | find "ESTABLISHED" /C

步骤3 检查JedisPool连接池代码。

如果连接数接近配置的上限，请分析是业务并发原因，或是没有正确使用JedisPool所致。

对于JedisPool连接池的操作，每次调用**jedisPool.getResource()**方法之后，需要调用**jedisPool.returnResource()**或者**jedis.close()**进行释放，优先使用close()方法。

步骤4 检查客户端TIME_WAIT是否过多。

通过**ss -s**查看**time wait**链接是否过多。

```
[root@ecs-437a ~]# ss -s
Total: 345
TCP:   122 (estab 39, closed 69, orphaned 0, timewait 0)

Transport Total     IP        IPv6
RAW       2         1         1
UDP       3         2         1
TCP       53        41        12
INET      58        44        14
FRAG      0         0         0
```

如果**TIME_WAIT**过多，可以调整内核参数（/etc/sysctl.conf）：

```
##当出现SYN等待队列溢出时，启用cookies来处理，可防范少量SYN攻击
net.ipv4.tcp_syncookies = 1
##允许将TIME-WAIT sockets重新用于新的TCP连接
net.ipv4.tcp_tw_reuse = 1
##开启TCP连接中TIME-WAIT sockets的快速回收
net.ipv4.tcp_tw_recycle = 1
##修改系统默认的TIMEOUT时间
net.ipv4.tcp_fin_timeout = 30
```

调整后重启生效：**/sbin/sysctl -p**

步骤5 如果按照以上原因排查之后问题仍没有解决，可以通过抓包并将异常时间点、异常信息以及抓包文件发送给技术支持协助分析。

抓包可使用tcpdump工具，命令如下：

tcpdump -i eth0 tcp and port 6379 -n -nn -s 74 -w dump.pcap

Windows系统下还可以安装Wireshark工具抓包。

说明

公网访问Redis 3.0时请将端口改成36379。

网卡名请改成实际的网卡名称。

----结束

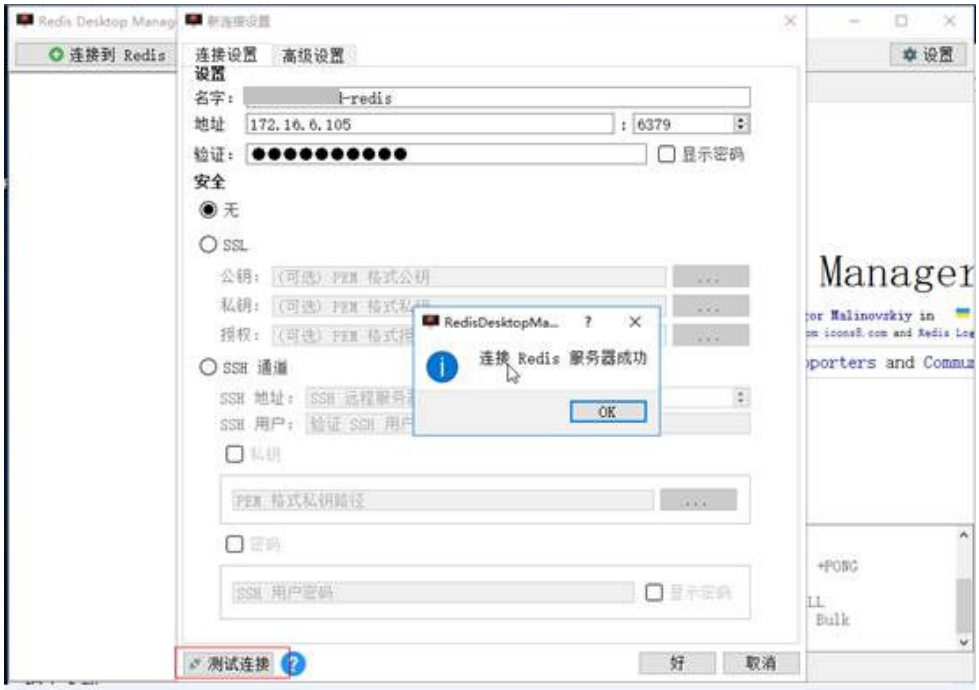
4.9 如何使用 Redis-desktop-manager 访问 Redis 实例?

如下分别介绍通过内网和公网使用Redis-desktop-manager访问Redis 3.0实例的操作。

使用VPC子网访问

1. 填写DCS实例子网地址，端口6379，以及相应密码。
2. 单击左下角“测试连接”。
   提示成功后，说明连接正常。

图 4-1 通过内网使用 Redis-desktop-manager 访问 Redis 实例



说明

使用Redis-desktop-manager访问DCS集群实例时，执行redis命令是正常的，但是左侧显示异常，这个是因为DCS集群是基于codis架构，info命令的输出和原生的redis不一样。

使用公网访问

使用公网访问，需要注意SSL是否开启。

- 如果SSL关闭，则只需要把地址填写实例的公网访问地址即可。
  注意需要放开实例安全组的**入方向6379**端口。
- 如果SSL开启，需要在本地安装Stunnel客户端，才能通过redis-desktop-manager访问Redis。需要注意的是：
  - 必须安装Stunnel客户端，安装配置可参考**Stunnel客户端安装配置**。
  - Redis-desktop-manaer访问地址必须填写**127.0.0.1**，不能填写公网IP，否则出现“connection reset”。

因为SSL开启时，本地访问公网Redis，实际是通过Stunnel搭建的加密隧道。即Redis-desktop-manager访问Stunnel在127.0.0.1监听的socket后，Stunnel通过加密传输到公网Redis的36379端口。

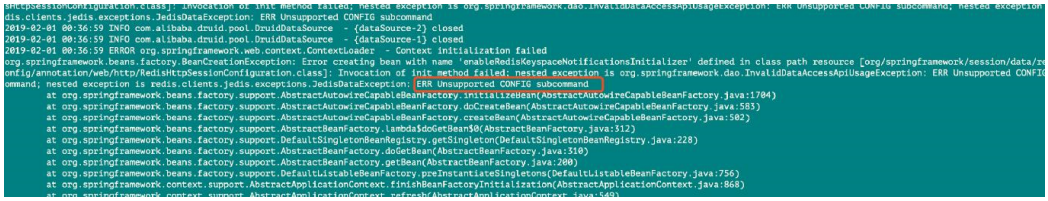
注意需要放开实例安全组的入方向36379端口。

如果SSL已经关闭，需要将其打开，只需要把公网访问关闭，重新打开的时候，启用SSL即可。相反，SSL已经打开，要将其关闭，操作是类似的。

4.10 使用 SpringCloud 时出现 ERR Unsupported CONFIG subcommand 怎么办?

DCS的Redis实例可以配合Spring_Session进行Session共享。DCS的Redis实例对接SpringCloud时，遇到如下错误信息：

图 4-2 Spring Cloud 报错信息



原因为出于安全考虑，DCS暂不支持客户端发起的CONFIG命令，需要按如下步骤进行操作：

1. 通过管理控制台修改Redis实例的配置参数notify-keyspace-event，将值指定为“Egx”。
2. 在Spring框架的XML配置文件中，增加如下：
   ```
   <util:constant
     static-field="org.springframework.session.data.redis.config.ConfigureRedisAction.NO_OP"/>
   ```
3. 修改Spring相关代码，通过启用ConfigureRedisAction.NO_OP这个bean组件，禁止通过客户端调用CONFIG命令，避免报错。
   ```
   @Bean
   public static ConfigureRedisAction configureRedisAction() {
     return ConfigureRedisAction.NO_OP;
   }
   ```

更多说明，可参考**Spring官方文档**。

须知

Redis单机、主备和读写分离实例支持Spring的Session共享，Redis集群实例不支持。

4.11 客户端无法使用域名连接 DCS 缓存实例时如何处理?

DCS缓存实例支持域名访问后，若客户端无法使用域名连接DCS缓存实例，需要将租户子网的DNS服务地址配置为内网域名服务器地址。

具体配置方法请参考**修改VPC子网的DNS服务器地址**。

4.12 本地环境是否可以连接缓存实例?

- 未开启公网访问的DCS缓存实例，本地环境不能直接连接DCS缓存实例。云服务采用虚拟私有云（VPC）管理各服务的网络安全，用户创建的DCS缓存实例，只允许被与实例网络互通的虚拟私有云的弹性云服务器访问。

  用户可以通过网络代理转发的方式，用一台能与DCS缓存实例网络互通的弹性云服务器（ECS）做中转，实现本地环境连接DCS缓存实例。具体操作参考**使用SSH隧道代理机制实现公网访问DCS实例**。
- 已开启公网访问的缓存实例，可以通过本地环境连接DCS缓存实例。

4.13 使用 Redis 实例的发布订阅(pubsub)有哪些注意事项?

Redis实例的发布订阅功能详细信息参见**Pub/Sub**，使用Redis发布订阅功能时有如下事项请注意：

- 客户端需要及时消费和处理消息。

  客户端订阅了channel之后，如果接收消息不及时，可能导致DCS实例消息堆积，当达到消息堆积阈值（默认值为32MB），或者达到某种程度（默认8MB）一段时间（默认为1分钟）后，服务器端会自动断开该客户端连接，避免导致内部内存耗尽。
- 客户端需要支持重连。

  当连接断开之后，客户端需要使用subscribe或者psubscribe重新进行订阅，否则无法继续接收消息。
- 不建议用于消息可靠性要求高的场景中。

  Redis的pubsub不是一种可靠的消息系统。当出现客户端连接退出，或者极端情况下服务端发生主备切换时，未消费的消息会被丢弃。

4.14 Redis 3.0 实例公网开关被关闭是什么原因?

问题现象：已经开启了公网访问的Redis 3.0实例，公网开关被突然关闭（非人为操作）。

可能原因：该Redis实例绑定的弹性公网IP被解绑，导致Redis实例公网被关闭。

4.15 使用短连接访问 Redis 出现“Cannot assign requested address”错误

问题描述

应用程序通过短连接访问Redis实例时，报错：Cannot assign requested address。

问题分析

出现这种错误的应用程序使用的架构基本都是php-fpm加上phpredis，这种架构在并发量较大的情况下，处于TIME-WAIT状态下的TCP连接数较多，客户端无法分配出新的端口，则会出现“Cannot assign requested address”问题。

处理方案

- 方案一：使用pconnect替换connect。

  此方案的思路是用长连接替代短连接，减少TCP连接，同时可以避免每次请求都会重新建立连接的问题，减少延时。

  之前连接Redis的代码如下：

  ```
  $redis->connect('${Hostname}',${Port});
  $redis->auth('${Inst_Password}');
  ```

  现使用pconnect替换connect，即使用persistent connection的方式连接。

  ```
  $redis->pconnect('${Hostname}', ${Port}, 0, NULL, 0, 0, ['auth' => ['${Inst_Password}']]);
  ```

  说明

  - 示例中的连接参数请根据业务实现情况修改，${Hostname}、${Port}和${Inst_Password}为Redis实例的连接地址、端口号和密码。
  - PhpRedis应为5.3.0及以上版本，且建议使用这种pconnect初始化方式，避免断连时出现no auth问题。

- 方案二：修改客户端所在ECS实例的tcp_max_tw_buckets内核参数。

  此方案的思路是直接复用处于TIME-WAIT状态的端口，但是如果ECS和后端服务之间有重传，连接可能会失败，所以建议使用pconnect的方案。

  a. 连接客户端所在ECS实例。

  b. 执行以下命令，查看ip_local_port_range和tcp_max_tw_buckets参数。

     sysctl net.ipv4.tcp_max_tw_buckets net.ipv4.ip_local_port_range

     系统显示类似如下：

     ```
     net.ipv4.tcp_max_tw_buckets = 262144
     net.ipv4.ip_local_port_range = 32768  61000
     ```

  c. 执行以下命令，修改tcp_max_tw_buckets参数，确保tcp_max_tw_buckets的值比ip_local_port_range范围的值小。

     sysctl -w net.ipv4.tcp_max_tw_buckets=10000

一般情况推荐使用方案一，对于一些特定场景（业务代码牵涉过多组件不易变更等场景），需要更快的满足高并发，可以使用方案二

4.16 连接池选择及 Jedis 连接池参数配置建议

Jedis 连接池优势

Lettuce客户端及Jedis客户端比较如下：

- Lettuce：
  - Lettuce客户端没有连接保活探测，错误连接存在连接池中会造成请求超时报错。
  - Lettuce客户端未实现testOnBorrow等连接池检测方法，无法在使用连接之前进行连接校验。

- Jedis：
  - Jedis客户端实现了testOnBorrow、testWhileIdle、testOnReturn等连接池校验配置。

    开启testOnBorrow在每次借用连接前都会进行连接校验，可靠性最高，但是会影响性能（每次Redis请求前会进行探测）。
  - testWhileIdle可以在连接空闲时进行连接检测，合理配置阈值可以及时剔除连接池中的异常连接，防止使用异常连接造成业务报错。
  - 在空闲连接检测之前，连接出现问题，可能会造成使用该连接的业务报错，此处可以通过参数控制检测间隔（timeBetweenEvictionRunsMillis）。

因此，Jedis客户端在面对连接异常，网络抖动等场景下的异常处理和检测能力明显强于Lettuce，可靠性更强。

Jedis 连接池参数配置建议

表 4-1 Jedis 连接池参数配置建议

参数	配置介绍	配置建议
maxTotal	最大连接，单位：个	根据Web容器的Http线程数来进行配置，估算单个Http请求中可能会并行进行的Redis调用次数，例如：Tomcat中的Connector内的maxConnections配置为150，每个Http请求可能会并行执行2个Redis请求，在此之上进行部分预留，则建议配置至少为：150 x 2 + 100= 400 **限制条件：**单个Redis实例的最大连接数。maxTotal和客户端节点数（CCE容器或业务VM数量）数值的乘积要小于单个Redis实例的最大连接数。 例如：Redis主备实例配置maxClients为10000，单个客户端maxTotal配置为500，则最大客户端节点数量为20个。
maxIdle	最大空闲连接，单位：个	配置与maxTotal一致。
minIdle	最小空闲连接，单位：个	一般来说建议配置为maxTotal的X分之一，例如此处常规配置建议为：100。 对于性能敏感的场景，为了防止经常连接数量抖动造成影响，可以配置与maxIdle一致，例如：400。
maxWaitMillis	最大获取连接等待时间，单位：毫秒	获取连接时最大的连接池等待时间，根据单次业务最长容忍的失败时间减去执行命令的超时时间得到建议值。例如：Http最长容忍的失败时间为15s，Redis请求的timeout设置为10s，则此处可以配置为5s。

参数	配置介绍	配置建议
timeout	命令执行超时时间，单位：毫秒	单次执行Redis命令最大可容忍的超时时间，根据业务程序的逻辑进行选择，出于对网络容错等考虑建议配置为不小于210ms。特殊的探测逻辑或者环境异常检测等，可以适当调整达到秒级。
minEvictableIdleTimeMillis	空闲连接逐出时间，大于该值的空闲连接一直未被使用则会被释放，单位：毫秒	如果希望系统不会经常对连接进行断链重建，此处可以配置一个较大值（xx分钟），或者此处配置为-1并且搭配空闲连接检测进行定期检测。
timeBetweenEvictionRunsMillis	空闲连接探测时间间隔，单位：毫秒	根据系统的空闲连接数量进行估算，例如系统的空闲连接探测时间配置为30s，则代表每隔30s会对连接进行探测，如果30s内发生异常的连接，经过探测后会进行连接排除。根据连接数的多少进行配置，如果连接数太大，配置时间太短，会造成请求资源浪费。对于几百级别的连接，常规来说建议配置为30s，可以根据系统需要进行动态调整。
testOnBorrow	向资源池借用连接时是否做连接有效性检测（ping），检测到的无效连接将会被移除。	对于业务连接极端敏感的，并且性能可以接受的情况下，可以配置为True，一般来说建议配置为False，不启用连接空闲检测。
testWhileIdle	是否在空闲资源监测时通过ping命令监测连接有效性，无效连接将被销毁。	True
testOnReturn	向资源池归还连接时是否做连接有效性检测（ping），检测到无效连接将会被移除。	False
maxAttempts	在JedisCluster模式下，您可以配置maxAttempts参数来定义失败时的重试次数。	建议配置3-5之间，默认配置为5。 根据业务接口最大超时时间和单次请求的timeout综合配置，最大配置不建议超过10，否则会造成单次请求处理时间过长，接口请求阻塞。

4.17 如何解决 Lettuce 6.x 版本客户端使用 DCS 实例兼容性问题?

问题现象

使用Lettuce 6.x版本客户端，连接DCS的Redis Proxy（4.x/5.x）集群，会报错"NOAUTH Authentication required"。

图 4-3 报错示例

```
[2022-01-04 18:33:35.219] [lettuce-nioEventLoop-4-1] [DEBUG] [io.lettuce.core.AbstractRedisClient:?] - Connecting to Redis at 192.168.xxx.xxx:6379, initialization
java.util.concurrent.CompletionException: io.lettuce.core.RedisCommandExecutionException: NOAUTH Authentication required.
    at java.util.concurrent.CompletableFuture.encodeThrowable(CompletableFuture.java:292)
    at java.util.concurrent.CompletableFuture.completeThrowable(CompletableFuture.java:308)
```

问题分析

Lettuce 6.x版本开始，使用RESP3（Redis 6.x引入）的HELLO命令进行版本自适应判断，但是对于不支持HELLO命令的低版本实例，兼容性存在一定问题。所以对于低版本的实例，建议直接在Lettuce中指定使用RESP2协议（兼容Redis 4/5）的版本来使用。

解决方案

添加一段代码，指定RESP2协议访问Redis即可解决：

```
package com.chinaroad.parking.config;

import io.lettuce.core.ClientOptions;
import io.lettuce.core.protocol.ProtocolVersion;
import org.springframework.boot.autoconfigure.data.redis.LettuceClientConfigurationBuilderCustomizer;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.redis.connection.lettuce.LettuceClientConfiguration;

@Configuration
public class SpringConfig implements LettuceClientConfigurationBuilderCustomizer {

   @Override
   public void customize(LettuceClientConfiguration.LettuceClientConfigurationBuilder
clientConfigurationBuilder) {
       // manually specifying RESP2
       clientConfigurationBuilder.clientOptions(ClientOptions.builder()
               .protocolVersion(ProtocolVersion.RESP2)
               .build());
   }

}
```

4.18 应该选择域名还是 IP 地址连接 Redis 实例?

- 对于Redis单机、Proxy集群、读写分离实例：

  每个实例只有1个IP地址和1个域名连接地址。实例发生主备交换前后，实例的IP地址和域名连接地址都不会改变。选择域名连接或IP连接不影响功能的使用。

- 对于Redis基础版主备实例：

每个实例有1个IP地址和2个域名连接地址，包含1个只读域名。实例发生主备交换前后，实例的IP地址和域名连接地址都不会改变。选择域名连接或IP连接不影响功能的使用。

使用域名连接时，需要考虑业务的读写请求区别，选择“连接地址”或“IP地址”连接不影响功能的使用，使用“只读地址”连接时只处理读请求（有读写分离需求的用户推荐直接使用读写分离实例）。

- 对于Redis 6.0企业版：

  请使用域名连接实例，IP地址可能有多个或发生变化。

- 对于Cluster集群实例：

  Cluster集群是多主多从架构，有多组主从节点IP地址和1个域名连接地址。选择域名连接或IP连接不影响功能的使用。

  使用IP地址连接实例时，可以使用任意一个IP地址连接实例，连接的节点会将请求发送到正确的节点上，使Cluster的全部节点都可以接收请求。**建议配置多个或全部IP地址连接**，避免所配置的IP地址所在节点故障时导致连接失败。

  域名解析返回的IP数量最多为50个，如需限制解析返回的IP数量，请联系后台管理人员。

说明

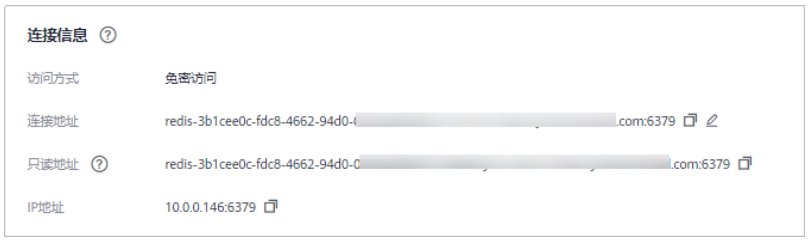
- 如果客户端服务器和Redis实例不在同一Region，需要跨Region访问Redis实例时，实例域名无法跨Region解析，无法通过域名访问。可以通过在hosts中手动配置域名与IP绑定关系或使用IP进行访问。参考**Redis连接约束**。
- 连接实例请参考**连接Redis缓存实例**。

4.19 主备实例的只读地址是连接到主节点还是备节点?

Redis 4.0及以上版本基础版主备实例的连接信息中，有“连接地址”和“只读地址”。其中，连接地址是连接主备实例的主节点，只读地址是连接主备实例的备节点。

详情可以参考**Redis 4.0/5.0/6.0基础版主备实例架构设计**。

图 4-4 连接信息



默认情况下，客户端通过主备实例的主节点读、写数据，备节点用于同步数据。如需使用“只读地址”实现读写分离，需要在客户端增加用户读写请求判断，如果是写请求，将请求发送给读写域名，如果是读请求，将请求发送给只读域名。

5 Redis 使用

5.1 是否支持 CPU 架构的变更?

不支持变更CPU架构。

如需改变CPU架构，可通过“数据迁移+交换IP”方式的方式，创建新的CPU架构的实例，并进行数据迁移，实现CPU架构的变更。具体操作请参考**使用迁移任务在线迁移Redis实例**。

5.2 实例是否支持变更可用区

不支持直接变更可用区。

如需改变可用区，可通过“数据迁移+交换IP”方式的方式，在新的可用区创建实例后，进行数据迁移，实现可用区的变更。具体操作请参考**使用迁移任务在线迁移Redis实例**。

5.3 Redis 实例能否修改 VPC 和子网?

实例的VPC和子网，创建后不允许修改。如果要修改，请重新创建实例，在创建时选择指定的VPC和子网。如果实例已有数据需要迁移，可在创建实例之后进行迁移。具体数据迁移操作，可参考**迁移方案说明**。

5.4 实例是否支持自定义或修改端口?

如果是Redis 3.0、Redis 6.0企业版实例和Memcached实例，访问端口固定为以下端口，不支持指定端口，也不支持修改；如果是Redis 4.0/5.0/6.0基础版实例，支持自定义端口，同时也支持修改端口。

- Redis 3.0

  VPC内使用实例6379端口；公网非SSL方式，访问实例6379端口；公网SSL方式，访问实例36379端口。

- Memcached

  仅支持VPC内访问，端口为11211端口。

- Redis 4.0/5.0/6.0

  创建Redis 4.0/5.0/6.0基础版实例时，可选择自定义端口或使用默认端口，定义端口范围为1~65535，如果没有自定义，则使用默认端口6379。

  创建Redis 6.0企业版实例不支持自定义和修改端口，默认端口为6379。

如果实例与客户端的安全组不同，还需要修改安全组配置，放开端口访问。具体修改方法，请参考**如何配置安全组**。

自定义端口

创建Redis 4.0/5.0/6.0基础版实例时，可在"IP地址"配置项后输入指定的端口号，如不指定，则为默认的端口号6379。

修改端口

Redis 4.0/5.0/6.0基础版实例创建后，如需修改端口号，可按如下步骤操作：

1. 单击DCS控制台左侧菜单栏的“缓存管理”，进入缓存实例管理页面。
2. 单击需要修改端口的实例名称，进入实例基本信息页。
3. 在“连接信息”区域，单击“连接地址”后的  ，可修改端口。

须知

Redis实例的访问端口修改后，Redis实例的所有连接将会中断，业务需要重新连接Redis的新端口。

5.5 实例是否支持修改访问地址?

DCS实例创建后，VPC内访问方式，实例IP连接地址和域名连接地址不支持修改。如果实例开通公网访问，实例绑定的弹性公网IP可以修改。

如果需要更换实例IP地址，需要重新创建实例，在创建实例时，选择“手动分配IP地址”，指定实例的IP地址，然后使用在线迁移方式，将旧的实例数据迁移到新的实例。

5.6 实例无法删除是什么原因?

可能原因如下：

- 实例状态不是“运行中”。

  只有当实例处于“运行中”状态，才能执行删除操作。

- 确认实例是否为创建失败的实例。

  单击缓存实例列表上方“创建失败的实例”后的图标或者数量，如果是创建失败的实例，会显示在弹出的“创建失败的实例”窗口中，请从该界面中进行删除。

5.7 集群实例启动时间过长是什么原因?

可能原因：在集群实例启动过程中，实例节点内部会进行状态、数据的同步。如果在完成同步之前就持续写入较多的数据，会导致实例内部同步耗费较长时间，实例状态一直处于“启动中”。直到同步完成，集群实例状态才会切换到“运行中”。

解决方案：建议等集群实例启动完成后，再恢复业务数据写入。

5.8 使用 redis_exporter 出错怎么办?

通过在命令行启动redis_exporter，根据界面输出，查看是否存在错误，根据错误描述，进行问题排查。

```
[root@ecs-swk /]./redis_exporter -redis.addr 192.168.0.23:6379
INFO[0000] Redis Metrics Exporter V0.15.0    build date:2018-01-19-04:08:01 sha1:
a0d9ec4704b4d35cd08544d395038f417716a03a
  Go:go1.9.2
INFO[0000] Providing metrics at :9121/metrics
INFO[0000] Connecting to redis hosts: []string{192.168.0.23:6379}
INFO[0000] Using alias:[]string{""}
```

5.9 什么是预留内存，如何配置预留内存?

预留内存介绍

预留内存是一部分不能用来存储数据的内存空间，主要用于数据持久化，主备同步，执行实例备份等操作。

配置参数名称：reserved-memory-percent

须知

监控中内存利用率统计是扣除预留内存的。

仅如下实例类型需要预留内存（其他实例类型不涉及）：

- Redis 3.0 单机
- Redis 3.0 主备
- Memcached 单机
- Memcached 主备

未配置足够的预留内存（数据部分占用的内存容量过高），可能会导致：

- 缓存实例操作速度变慢（系统启用swap，性能下降）。
- 无法备份数据。
- 数据无法及时主备同步。
- 实例规格变更失败。
- 可能会造成进程重启。

配置预留内存操作步骤

请参考**修改实例配置参数**修改“reserved-memory-percent”参数的值。

说明

- 建议您的实例预留内存值至少配置为30%，2021年之后创建的实例预留内存默认值已经修改为30%。
- 预留内存百分比是以实例规格的最大可用内存为基数，而不是以内存规格为基数的，最大可用内存可参考**实例规格**中实例产品规格表中“实例可使用内存”列的值。

5.10 创建的缓存实例为什么可使用内存比实例规格少一些?

Redis 3.0、Memcached采用虚拟机部署，系统会占用小部分内存。其他版本的实例不会存在该问题。

5.11 Redis 3.0 Proxy 集群不支持 redisson 分布式锁的原因

redisson分布式锁的加锁和解锁流程如下：

1. redisson分布式锁的加锁和解锁都是执行一段lua脚本功能实现的。
2. 在加锁阶段，需要在lua脚本中执行exists、hset、pexpire、hexists、hincrby、pexpire、pttl命令。
3. 在解锁阶段，需要在lua脚本中执行exists、publish、hexists、pexpire、del命令。

由于Proxy集群支持publish/subscribe(redis的发布订阅)时，是需要在Proxy节点上识别publish/subscribe命令，做一些特殊处理（转发给所有redis-server的节点），因此不支持直接在lua脚本中执行publish命令。

因此，Redis 3.0 Proxy集群无法支持redisson的分布式锁机制，**如果需要使用redisson分布式锁功能，建议使用Redis 4.0或Redis 5.0集群**。

5.12 DCS Redis 有没有后台管理软件?

没有。Redis的配置信息与使用信息可通过Redis-cli查询；对Redis实例的监控数据可通过云监控服务查看，监控数据的查看方法如下。

查看 DCS 性能监控

步骤1 登录**分布式缓存服务管理控制台**。

步骤2 在管理控制台左上角单击 ，选择实例所在的区域。

步骤3 单击左侧菜单栏的“缓存管理”，进入缓存实例信息页面。

步骤4 单击需要查看性能监控指标的缓存实例，进入实例基本信息页面。

步骤5 单击“性能监控”，页面显示该实例的所有监控指标信息。

说明

您也可以在需要查看的缓存实例的“操作”列，单击“查看监控”，进入云监控服务的页面查看，这和在缓存实例信息页面“性能监控”页签内容一致。

----结束

5.13 DCS 缓存实例的数据被删除后，能否找回?

如果DCS缓存实例执行了备份操作，实例的数据被删除后，可通过备份文件对数据进行恢复，但是恢复会覆盖备份时间到恢复这段时间的写入数据。

主备、集群和读写分离实例通过控制台的“备份与恢复”功能将已备份的数据恢复到DCS缓存实例中，参考**实例恢复**。

另外，如果DCS缓存实例被删除，实例中原有的数据将被删除，实例的备份数据也会删除，请谨慎操作。在删除实例之前，您可以将实例的备份文件下载，本地永久保存，如需恢复数据，可将本地备份文件迁移到新的实例中。下载备份数据的方式，请参考**如何导出Redis实例数据?** 。

5.14 为什么实例实际可用内存比申请规格小而且已使用内存不为 0?

由于系统开销会占用部分资源，主备实例的持久化也需要一部分资源，所以Redis 3.0和Memcached实例创建后，缓存实例实际可用内存小于申请规格。除了用户存储数据外，Redis-server内部的buffer以及内部数据结构会占用一部分内存。所以缓存实例创建后，实例已使用内存量不为0。其他版本的实例不涉及该问题。

5.15 如何查看 Redis 内存占用量

当前DCS Redis提供了以下与内存相关的指标。查看监控指标的方式请参考**查看性能监控**。

表 5-1 Redis 实例支持的监控指标

指标ID	指标名称	含义	取值范围	测量对象&维度	监控周期（原始指标）
memory_usage	内存利用率	该指标用于统计测量对象的内存利用率。 单位：%。	0-100 %	测量对象：Redis实例 测量维度：dcs_instance_id	1分钟

指标ID	指标名称	含义	取值范围	测量对象&维度	监控周期（原始指标）
used_memory	已用内存	该指标用于统计Redis已使用的内存字节数。 单位：可在控制台进行选择，如KB、MB、byte等。	>=0byte	测量对象： Redis实例 测量维度： dcs_instance_id	1分钟
used_memory_dataset	数据集使用内存	该指标用于统计Redis中数据集使用的内存。 单位：可在控制台进行选择，如KB、MB、byte等。	>=0byte	测量对象： Redis实例 仅Redis 4.0及以上的版本支持 测量维度： dcs_instance_id	1分钟
used_memory_dataset_perc	数据集使用内存百分比	该指标用于统计Redis中数据内存所占当前已用总内存的百分比。 单位：%。	0-100%	测量对象： Redis实例 仅Redis 4.0及以上的版本支持 测量维度： dcs_instance_id	1分钟
used_memory_rss	已用内存RSS	该指标用于统计Redis已使用的RSS内存。即实际驻留“在内存中”的内存数。包含堆内存，但不包括换出的内存。 单位：可在控制台进行选择，如KB、MB、byte等。	>=0byte	测量对象： Redis实例 测量维度： dcs_instance_id	1分钟
memory_frag_ratio	内存碎片率	该指标用于统计当前的内存碎片率。其数值上等于used_memory_rss / used_memory。	>=0	测量对象： Redis实例 测量维度： dcs_instance_id	1分钟

指标ID	指标名称	含义	取值范围	测量对象&维度	监控周期（原始指标）
used_memory_peak	已用内存峰值	该指标用于统计Redis服务器启动以来使用内存的峰值。 单位：可在控制台进行选择，如KB、MB、byte等。	>=0byte	测量对象： Redis实例 测量维度： dcs_instance_id	1分钟
used_memory_lua	Lua已用内存	该指标用于统计Lua引擎已使用的内存字节。 单位：可在控制台进行选择，如KB、MB、byte等。	>=0byte	测量对象： Redis实例 测量维度： dcs_instance_id	1分钟

5.16 Cluster 集群实例容量和性能未达到瓶颈，但某个分片容量或性能已过载是什么原因?

这是由于Cluster集群采用的是分片设计理念，每个**具体的Key只能分布到某一个具体的分片节点上**，计算Key的分布过程有以下两个步骤：

1. 针对Key值进行CRC16算法计算后对16384取模，得到对应的槽位（Slot）值。
2. 根据S槽位（Slot）和分片的映射关系，找到Key具体应该属于的分片，并且进行存取。

所以，Key并没有均匀分布在实例的各个分片上，是根据计算结果进行存取的。在大Key和热Key存在时，就会出现某个分片容量或性能已过载，但其他分片内存负载还是很低，并没有达到容量和性能的瓶颈。

5.17 访问 Redis 报 OOM 错误提示

问题描述

访问Redis返回Error in execution; nested exception is io.lettuce.core.RedisCommandExecutionException: OOM command not allowed when used memory > 'maxmemory'。

问题排查

OOM代表的就是超过了最大内存，报错中OOM command not allowed when used memory > 'maxmemory'的‘maxmemory’这个参数是Redis服务端对最大内存的配置，可以看到这是内存使用满了。

若Redis实例内存使用率并未达到100%，有可能当前写入数据的那个节点的mem达到最大值。通过**redis-cli -h <redis_ip> -p 6379 -a <redis_password> -c --bigkeys**连接到集群的各个节点进行分析。如果连接的从节点，需要在执行**bigkeys**命令之前，先发送**READONLY**命令。

5.18 不同编程语言如何使用 Cluster 集群客户端

当前DCS Cluster集群对比Proxy集群的优势和特性：

表 5-2 Cluster 集群与 Proxy 集群差异

对比项	Cluster集群	Proxy集群
原生兼容性	高	中
客户端兼容性	中（需要客户端开启集群模式）	高
性价比	高	中
时延	低时延	中等时延
读写分离	原生支持（客户端SDK配置）	Proxy实现
性能	高	中

Cluster集群由于没有代理层，在时延和性能方面具备一定的优势；但是对于客户端使用方面，由于Cluster集群使用开源的Redis Cluster协议，在客户端的兼容性方面略差于Proxy集群。

推荐的Cluster集群客户端：

表 5-3 Cluster 集群客户端

客户端语言	客户端类型	Cluster集群参考文档
Java	Jedis	https://github.com/xetorthio/jedis#jedis-cluster
Java	Lettuce	https://github.com/lettuce-io/lettuce-core/wiki/Redis-Cluster

客户端语言	客户端类型	Cluster集群参考文档
PHP	php redis	https://github.com/phpredis/phpredis#readme
Go	Go Redis	Cluster集群：https://pkg.go.dev/github.com/go-redis/redis/v8#NewClusterClient Proxy集群或单机主备：https://pkg.go.dev/github.com/go-redis/redis/v8#NewClient
Python	redis-py-cluster	https://github.com/Grokzen/redis-py-cluster#usage-example
C	hiredis-vip	https://github.com/vipshop/hiredis-vip?_ga=2.64990636.268662337.1603553558-977760105.1588733325
C++	redis-plus-plus	https://github.com/sewenew/redis-plus-plus?_ga=2.64990636.268662337.1603553558-977760105.1588733325#redis-cluster
Node.js	node-redis io-redis	https://github.com/NodeRedis/node-redis https://github.com/luin/ioredis

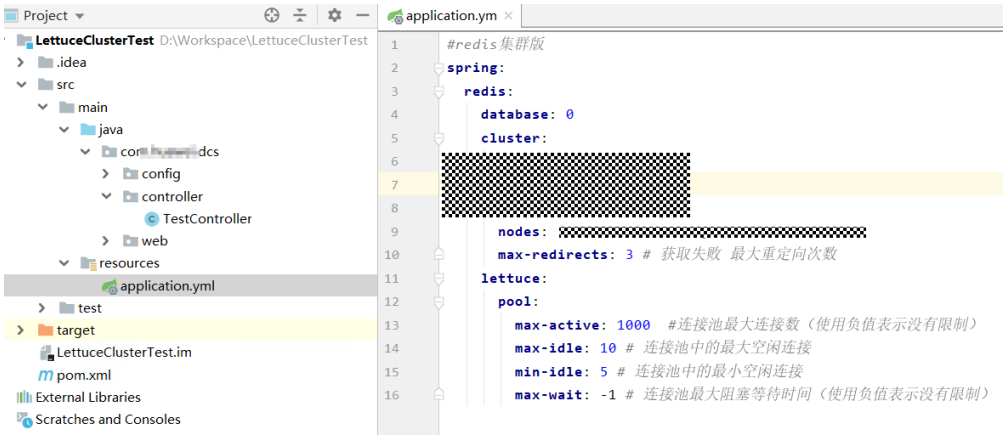
官方推荐的开源客户端列表：**https://redis.io/clients**。

5.19 使用 Cluster 的 Redis 集群时建议配置合理的超时时间

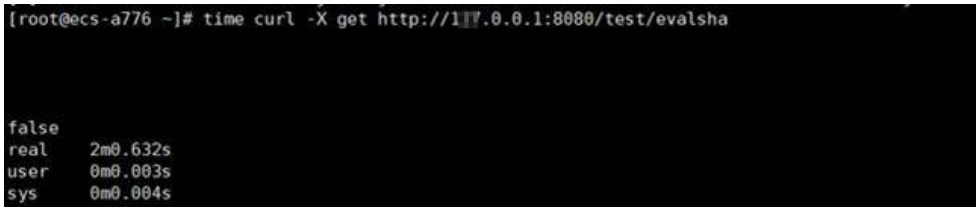
客户端配置问题导致无法连接。

当集群实例备节点故障情况下，客户端使用SpringBoot + Lettuce的方式连接Redis，使用的Lettuce客户端在连接集群时，需要与所有节点先建立连接（包括故障节点）。

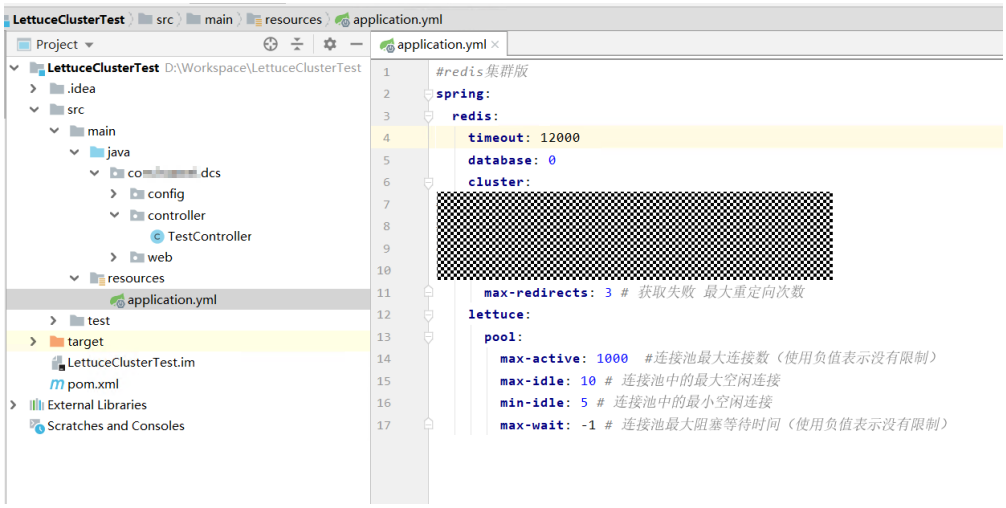
- 在未配置timeout超时的情况下，模拟备节点故障时，可能出现分钟级的超时阻塞（Lettuce客户端的老版本默认超时为120s，新版本默认为60s），配置如下图：



可能会导致端到端业务访问时间过长（最长达到默认超时时间），如下图所示：



- 在客户端侧添加timeout参数后，备节点超时时间大幅度缩短，并且可以根据客户自己的业务诉求进行调整，配置如下：



配置后查看端到端业务访问时间如下图所示：

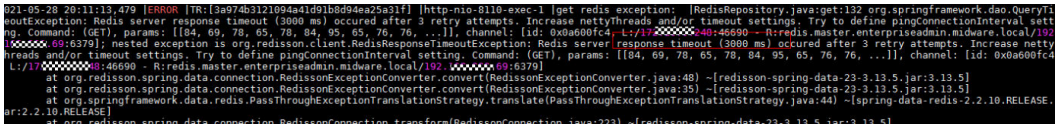
因此在未配置timeout参数情况下，客户端在建立连接时，故障节点由于未配置timeout超时，在建立连接时会出现连接阻塞的情况。

建议：用户需根据业务能容忍的超时时间进行设置，例如在一次HTTP端到端请求中，需要请求两次Redis，而HTTP请求的最大超时时间为10s，则建议将超时时间配置为5s，防止由于超时时间过长或者未配置超时时间造成故障场景下的业务受损。

5.20 读取 redis 数据报超时错误

问题描述

读取redis数据报超时错误：redis server response timeout（3000ms） occurred after 3 retry attempts。



问题排查

1. 根据报错后的提示，将客户端超时时间改大一些。
2. 确认问题发生时的操作，若为执行操作时Key值过大，也会造成超时报错。一般建议key不要超过10KB。

   单key大小，Redis限制每个String类型value大小不超过512MB， 实际开发中，不要超过10KB，否则会对CPU和网卡造成极大负载。

   hash、list、set、zset元素个数不要超过5000。

   理论上限: 每个hashset里元素数量< 2^32。
3. 根据报错后的提示，将客户端参数PingConnectionInterval调大试试。

5.21 hashtag 的原理、规则及用法示例

hashtag 原理

单实例上的mset、lua脚本等处理多key时，是一个原子性(atomic)操作，所有给定key都会在同一时间内被执行。集群每次通过对key进行hash计算到不同的分片，所以集群上同时执行多个key，不再是原子性操作，会存在某些给定 key 被更新而另外一些给定key没有改变的情况，其原因是需要设置的多个key可能分配到不同的机器上。因此集群引入了hashtag来对多key同时操作，在设置了hashtag的情况下，集群会根据hashtag决定key分配到的slot， 当两个key拥有相同的hashtag时, 它们会被分配到同一个slot。

hashtag 使用规则

第一次出现“{”和接下来第一次出现的“}”之间有内容。

例如：

- 这两个键{user1000}.following和{user1000}.followers由于只有一对{}，将user1000来计算hash。
- 对于键foo{}{bar}，整个键foo{}{bar}将像往常一样计算hash，因为第一次出现的“{”后面跟“}”中间没有字符。
- 对于键foo{{bar}}zap，子字符串{bar将被计算hash，因为它是第一次出现“{”和第一次出现“}”之间的子字符串。
- 对于键foo{bar}{zap}的子字符串bar将被计算hash，因为只使用第一个“{”和“}”。

hashtag 用法示例

当如下操作时：

```
EVAL "redis.call('set',KEYS[1],ARGV[1]) redis.call('set',KEYS[2],ARGV[2])" 2 key1 key2 value1 value2
```

出现以下报错：

ERR 'key1' and 'key2' not in the same slot

可通过hashtag进行解决：

```
EVAL "redis.call('set',KEYS[1],ARGV[1]) redis.call('set',KEYS[2],ARGV[2])" 2 {user}key1 {user}key2 value1 value2
```

5.22 Redis key 丢失是什么原因

redis实例是不会主动丢失数据的，key丢失一般有这几种情况：1、key过期；2、key被逐出；3、key被删除。

按照顺序进行排查：

1. 查看key是否过期。
2. 查看监控，分析是否会触发键逐出机制。
3. 去服务端分析info查看是否有删除key的操作。

5.23 重启实例后缓存数据会保留吗?

单机缓存实例重启后，原有的数据将被删除。

主备、读写分离和集群实例（单副本集群除外）默认支持AOF持久化，实例重启后原有的数据会保留。如果关闭了AOF持久化（appendonly参数修改为no即AOF持久化功能关闭），实例重启后原有的数据将被删除。

5.24 如何确认实例是单 DB 还是多 DB

单机、主备、读写分离实例类型都为多DB（256个，DB编号为0-255）。

Proxy集群实例默认只有一个DB，支持手动开启多DB，如需开启多DB的操作请参考**Proxy集群使用多DB限制**。

Redis Cluster集群实例不支持多DB，只有一个DB。

Redis 4.0及以上版本的实例，通过控制台连接redis实例后，即可以查看是否为多DB，如下图所示。

图 5-1 连接 Redis

图 5-2 查看 Database

5.25 Proxy 集群开启多 DB 的使用限制及操作方式

DCS对于实现多DB存在一定的约束，建议针对客户业务进行评估：

- **使用约束：**
  a. swapdb不支持多DB。
  b. info keyspace不支持多DB展示。
  c. 需要查询每个DB的key总数，可以使用自定义dbstats命令。命令执行数据节点上会有CPU冲高。
  d. LUA脚本中不支持多DB。
  e. RANDOMKEY命令不支持。
  f. 不支持在lua脚本中使用publish。
  g. DB数支持范围为0 ~ 255。
- **性能约束：**
  a. flushdb命令采用逐个key删除的方式执行，耗时久，慢于开源原生实现，速度与SCAN命令相同（需要客户实际测试）。
  b. dbsize命令耗时长，禁止在代码中使用。
  c. 多DB场景下keys命令和scan命令性能会有损失（最多50%）。
- **其他约束：**
  后端存储会按照一定规则对key进行改写，导出RDB数据中的key不是原始的key，但通过Redis协议访问无影响。

开启/关闭多 DB 操作步骤

Proxy集群实例默认不开启多DB，支持按照以下操作进行多DB开启。

步骤1 登录分布式缓存服务控制台。

步骤2 连接实例，执行**flushall**命令清空原有数据。

说明

开启或关闭多DB操作时，需要确保实例数据已清空且无新数据写入，否则会操作失败。

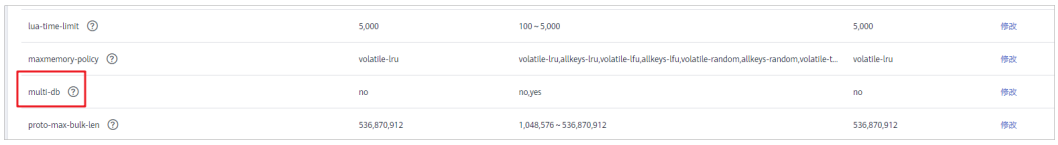
步骤3 在缓存管理页面，单击缓存实例进入实例详情页面。

步骤4 单击“实例配置 > 参数配置”进入参数配置页面。

步骤5 单击multi-db参数后的“修改”，将参数运行值修改为“yes”，即开启多DB。

如需关闭多DB，将参数运行值修改为“no”。

步骤6 单击“保存”，在修改参数配置弹框中单击“是”，完成开启/关闭多DB操作，无需重启实例。



----结束

5.26 如何创建多 DB 的 Proxy 集群实例?

创建Proxy集群实例时，默认DB数为1，如需创建多DB的Proxy集群实例，请参考如下步骤：

说明

创建多DB Proxy集群实例前，建议了解**Proxy集群使用多DB限制**。

步骤1 登录**分布式缓存服务管理控制台**。

步骤2 在管理控制台左上角单击 ，选择区域。

步骤3 单击“参数模板”进入“系统默认模板”页面。

步骤4 选择要创建的缓存版本和类型（Proxy集群），单击对应的“创建为自定义模板”。

步骤5 将“参数配置”下的“multi-db”设置为yes。

步骤6 输入新的模板名称后单击“确定”，创建自定义模板成功。

步骤7 单击“缓存管理>购买缓存实例”，创建Proxy集群实例。

创建实例时，需将“参数配置”选择为“使用自定义模板”，并选择如上步骤中创建的自定义模板，即可创建多DB的Proxy集群实例。

创建成功后，可连接Redis查看是否为多DB实例。

----结束

6 扩容缩容与实例升级

6.1 Redis 实例是否支持版本升级，如 Redis 4.0 升级到 Redis 5.0？

暂不支持直接升级。Redis不同版本的底层架构不一样，在创建Redis实例时，确定Redis版本后，不能修改，如Redis 4.0的实例不能升级到Redis 5.0。

如您的业务需要使用Redis高版本的功能特性，可重新创建高版本Redis实例，然后将原有Redis实例的数据迁移到高版本实例上。具体数据迁移操作，可参考**迁移方案说明**。

6.2 升级 Redis 3.0 实例到高版本实例

方案概述

Redis开源社区自2019年5月19日发布Redis 3.0最后一个小版本后，一直未对Redis 3.0进行更新。华为云DCS也于2021年3月发布了停售DCS Redis 3.0的公告。

鉴于Redis 3.0版本较老，开源社区已不再对其进行更新，DCS提供的Redis 4.0/5.0/6.0高版本兼容Redis 3.0，建议客户尽快将DCS Redis 3.0升级到高版本。

DCS暂不支持直接升级实例版本，只能通过数据迁移将低版本实例中的数据迁移到高版本，从而实现Redis版本升级。本章节介绍如何通过数据迁移+交换实例IP的方式升级Redis 3.0实例到高版本。

约束与限制

- 通过数据迁移的方式升级Redis版本，对客户业务可能有以下影响：
  - 数据同步完成后，需要交换源Redis与目标Redis实例的IP地址，交换IP地址时会有一分钟内只读和30秒左右的中断。
  - 如果升级后实例与原实例密码不一致，数据同步完成后，需要切换访问Redis的密码，切换时需要停止业务。因此，建议升级前后实例密码保持一致。
- 建议在业务低峰期进行实例升级操作。

前提条件

- 创建与Redis 3.0相同VPC和子网，相同实例类型、相同访问密码、且规格不小于原实例规格的高版本Redis实例。例如，用户需要将Redis 3.0 16GB主备实例升级到Redis 5.0版本，则需要提前创建一个不小于16GB的Redis 5.0主备实例。

  创建Redis实例的操作，请参考**创建DCS Redis缓存实例**。
- 手动备份Redis 3.0源实例数据。备份数据的操作，请参考**如何导出Redis实例数据？**。

迁移实例数据

步骤1 登录**分布式缓存服务管理控制台**。

步骤2 在管理控制台左上角单击 ，选择源Redis所在的区域。

步骤3 单击左侧菜单栏的“数据迁移”。页面显示迁移任务列表页面。

步骤4 单击右上角的“创建在线迁移任务”。

步骤5 设置迁移任务名称和描述。

步骤6 配置在线迁移任务虚拟机资源的VPC、子网和安全组。

- 迁移任务需要与源Redis和目标Redis实例网络互通，请选择与Redis实例相同的VPC。
- 迁移任务创建后，会占用一个租户侧IP，即控制台上迁移任务对应的“迁移机IP”，如果目标Redis配置了IP白名单，需要放通迁移机IP。
- 迁移任务所选安全组的“出方向规则”需放通源端Redis和目标端Redis的IP和端口（安全组默认情况下为全部放通，则无需单独放通），以便迁移任务的虚拟机资源能访问源Redis和目标Redis。

步骤7 在线迁移任务创建完成后，单击在线迁移任务右侧“操作”列的“配置”，配置在线迁移的源Redis、目标Redis等信息。

步骤8 迁移方法请选择“全量迁移+增量迁移”，仅当选择“全量迁移+增量迁移”的迁移方法时，支持通过控制台交换源端与目标端实例的IP地址。如果选择“全量迁移”，需要手动切换业务连接Redis的IP地址。

表 6-1 在线迁移方法说明

迁移类型	描述
全量迁移	该模式为Redis的一次性迁移，适用于可中断业务的迁移场景。全量迁移过程中，**如果源Redis有数据更新，这部分更新数据不会被迁移到目标Redis**。

迁移类型	描述
全量迁移 + 增量迁移	该模式为Redis的持续性迁移，适用于对业务中断敏感的迁移场景。增量迁移阶段通过解析日志等技术， 持续保持源Redis和目标端Redis的数据一致。 增量迁移，**迁移任务会在迁移开始后，一直保持迁移中状态，不会自动停止**。需要您在合适时间，在“操作”列单击“停止”，手动停止迁移。停止后，源端数据不会丢失，只是目标端不再写入数据。增量迁移在传输链路网络稳定情况下是秒级时延，具体的时延情况依赖于网络链路的传输质量。

步骤9 当迁移方法选择“全量迁移+增量迁移”时，支持选择是否启用“带宽限制”。

启用带宽限制功能，当数据同步速度达到带宽限制时，将限制同步速度的继续增长。

步骤10 选择是否“自动重连”。如开启自动重连模式，迁移过程中在遇到网络等异常情况时，会无限自动重连。

自动重连模式在无法进行增量同步时，会触发全量同步，增加带宽占用，请谨慎选择。

步骤11 “源Redis实例”和“目标Redis实例”，请分别选择需要升级的Redis 3.0实例和新建的高版本Redis实例。

步骤12 如果源Redis和目标Redis为密码访问模式，请分别在“源Redis实例密码”和“目标Redis实例密码”处输入实例密码后，单击密码右侧的“测试连接”，检查实例密码是否正确、网络是否连通。如果源Redis和目标Redis为免密访问模式，无需输入密码，直接单击“测试连接”。

步骤13 在“源DB”和“目标DB”中，可以选择是否需要指定具体迁移的DB。例如源端输入5，目标端输入6时，表示迁移源Redis DB5中的数据到目标Redis的DB6。当源端不指定DB，目标端指定DB时，表示默认迁移源端的全部数据到目标端指定的DB；当目标端不指定DB时，表示默认迁移到与源端对应的DB。本次操作“源DB”和“目标DB”置空即可。

步骤14 单击“下一步”。

步骤15 确认迁移信息，然后单击“提交”，开始创建迁移任务。

可返回迁移任务列表中，观察对应的迁移任务的状态，迁移成功后，任务状态显示“成功”。

说明

- 如果是增量迁移，会一直保持迁移中的状态。
- 如需手动停止迁移，请选中需要停止的迁移任务，单击“停止”。
- 数据迁移后，目标端与源端重复的Key会被覆盖。

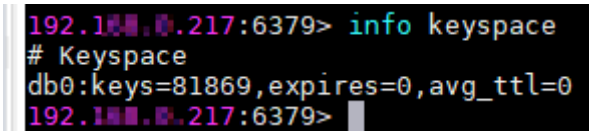
如果出现迁移失败，可以单击迁移任务名称，进入迁移任务详情页面，查看“迁移日志”。

----结束

迁移后验证

数据迁移前如果目标Redis中数据为空，迁移完成后，可以通过以下方式确认数据的完整性：

1. 连接源Redis和目标Redis。连接Redis的方法请参考**Redis-cli客户端连接Redis**。
2. 输入info keyspace，查看keys参数和expires参数的值。



3. 对比源Redis和目标Redis的keys参数分别减去expires参数的差值。如果差值一致，则表示数据完整，迁移正常。

注意：如果是全量迁移，迁移过程中源Redis更新的数据不会迁移到目标实例。

交换 DCS 实例 IP

当DCS源Redis与目标Redis满足以下条件时，支持交换源Redis与目标Redis的IP地址。交换实例IP后，客户端代码无需修改源端实例的访问地址，即可自动连接到目标Redis。

- 源端和目标端必须是基础版Redis，且不能是Cluster集群实例。企业版实例和Cluster集群实例不支持实例交换IP。
- 源端是Redis 3.0实例，需要先联系后台管理人员开通Redis 3.0实例交换IP的白名单，并且仅当Redis 3.0作为源端实例，目标端为Redis 4.0/5.0/6.0基础版实例时，支持交换实例IP。
- 开启公网访问后的源实例或目标实例，不支持控制台交换IP，只能手动更换业务连接Redis的IP地址。
- 实例**选择迁移方法**步骤时，必须选择的是“**全量迁移＋增量迁移**”，如果是“全量迁移”的方式，则不支持交换实例IP。
- 源Redis与目标Redis实例的端口需要一致。

须知

1. 交换IP过程中，会自动停止在线迁移任务。
2. 源端实例为Redis 3.0，交换IP地址时，会有一分钟内只读和30秒左右的中断。
3. 请确保您的客户端应用具备重连机制和处理异常的能力，否则在交换IP后有可能需要重启客户端应用。
4. 如果源端是主备实例，交换IP时不会交换备节点IP，请确保应用中没有直接引用备节点IP。
5. 如果应用中有直接引用域名，请选择交换域名，否则域名会挂在源实例中。
6. 请确保目标Redis和源Redis密码一致，否则交换IP后，客户端会出现密码验证错误。
7. Redis3.0实例交换IP地址后，需要将源实例的安全组配置同步至目标实例的白名单配置中。

步骤1 在“数据迁移 > 在线迁移”页面，当迁移任务状态显示为“增量迁移中”时，单击操作列的“更多 > 交换IP”打开交换IP弹框。

步骤2 在交换IP弹框中的交换域名区域，选择是否交换域名。

说明

- 如果客户端使用域名连接Redis，必须选择交换域名，否则客户端应用需要修改使用的域名。
- 如果没有选择交换域名，则只交换实例的IP地址。

步骤3 单击“确定”，交换IP任务提交成功，当迁移任务的状态显示为“IP交换成功”，表示交换IP任务完成。

----结束

业务功能验证

- 验证业务功能是否正常。例如，检查客户端访问Redis是否有报错。
- 观察关键性能监控指标是否异常。例如，活跃客户端连接数、每秒并发操作数、CPU使用率、内存使用率等监控参数。

6.3 在维护时间窗内对实例维护是否有业务中断?

在实例维护时间窗内，服务运维要对实例进行维护操作时，会提前和用户沟通确认；具体升级操作以及影响，服务运维人员会提前和用户确认，用户不用担心维护窗内，实例运行异常的问题。

6.4 DCS 实例规格变更是否需要关闭或重启实例?

实例处于运行中的状态即可进行规格变更，不涉及关闭实例资源或重启实例的操作。

6.5 DCS 支持哪些实例类型变更?

表 6-2 DCS 实例类型变更明细

实例版本	支持的实例变更类型	变更须知及影响
Redis 3.0	单机实例变更为主备实例	连接会有秒级中断，大约1分钟左右的只读。
	主备实例变更为Proxy集群实例	1. 如果Redis 3.0主备实例数据存储在多DB上，或数据存储在非DB0上，不支持变更为Proxy集群；数据必须是只存储在DB0上的主备实例才支持变更为Proxy集群。 2. 连接会中断，5~30分钟只读。
Memcached	单机实例变更为主备实例	会有秒级业务中断、大约1分钟只读。

实例版本	支持的实例变更类型	变更须知及影响
Redis 4.0/5.0/6.0	主备实例或读写分离实例变更为Proxy集群实例 Proxy集群实例变更为主备实例或读写分离实例	1. 变更为proxy集群时，需要评估proxy集群的多DB使用限制和命令使用限制对业务的影响。具体请参考**proxy集群使用多DB限制**，**实例受限使用命令**。 2. 变更前实例的已用内存必须小于变更后最大内存的70%，否则将不允许变更。 3. 如果变更前实例的已用内存超过总内存的90%，变更的过程中可能会导致部分key逐出。 4. 变更完成后需要对实例重新**创建告警规则**。 5. 如果原实例是主备实例，请确保应用中没有直接引用只读IP或只读域名。 6. 请确保您的客户端应用具备重连机制和处理异常的能力，否则在变更规格后有可能需要重启客户端应用。 7. 变更规格过程中会有秒级业务中断、大约1分钟只读，建议在业务低峰时进行变更。
Redis 4.0/5.0/6.0	主备实例变更为读写分离实例 **说明** 读写分离实例暂不支持直接变更为主备实例。	1. 目前只支持主备实例变更为相同容量的读写分离实例，小于4G规格的主备实例不支持变更为读写分离实例。 2. 如果变更前实例的已用内存超过总内存的90%，变更的过程中可能会导致部分key逐出。 3. 变更完成后需要对实例重新**创建告警规则**。 4. 请确保主备实例的应用中没有直接引用只读IP或只读域名。 5. 请确保您的客户端应用具备重连机制和处理异常的能力，否则在变更规格后有可能需要重启客户端应用。 6. 变更规格过程中会有秒级业务中断，建议在业务低峰时进行变更。 7. 主备实例如果创建了ACL账号，不支持变更为读写分离实例。 8. Redis 6.0如果开启了SSL链路加密传输，不支持变更为读写分离实例。

实例类型变更后支持的命令，请参考对应的**开源命令兼容性**。

除了上表中提到的实例外，其他实例类型目前不支持实例类型的变更，若您想实现跨实例类型的规格变更，可参考**使用迁移任务在线迁移Redis实例**进行操作。

实例类型是否支持变更，以控制台实例的“变更规格”操作界面为准。

6.6 DCS 实例规格变更的业务影响

执行实例规格变更操作，建议在业务低峰期进行。

业务高峰期（如实例在内存利用率、CPU利用率达到90%以上或写入流量过大）变更规格可能会失败，若变更失败，请在业务低峰期再次尝试变更。

在实例规格变更时，可能会存在的影响如下:

实例类型变更须知

表 6-3 DCS 实例类型变更明细

实例版本	支持的实例变更类型	变更须知及影响
Redis 3.0	单机实例变更为主备实例	连接会有秒级中断，大约1分钟左右的只读。
	主备实例变更为Proxy集群实例	1. 如果Redis 3.0主备实例数据存储在多DB上，或数据存储在非DB0上，不支持变更为Proxy集群；数据必须是只存储在DB0上的主备实例才支持变更为Proxy集群。 2. 连接会中断，5~30分钟只读。
Memcached	单机实例变更为主备实例	会有秒级业务中断、大约1分钟只读。
Redis 4.0/5.0/6.0	主备实例或读写分离实例变更为Proxy集群实例	1. 变更为proxy集群时，需要评估proxy集群的多DB使用限制和命令使用限制对业务的影响。具体请参考**proxy集群使用多DB限制**，**实例受限使用命令**。 2. 变更前实例的已用内存必须小于变更后最大内存的70%，否则将不允许变更。 3. 如果变更前实例的已用内存超过总内存的90%，变更的过程中可能会导致部分key逐出。 4. 变更完成后需要对实例重新**创建告警规则**。 5. 如果原实例是主备实例，请确保应用中没有直接引用只读IP或只读域名。 6. 请确保您的客户端应用具备重连机制和处理异常的能力，否则在变更规格后有可能需要重启客户端应用。 7. 变更规格过程中会有秒级业务中断、大约1分钟只读，建议在业务低峰时进行变更。
	Proxy集群实例变更为主备实例或读写分离实例	

实例版本	支持的实例变更类型	变更须知及影响
Redis 4.0/5.0/6.0	主备实例变更为读写分离实例 **说明** 读写分离实例暂不支持直接变更为主备实例。	1. 目前只支持主备实例变更为相同容量的读写分离实例，小于4G规格的主备实例不支持变更为读写分离实例。 2. 如果变更前实例的已用内存超过总内存的90%，变更的过程中可能会导致部分key逐出。 3. 变更完成后需要对实例重新**创建告警规则**。 4. 请确保主备实例的应用中没有直接引用只读IP或只读域名。 5. 请确保您的客户端应用具备重连机制和处理异常的能力，否则在变更规格后有可能需要重启客户端应用。 6. 变更规格过程中会有秒级业务中断，建议在业务低峰时进行变更。 7. 主备实例如果创建了ACL账号，不支持变更为读写分离实例。 8. Redis 6.0如果开启了SSL链路加密传输，不支持变更为读写分离实例。

除了上表中提到的实例外，其他实例类型目前不支持实例类型的变更。若您想实现跨实例类型的规格变更，建议您创建新的实例后进行数据迁移并交换实例IP，迁移操作请参考**使用迁移任务在线迁移Redis实例**进行操作。

实例类型变更后支持的命令，请参考对应的**开源命令兼容性**。

实例规格变更须知

- **支持实例规格变更明细如下：**

表 6-4 实例规格变更明细

缓存类型	单机实例	主备实例	Cluster集群实例	Proxy集群实例	读写分离实例
Redis 3.0	支持扩容和缩容	支持扩容和缩容	-	仅支持扩容	-
Redis 4.0	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	支持扩容和缩容	支持扩容、缩容和副本数变更
Redis 5.0	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	支持扩容和缩容	支持扩容、缩容和副本数变更

缓存类型	单机实例	主备实例	Cluster集群实例	Proxy集群实例	读写分离实例
Redis 6.0基础版	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	支持扩容缩容	支持扩容、缩容和副本数变更
Redis 6.0企业版	-	支持扩容和缩容	-	-	-
Redis 7.0	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	-	-
Memcached	支持扩容和缩容	支持扩容和缩容	-	-	-

说明

- Redis 3.0和Memcached实例在预留内存不足的情况下，内存用满可能会导致扩容失败，具体可参考**预留内存**。
- 副本数变更和容量变更不支持同时进行，需分开两次执行变更。
- 删除副本时，每次操作仅支持删除一个副本。

- **实例规格变更的影响：**

表 6-5 实例规格变更的影响

实例类型	规格变更类型	实例规格变更的影响
单机、主备和读写分离实例	扩容/缩容	• Redis 4.0及以上版本基础版实例，扩容期间连接会有秒级中断，大约1分钟的只读，缩容期间连接不会中断。 • Redis 3.0实例，规格变更期间连接会有秒级中断，5~30分钟只读。 • Redis企业版实例，规格变更期间连接会有秒级中断，大约1分钟的只读。 • 如果是扩容，只扩大实例的内存，不会提升CPU处理能力。 • 单机实例不支持持久化，变更规格不能保证数据可靠性。在实例变更后，需要确认数据完整性以及是否需要再次填充数据。如果有重要数据，建议先把数据用迁移工具迁移到其他实例备份。 • 主备和读写分离实例缩容前的备份记录，缩容后不能使用。如有需要请提前下载备份文件，或缩容后重新备份。

实例类型	规格变更类型	实例规格变更的影响
Proxy和Cluster集群实例	扩容/缩容	• 水平扩容（分片数增加）： – **连接不中断，但会占用CPU，导致性能有20%以内的下降。** – 分片数增加时，会新增数据节点，数据自动负载均衡到新的数据节点，访问时延会增大。 • 水平缩容（分片数减少）： – 分片数减少时，会删除节点。**Cluster集群实例缩容前，请确保应用中没有直接引用这些删除的节点，否则可能导致业务访问异常。** – **删除节点会导致连接闪断**，请确保您的客户端应用具备重连机制和处理异常的能力，否则在变更规格后可能需要重启客户端应用。 • 垂直扩容（分片数不变，分片容量增加）： – **如果节点所在的虚拟机内存容量不足，会发生节点迁移，迁移时业务连接会有闪断和只读。** – 如果虚拟机内存容量充足，则直接扩大节点容量，对业务无影响。 **说明** Redis 3.0版本集群实例不支持垂直扩缩容。 • 垂直缩容（分片数不变，分片容量减少）：无影响。 • 实例缩容前，每个节点的已用内存要小于缩容后节点最大内存的70%，否则将不允许变更。 • **实例规格变更期间，可能会进行数据迁移，访问时延会增大。Cluster集群请确保客户端能正常处理MOVED和ASK命令，否则会导致请求失败。** • 实例规格变更期间，如果有大批量数据写入导致节点内存写满，将会导致变更失败。 • 在实例规格变更前，请先使用缓存分析中的大key分析，**确保实例中没有大key存在**，否则在规格改变后，节点间进行数据迁移的过程中，单个key过大（≥512MB）会触发Redis内核对于单key的迁移限制，造成数据迁移超时失败，进而导致规格变更失败，key越大失败的概率越高。 • **Cluster集群实例扩容或缩容时，请确保客户端开启集群拓扑自动刷新配置**，否则在变更后需要重启客户端。Lettuce客户端开启集群拓扑自动刷新配置请参考**Lettuce客户端连接Cluster集群实例**中的示例。 • 实例规格变更前的备份记录，变更后不能使用。如有需要请提前下载备份文件，或变更后重新备份。

实例类型	规格变更类型	实例规格变更的影响
主备、读写分离和Cluster集群实例	副本数变更	• Cluster集群实例增加或删除副本时，请确保客户端开启集群拓扑自动刷新配置，否则在变更后需要重启客户端。Lettuce客户端开启集群拓扑自动刷新配置请参考**Lettuce客户端连接Cluster集群实例**中的示例。 • 删除副本会导致连接中断，需确保您的客户端应用具备重连机制和处理异常的能力，否则在删除副本后需要重启客户端应用。增加副本不会连接中断。 • 当副本数已经为实例支持的最小副本数时，不支持删除副本。

6.7 Redis/Memcached 实例变更失败的原因

- 检查是否有其他任务在执行。

  实例变更过程中，同时有其他任务在执行。例如实例正在重启的同时，执行删除或扩容操作，或者实例正在扩容的时候，执行删除操作。

  遇到实例变更操作失败，可以稍后尝试，如果仍然存在问题，请提工单联系技术支持。
- 如果是主备变更为Proxy集群，请确认主备实例DB0以外的DB是否有数据，如果非DB0外的其他DB上有数据（如DB1有数据），会出现变更失败。

  数据必须是只存储在DB0上的主备实例才支持变更为Proxy集群。

6.8 DCS 实例如何缩容?

DCS实例支持扩容和缩容明细如下**表6-6**。

表 6-6 实例规格变更明细

缓存类型	单机实例	主备实例	Cluster集群实例	Proxy集群实例	读写分离实例
Redis 3.0	支持扩容和缩容	支持扩容和缩容	-	仅支持扩容	-
Redis 4.0	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	支持扩容和缩容	支持扩容、缩容和副本数变更
Redis 5.0	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	支持扩容和缩容	支持扩容、缩容和副本数变更

缓存类型	单机实例	主备实例	Cluster集群实例	Proxy集群实例	读写分离实例
Redis 6.0 基础版	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	支持扩容缩容	支持扩容、缩容和副本数变更
Redis 6.0 企业版	-	支持扩容和缩容	-	-	-
Redis 7.0	支持扩容和缩容	支持扩容、缩容和副本数变更	支持扩容、缩容和副本数变更	-	-
Memcached	支持扩容和缩容	支持扩容和缩容	-	-	-

实例扩容、缩容操作请参考**规格变更**。

如果Redis 3.0 Proxy集群需要缩容，可以先进行数据备份，然后另外创建对应规格的Proxy集群实例，使用备份文件导入方式，将备份数据文件导入到新的Proxy集群实例。待数据迁移完成后，再释放原来规格的Proxy集群实例。在线迁移操作，可以参考**备份文件导入方式**。

6.9 Redis 集群实例如何内存不变，只扩分片数?

Proxy集群和Cluster集群实例创建后，支持变更单分片容量，从而实现内存不变，只增加分片数。

例如，单分片容量2GB，分片数为4，内存为8GB的实例，可以变更为单分片容量1GB，分片数为8，内存为8GB的实例。

约束与限制

当已创建实例的单分片容量为最小容量1GB时，单分片容量不能变更。

操作步骤

步骤1 登录**分布式缓存服务管理控制台**。

步骤2 在管理控制台左上角单击 ，选择区域和项目。

步骤3 单击左侧菜单栏的“缓存管理”。

步骤4 在需要规格变更的实例右侧，单击“操作”栏下的“更多 > 变更规格”，进入到变更实例规格页面。

步骤5 在变更实例规格页面中，选择需要的“单分片容量”和“实例规格”。

图 6-1 选择单分片容量

步骤6 单击“下一步”，确认变更详情，然后单击“提交订单”，开始变更DCS缓存实例。

实例规格变更大约需要5到30分钟，实例规格变更成功后，实例状态切换为“运行中”。

----结束

6.10 使用 Lettuce 连接 Cluster 集群实例时，规格变更的异常处理

问题现象

使用lettuce连接Cluster集群实例，实例执行规格变更后，分片数有变化时，部分槽位（Slot）会迁移到新分片上，当客户端连接到新分片时会出现以下异常问题：

图 6-2 异常现象

```
org.springframework.data.redis.RedisSystemException: Redis exception; nested exception is io.lettuce.core.RedisException: io.lettuce.core.R
edisException: java.lang.IllegalArgumentException: Connection to 192.168.70.125:6379 not allowed. This connection point is not known in the
 cluster view
```

详情可参考Lettuce社区：**Connection to X not allowed. This connection point is not known in the cluster view**.

问题分析

Cluster集群规格变更原理：

客户端根据RESP2协议的内容，启动后从Cluster集群获取节点拓扑信息（Cluster Nodes），并将其拓扑关系维护在客户端的内存数据结构中。

对于数据访问，客户端会根据Key值按照CRC16算法进行Hash计算Slot信息，根据内存中保存的节点拓扑关系和Slot的对应信息进行请求自动路由。

在扩容/缩容过程中，当实例分片数发生变化时，存在节点拓扑关系和Slot对应信息的变化，需要客户端进行拓扑关系的自动更新，否则可能造成请求路由失败或者路由位置错误等，造成客户端访问报错。

例如，3分片Cluster集群实例扩容为6分片Cluster集群实例时，节点拓扑关系和Slot对应信息变化如下图所示：

图 6-3 Cluster 集群实例扩容前

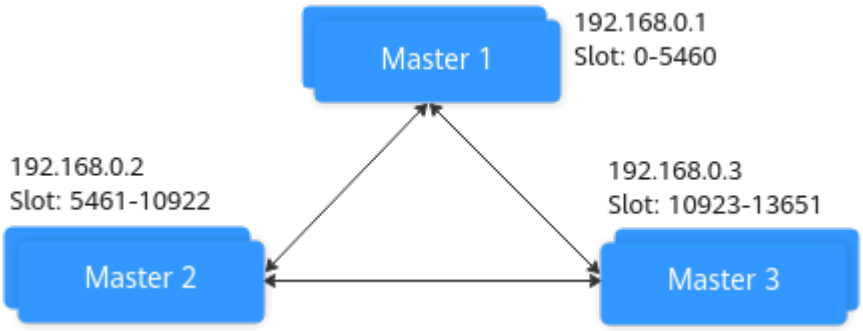
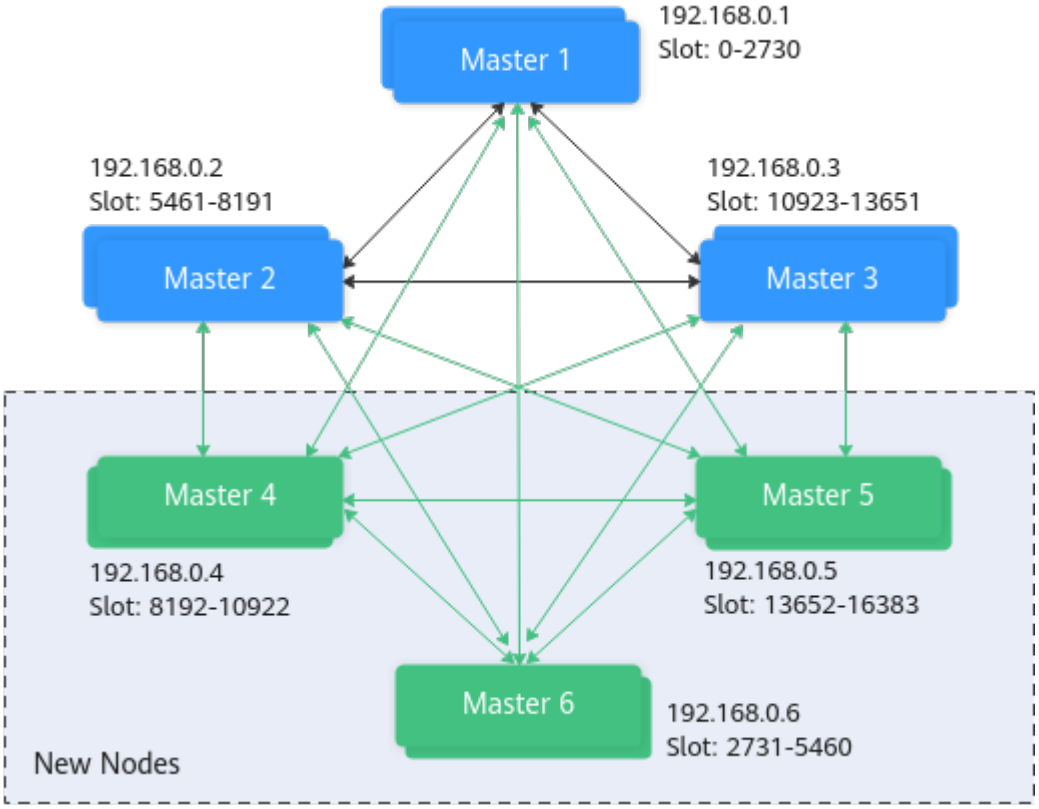


图 6-4 Cluster 集群实例扩容后



解决方案

方案一（推荐方案）：

开启Cluster集群自动刷新拓扑配置。

```
ClusterTopologyRefreshOptions topologyRefreshOptions = ClusterTopologyRefreshOptions.builder()
        // 每隔time毫秒周期性刷新
        .enablePeriodicRefresh(Duration.ofMillis(time))
      // MOVED重定向, ASK重定向, 重连, 未知节点(since 5.1), 槽位不在当前所有分片中(since 5.2),当出现这五种情况时会触发自适应刷新
        .enableAllAdaptiveRefreshTriggers()
        .build();
```

具体实现请参考**Lettuce客户端连接Cluster集群实例**。

说明

Lettuce客户端连接Cluster集群实例，如果未开启拓扑刷新，规格变更后，需要重启客户端。

方案二：

关闭“验证集群节点成员资格开关”，关闭方式如下：

```
ClusterClientOptions clusterClientOptions = ClusterClientOptions.builder()
        .validateClusterNodeMembership(false)
        .build();
```

原理：若validateClusterNodeMembership为true时，连接前检查当前连接地址是否在集群拓扑关系中（通过CLUSTER NODES获得），若不在则会出现上述异常问题。

说明

关闭“验证集群节点成员资格开关”的影响：

- 缺少防止安全漏洞的检验；
- 若未开启集群自动刷新拓扑，当Cluster集群执行变更规格后，若分片数增加时，可能会产生MOVED重定向请求，这个重定向过程会增加集群的网络负担和单次请求耗时；若分片数因删除减少时，会出现无法连接已删除分片的异常情况。

6.11 集群实例是否支持单分片扩容（垂直扩容）

Redis 3.0集群实例不支持垂直扩容，Redis 4.0及以上版本的集群实例如需垂直扩容实例单分片容量，请联系技术支持添加垂直扩容特性的白名单。

垂直扩容的影响请参考**实例规格变更的影响**。

操作步骤

步骤1 登录**分布式缓存服务管理控制台**。

步骤2 在管理控制台左上角单击 ，选择实例所在的区域。

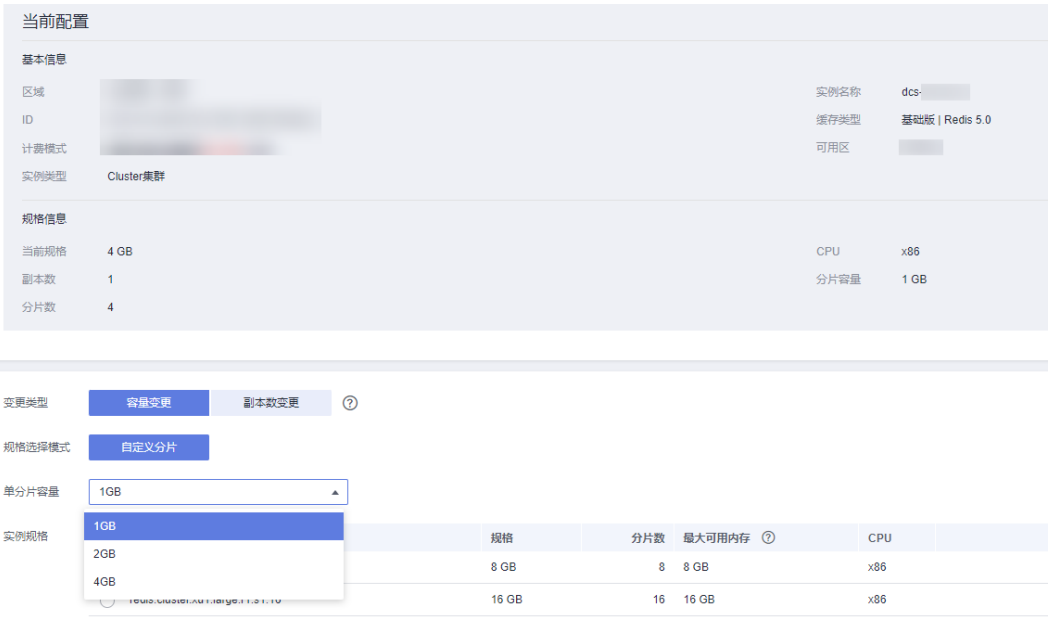
步骤3 单击左侧菜单栏的“缓存管理”。

步骤4 在需要规格变更的实例右侧，单击“操作”栏下的“更多 > 变更规格”，进入到变更实例规格页面。

步骤5 选择“规格选择模式”为“自定义分片”，并选择需要的“单分片容量”。

如果未开启该特性的白名单，默认仅支持选择不大于当前单分片容量的数值。

图 6-5 设置单分片容量



----结束

7 数据备份/导出/迁移

7.1 DCS 实例是否兼容低版本 Redis 迁移到高版本

支持，目前Redis高版本是支持兼容低版本的。

源端是DCS Redis，自建Redis，或者其他云厂商Redis的低版本或相同版本实例，都可以迁移到DCS的目标端实例。

7.2 不同类型的操作系统间进行数据传递和操作，需要注意什么?

建议将数据文件格式转换后再执行导入。

windows系统转换成类unix系统的文件格式：

dos2unix ***{filename}***

类unix系统转换成windows系统的文件格式：

unix2dos ***{filename}***

7.3 源 Redis 使用了多 DB，能否迁移数据到集群实例?

DCS单机、读写分离和主备实例支持256个库，编号0-255。

- 如果目的实例为Cluster集群实例。Cluster集群实例只有1个库。

  两个解决思路：

  a. 源Redis的不同DB合到同一个数据库。

  b. 申请多个DCS缓存实例。

  迁移后实例连接地址和数据库编号有变化，业务注意改造和适配。

- 如果目的实例为Proxy集群。

  Proxy集群默认不开启多DB，仅有一个DB0，请参考**开启多DB操作**开启Proxy集群多DB设置。再进行迁移。

7.4 源 Redis 迁移到集群实例中有哪些限制和注意事项?

- Proxy版集群实例

  使用方式与单机、主备实例类似，但是默认只有1个DB，不支持select命令。数据文件批量导入时，遇到select命令会返回错误提示并忽略，同时继续将剩余数据导入。

  举例：

  源Redis在数据库编号0和2中有数据，生成的AOF或RDB文件包含了这两个库。

  在导入到Proxy集群实例时会忽略“select 2”的命令，然后继续导入源数据库2中的数据到DB0中。

  用户需要注意以下：

  - 源Redis中不同数据库包含了相同的key，则导入时，编号靠前的数据库的key的value会被靠后的数据库中的key覆盖。
  - 源Redis使用了多个数据库，数据迁移到DCS集群实例后，都存储在同一数据库中，不支持select命令。业务需要做适配。

- Cluster版集群实例

  Cluster版集群除了只有1个DB外，导入方式与其他类型的Redis实例也有差异。Cluster集群的数据，必须由客户端分别连接各分片节点，将数据分别导入。各分片节点的IP地址查询命令：

  redis-cli -h *{Redis Cluster IP}* -p 6379 -a *{password}* cluster nodes

  返回的节点地址清单中，标记为master的节点IP地址即为Cluster集群的分片节点地址。

7.5 在线迁移需要注意哪些?

- 网络

  在线迁移首先需要打通网络，迁移任务必须和源Redis、DCS缓存实例二者网络互通。

- 工具

  在线迁移工具，推荐使用DCS控制台的在线迁移功能。

- 数据完整性

  如果选择中断业务，则迁移完成后检查数据量和关键key。

  如果选择不中断业务，则用户需要考虑增量数据的迁移。

- 迁移过程源端扩容影响迁移结果

  在线迁移期间源端扩容操作会影响迁移，有可能导致迁移失败，也有可能会影响客户的数据，客户如果在迁移期间源端实例的内存不够用需要扩容，建议先中断迁移任务 ，然后再扩容。

- 迁移时间

  迁移操作建议在业务低峰期进行。

- 版本限制

  低版本可以到高版本，高版本也可以到低版本，不同版本，在迁移时需要分析业务系统使用到的缓存命令在目的端实例是否兼容。

- 多db限制

  如果目标端与源端均使用DCS的Proxy集群实例，请注意二者的multi-db参数需要配置一致，否则会导致迁移失败。

7.6 在线迁移能否做到完全不中断业务?

可以使用应用双写的方式，即在迁移过程中业务数据继续从源Redis中正常读取，同时将数据的增删改操作在华为云DCS的Redis实例中执行一遍。

保持以上状态运行一段时间后（等待较多的旧数据过期删除），把系统的缓存数据库正式切到华为云DCS。如涉及业务系统迁移云服务，需要在缓存数据库切换前完成业务系统的部署。

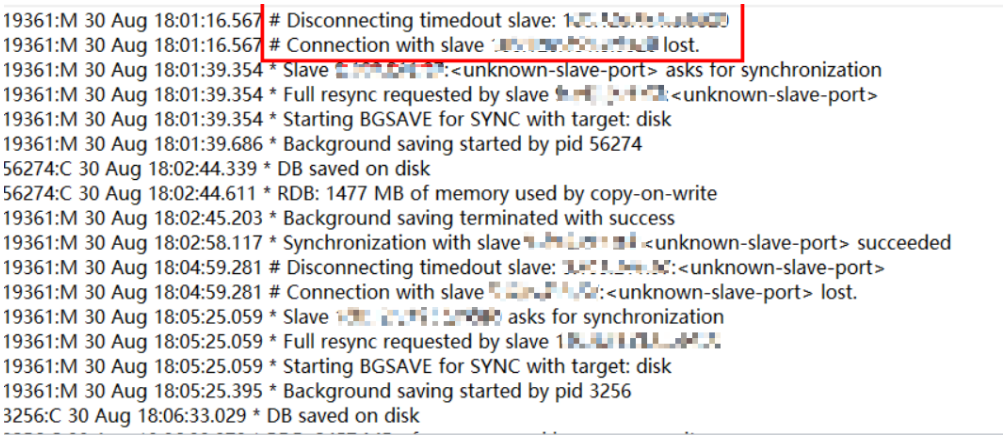
不推荐使用这种方式。原因如下：

1. 网络无法保证稳定快速，如果源Redis实例不在DCS，则需要使用公网访问DCS，效率不高。
2. 同时写2份数据，需要用户自行修改代码实现。
3. 源Redis实例的数据逐出策略各有差异，迁移耗时可能较长，数据完整性保障难度大。

7.7 在线迁移实例源端报“Disconnecting timedout slave”和“overcoming of output buffer limits”

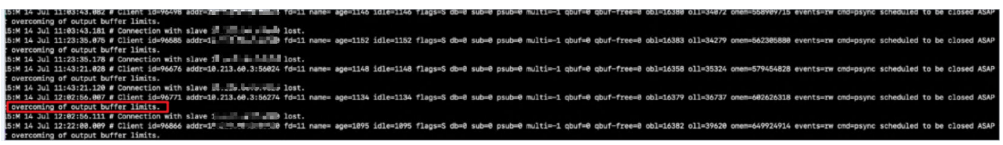
当进行在线迁移时可能会出现如下报错：

- 源端报“**Disconnecting timedout slave**”，如下图：

  解决方法：建议将源端Redis实例的repl-timeout参数值配置为300秒。

- 源端报“**overcoming of output buffer limits**”，如下图：

解决方法：建议将源端Redis实例的client-output-buffer-limit参数值配置为实例最大内存的20%。如果源端Redis是DCS实例，请配置client-output-buffer-slave-hard-limit和client-output-buffer-slave-soft-limit参数值为实例最大内存的20%。

7.8 如何导出 Redis 实例数据?

- 除单机实例外，DCS其他类型实例都支持通过管理控制台导出实例数据：
  a. 进入DCS管理控制台“缓存管理”页面。
  b. 单击需要导出数据的实例名称，进入该实例详情页面。
  c. 单击“备份与恢复”，查看该实例的备份记录。
  d. 如没有备份记录，执行手动备份后，单击“下载”，根据提示完成数据的下载操作。

  说明

  如果您的实例创建时间非常早，由于实例版本没有升级而无法兼容备份恢复功能，请联系技术支持将缓存实例升级到最新版本，升级后就可以支持备份恢复功能。
- 单机实例不支持备份功能，用户可以通过Redis-cli客户端导出rdb文件，但是使用Redis-cli导出rdb文件依赖SYNC命令。
  - 放通了SYNC命令的单机实例（例如Redis 3.0单机实例，未禁用SYNC命令），可以通过执行以下命令，将单机实例上的数据导出：

    redis-cli -h {source_redis_address} -p 6379 [-a password] --rdb {output.rdb}
  - 禁用了SYNC命令的单机实例（例如Redis 4.0/5.0/6.0单机实例，禁用了SYNC命令），建议将单机实例的数据迁移到主备实例，然后使用主备实例的控制台备份恢复功能。

7.9 使用 Rump 工具迁移数据，命令执行后无报错，但 Redis 容量无变化

Rump工具的具体使用，请参考**使用Rump在线迁移其他云厂商Redis**。

可能原因：

- Rump工具不支持迁移到集群实例。
- Rump命令参数有误。

7.10 是否支持控制台导出 RDB 格式的 Redis 备份文件?

- Redis 3.0实例

  Redis 3.0是通过AOF文件持久化的，控制台仅支持备份和下载AOF格式的备份文件。如果需要RDB格式的备份文件可以通过Redis-cli导出：

  redis-cli -h *{redis_address}* -p 6379 [-a *password*] --rdb {output.rdb}
- Redis 4.0及以上基础版实例

  Redis 4.0及以上基础版实例支持在控制台备份和下载AOF或RDB格式的备份文件。
- Redis 6.0企业版实例

  企业版高性能型实例支持备份和下载AOF或RDB格式的备份文件。

  企业版存储型实例仅支持备份和下载RDB格式的备份文件。

说明

单机实例不支持通过控制台备份数据。

7.11 缓存实例备份文件如何存放？备份文件的数量是否有限制？

DCS缓存实例备份文件存储在对象存储服务（OBS）中。目前每个缓存实例最多支持存储24个备份文件，当备份文件超过24个时，会自动删除最早的备份文件。

7.12 Redis 在线数据迁移是迁移整个实例数据么？

如果是单机、主备、读写分离多DB的实例之间进行迁移，是迁移实例所有的数据，不管存在哪个DB都会进行迁移，且数据所在的DB序号不会变。您也可以选择迁移单一DB进行迁移。

如果是集群实例，由于集群实例默认只有一个DB0节点，会迁移DB0上所有槽内的数据。

7.13 AOF 文件在什么情况下会被重写

AOF文件重写涉及到以下概念。

- 重写时间窗：目前该时间窗为凌晨1:00 - 4:59。
- 磁盘阈值：即磁盘的使用率超过50%，即认为达到阈值。
- 数据集使用内存：实例的一个监控指标，用于统计Redis中数据集占用的内存。

AOF文件在以下三种情况下会被重写。

- 如果磁盘达到阈值，无论是否处于时间窗内：当AOF文件大小 > 数据集使用内存时，实例AOF文件会被重写。
- 如果磁盘未达到阈值，处于重写时间窗内：当AOF文件大小 > 数据集使用内存的1.5倍时，实例AOF文件会被重写。
- 如果磁盘未达到阈值，未处于重写时间窗内：当AOF文件大小 > 实例最大内存的4.5倍时，实例AOF文件会被重写。

7.14 Redis 迁移失败有哪些常见原因？

- 在进行数据迁移时，如果Redis实例发生了主备倒换，可能会导致迁移失败。可联系技术支持，将主备倒换关闭，待迁移成功后，再开启主备倒换。
- 如果是在线迁移，请确认源Redis实例，是否禁用了SYNC和PSYNC命令，如果禁用了，需要先开启，允许数据同步。
- 如果是单机/主备实例迁移到Proxy集群实例，Proxy集群默认不开启多DB，仅有一个DB0，请先确保单机/主备实例DB0以外的DB是否有数据，如果有，请先参考**开启多DB操作**开启Proxy集群多DB设置。
- 如果是单机/主备实例迁移到Cluster集群实例，Cluster集群不支持多DB，仅有一个DB0，请先确保单机/主备实例DB0以外的DB是否有数据，如果有，请将数据转

存到DB0，否则会出现迁移失败，将数据转存到DB0的操作请参考**使用Rump在线迁移**。

7.15 一个数据迁移能迁移到多个目标实例么？

不能，一个迁移任务只能迁移到一个目标实例。要迁移到多个目标实例需要创建多个迁移任务。

7.16 怎么放通 SYNC 和 PSYNC 命令？

- DCS云服务内部的Redis之间进行迁移：
  - 如果迁移任务和源端实例在相同账号下的相同Region，在配置在线迁移任务时，源端实例通过选择DCS实例（云服务Redis）的方式进行配置，会自动放通源端实例的SYNC和PSYNC命令。
  - 如果迁移任务和源端实例在不同账号或不同Region，在配置在线迁移任务时，源端实例不能通过选择DCS实例（云服务Redis）的方式进行配置，不会自动放通源端实例的SYNC和PSYNC命令，因此无法使用控制台的在线迁移。推荐使用备份文件导入方式迁移。
  - 自建Redis迁移至DCS，默认没有禁用SYNC和PSYNC命令。
- 其他云厂商迁移到DCS云服务：
  - 一般云厂商都是禁用了SYNC和PSYNC命令，如果使用DCS控制台的在线迁移功能，需要联系源端的云厂商运维人员放通此命令。离线迁移，推荐使用备份文件导入方式。
  - 如果不需要增量迁移，可以参考**使用Redis-shake工具在线全量迁移其他云厂商Redis**进行全量迁移，该方式不依赖于SYNC和PSYNC。

7.17 迁移或导入备份数据时，相同的 Key 会被覆盖吗？

在迁移或导入备份数据时，源端与目标端重复的数据会被覆盖；源端没有，目标端有的数据会保留。

因此，如果在迁移后目标端与源端数据不一致，可能是目标端在迁移前有未清除的数据。

7.18 Cluster 集群实例使用内置 key 且跨 slot 的 Lua 脚本时迁移失败

当源实例是Cluster集群，且使用了内置key且跨slot的Lua脚本，迁移到DCS集群实例失败时，可以将目标端改为主备或读写分离实例。

在Cluster集群扩缩容、slot迁移等slot分布会发生变化的场景，内置key且跨slot的Lua脚本可能执行报错。因此，**不建议Cluster集群实例使用内置key且跨slot的Lua脚本。**

说明

Cluster集群实例支持内置key且跨slot的Lua脚本：

- 内置key：将Key写入lua脚本中，不作为参数传入。
- 跨slot：Lua脚本中涉及的所有slot属于一个分片。

问题现象

源实例是Cluster集群，且使用了内置key且跨slot的Lua脚本，迁移到DCS集群实例中时，可能会导致在线/备份导入迁移失败。

解决方案

迁移的目标端实例选择主备或读写分离实例。

问题建议

不建议Cluster集群实例使用内置key且跨slot的Lua脚本。

说明

- Cluster集群实例支持内置key且跨slot的Lua脚本：
  - 内置key：将Key写入lua脚本中，不作为参数传入。
  - 跨slot：Lua脚本中涉及的所有slot属于一个分片。
- 在Cluster集群扩缩容、slot迁移等slot分布会发生变化的场景，内置key且跨slot的Lua脚本可能执行报错。

7.19 迁移故障处理

本章节介绍了在迁移过程中不同迁移故障的处理建议。

重新启动数据同步失败

重新启动数据同步失败的处理建议：

1. **分析源Redis是否存在大Key**，如果源Redis存在大key，建议将大key打散成多个小key后再迁移。
2. 检查目标Redis的规格是否小于迁移数据大小、是否有其他任务在执行。
   - 如果目标Redis的实例规格小于迁移数据大小，迁移过程中，内存被占满，会导致迁移失败。
   - 如果目标Redis存在正在执行的主备倒换，建议联系客服关闭主备倒换后，重新执行数据迁移。待迁移完成后，重新开启主备倒换。
3. 提供错误信息，联系客服。

Redis 服务地址不通

Redis服务地址不通，建议从以下几个方面做排查：

- **Redis和ECS之间的连接问题**
- **公网连接Redis**

- **密码问题**
- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**
- **性能问题导致连接超时**

Redis 认证鉴权失败

redis认证鉴权失败的处理建议：

确保源Redis和目标Redis密码输入正确，且迁移过程中密码未被修改。

如果忘记了密码，**重置缓存实例密码**后，需重新配置迁移任务。

解析 RDB 失败

解析RDB失败的处理建议：

请检查源端Redis日志，通常是全量同步过久或者增量过大，导致output buffer打满。通常解决办法有如下几种:

- 修改增大源端output buffer的大小，通过**修改output-buffer-limit参数**，建议采用这种方式。
- 增大shake全量同步的并发度，调高parallel。
- 在业务低峰期再进行同步。

从断点恢复失败

从断点恢复失败，建议从以下几个方面做排查 ：

- **Redis和ECS之间的连接问题**
- **公网连接Redis**
- **密码问题**
- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**
- **性能问题导致连接超时**

Redis 的 IP 和 port 参数非法

Redis的IP和port参数非法的处理建议：请提供错误信息，联系客服。

任务失败

任务失败的处理建议：提供错误信息，联系客服。

下载文件失败

下载文件失败的处理建议：

参见**下载对象失败**处理步骤。

集群不支持 AOF 格式文件导入

集群不支持AOF格式文件导入的处理建议：

Cluster集群仅支持导入.rdb备份文件，不支持.aof备份文件。

迁移 AOF 文件到目标 Redis 失败

迁移AOF文件到目标redis失败，建议从以下几个方面做排查：

- **Redis和ECS之间的连接问题**
- **公网连接Redis**
- **密码问题**
- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**
- **性能问题导致连接超时**

迁移 RDB 文件到目标 Redis 失败

迁移RDB文件到目标redis失败，建议从以下几个方面做排查：

- **Redis和ECS之间的连接问题**
- **公网连接Redis**
- **密码问题**
- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**
- **性能问题导致连接超时**

解压文件失败

解压文件失败的处理建议：

1. 确保文件未损坏，且文件格式正常。
2. 排查迁移机规格太小，磁盘是否已写满，此情况建议扩容迁移机规格。

不支持该文件格式

不支持该文件格式的处理建议：

只支持.rdb、.aof、.zip、.tar.gz文件格式。

迁移文件失败

迁移文件失败，建议从以下几个方面做排查：

- **Redis和ECS之间的连接问题**
- **公网连接Redis**

- **密码问题**
- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**
- **性能问题导致连接超时**

文件或目录不存在

文件或目录不存在的处理建议：

1. 排查迁移机规格太小，磁盘是否已写满，此情况建议扩容迁移机规格。
2. 提供错误信息，联系客服。

无法连接到源 Redis

无法连接到源Redis的处理建议：

1. 参见**Redis连接失败问题排查和解决**。
2. 排查源Redis规格和迁移机内存大小，迁移机的内存小，源Redis数据量过大，迁移速度过慢使数据堆积在迁移机上时，也会导致该类问题，此情况建议扩容迁移机规格。
3. 排查迁移机路由是否正常，迁移机上执行命令查看：**route - n**
4. 提供错误信息，联系客服。

从源节点导出备份文件失败

从源节点导出备份文件失败，建议从以下几个方面做排查：

- **Redis和ECS之间的连接问题**
- **公网连接Redis**
- **密码问题**
- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**
- **性能问题导致连接超时**

导入备份文件到目标 Redis 失败

导入备份文件到目标Redis失败，建议从以下几个方面做排查：

- **Redis和ECS之间的连接问题**
- **公网连接Redis**
- **密码问题**
- **实例配置问题**
- **客户端连接问题**
- **带宽超限导致连接问题**

- **性能问题导致连接超时**

修改 redis-shake-conf 配置文件失败，参数错误

修改redis-shake-conf配置文件失败的处理建议：

1. 排查迁移机规格太小，磁盘是否已写满，此情况建议扩容迁移机规格。
2. 提供错误信息，联系客服。

同步数据失败，源节点：{0}，目标节点：{1}

同步数据失败的处理建议：

1. **分析源Redis是否存在大Key**，如果源Redis存在大key，建议将大key打散成多个小key后再迁移。
2. 确保目标Redis规格不小于源Redis。查看规格步骤参见**查看实例信息**；
3. 参见**Redis连接失败问题排查和解决**。
4. 排查源Redis规格和迁移机内存大小，迁移机的内存小，源redis数据量过大，迁移速度过慢使数据堆积在迁移机上时，也会导致该类问题，此情况建议扩容迁移机规格。

部署 migration 工具失败

部署migration工具失败的处理建议：

1. 排查数据面到OBS服务网络是否正常。
2. 提供错误信息，联系客服。

在线迁移失败

在线迁移失败的处理建议：提供错误信息，联系客服。

绑定 port 到 ECS 虚拟机失败

绑定port到ECS虚拟机失败的处理建议：

迁移任务底层资源可能不足，需要联系客服处理。

创建迁移 ECS 虚拟机失败

创建迁移ECS虚拟机失败的处理建议：提供错误信息，联系客服。

文件操作异常

文件操作异常的处理建议：

1. 排查源Redis规格和迁移机内存大小，迁移机的内存小，源Redis数据量过大，迁移速度过慢使数据堆积在迁移机上时，也会导致该类问题，此情况建议扩容迁移机规格。
2. 提供错误信息，联系客服。

执行命令异常

执行命令异常的处理建议：

- 错误信息中包含“listening-port”和“REPLCONF”相关，请检查源Redis是否放通SYNC和PSYNC命令，迁移任务底层资源与源Redis、目标Redis网络是否连通。

  在线迁移，必须满足源Redis和目标Redis的网络相通、源Redis已放通SYNC和PSYNC命令这两个前提，否则，会迁移失败。

  - 网络

    检查源Redis、目标Redis、迁移任务所需虚拟机是否在同一个VPC，如果是同一个VPC，则检查安全组（Redis 3.0实例）或白名单（Redis 4.0/5.0实例）是否放通端口和IP，确保网络是连通的；如果不在同一个VPC，则需要**建立VPC对等连接**，打通网络。

    源Redis和目标Redis必须允许迁移任务底层虚拟机访问。实例安全组或白名单配置，请参考**配置安全组**、**配置白名单**。

    源Redis和目标Redis属于不同的云厂商，请参考**云专线**打通网络。

  - 命令

    默认情况下，一般云厂商都是禁用了SYNC和PSYNC命令，如果要放通，需要联系云厂商运维人员放通命令。

    - 华为云内部进行迁移：
      - 自建Redis迁移至DCS，默认没有禁用SYNC和PSYNC命令；
      - 华为云DCS服务之间进行迁移，如果是同一账号相同Region进行在线迁移，在执行迁移时，会自动放通SYNC和PSYNC命令；
      - 如果是不同Region或相同Region不同账号进行在线迁移，不会自动放通SYNC和PSYNC命令，无法使用在线迁移。推荐使用备份文件导入方式迁移。
    - 其他云厂商迁移到华为云：

      一般云厂商都是禁用了SYNC和PSYNC命令，如果使用在线迁移功能，需要联系源端的云厂商运维人员放通此命令，离线迁移，推荐使用备份文件导入方式。

- 错误信息包含“read error”，且为全量迁移过程中失败，数据量过大的情况下，建议开始迁移时不要选择“自动重连”模式，等到进入“增量迁移”后，再选择“自动重连”模式，且**调大repl-timeout的时间值**；同时建议调整源端output buffer参数，buffer参数的大小需要根据源端内存大小来定，比如源端24G的内存大小，可以调整为2G的buffer，命令： **client-output-buffer-limit slave 2gb 2gb 600** 。
- 错误信息中包含“write: connection reset by peer”，可能原因目标Redis内存规格太小导致内存写满，无法同步数据，建议**扩大目标Redis实例规格**，至少与源端实例规格持平。
- 错误信息中包含“read: connection reset by peer”，源Redis为主备，且迁移过程中，频繁发生主备倒换，请**分析源Redis是否存在大Key**，如果源Redis存在大key，建议将大key打散成多个小key后再迁移。也可强行关闭主备倒换，等数据迁移完毕后再开启主备倒换，命令：**config set slave-priority 0**。如果目标Redis为proxy集群，请排查pipeline阈值大小，建议调整proxy节点pipeline阈值为5W，命令： **proxy.config set client-max-pipeline 50000**。
- 提供错误信息，联系客服。

解码或解析失败

解码或解析失败的处理建议：

1. 排查迁移机规格太小，磁盘是否已写满，此情况建议扩容迁移机规格。
2. 提供错误信息，联系客服。

未知或未支持的命令

未知或未支持的命令的处理建议：

检查源Redis有没有放通相关命令，如SYNC和PSYNC，需要联系客服放通命令。

在线迁移，必须满足源Redis和目标Redis的网络相通、源Redis已放通SYNC和PSYNC命令这两个前提，否则，会迁移失败。

- 网络

  检查源Redis、目标Redis、迁移任务所需虚拟机是否在同一个VPC，如果是同一个VPC，则检查安全组（Redis 3.0实例）或白名单（Redis 4.0/5.0实例）是否放通端口和IP，确保网络是连通的；如果不在同一个VPC，则需要**建立VPC对等连接**，打通网络。

  源Redis和目标Redis必须允许迁移任务底层虚拟机访问。实例安全组或白名单配置，请参考**配置安全组**、**配置白名单**。

  源Redis和目标Redis属于不同的云厂商，请参考**云专线**打通网络。

- 命令

  默认情况下，一般云厂商都是禁用了SYNC和PSYNC命令，如果要放通，需要联系云厂商运维人员放通命令。

  - 华为云内部进行迁移：
    - 自建Redis迁移至DCS，默认没有禁用SYNC和PSYNC命令；
    - 华为云DCS服务之间进行迁移，如果是同一账号相同Region进行在线迁移，在执行迁移时，会自动放通SYNC和PSYNC命令；
    - 如果是不同Region或相同Region不同账号进行在线迁移，不会自动放通SYNC和PSYNC命令，无法使用在线迁移。推荐使用备份文件导入方式迁移。
  - 其他云厂商迁移到华为云：

    一般云厂商都是禁用了SYNC和PSYNC命令，如果使用在线迁移功能，需要联系源端的云厂商运维人员放通此命令，离线迁移，推荐使用备份文件导入方式。

同步数据失败

同步数据失败的处理建议：

1. 错误信息包含“key name is busy”，目标Redis对应的key已经存在，删除目标Redis报错的key。
2. 错误信息包含“not in the same slot“，建议进行业务改造，不要在多key命令里面用跨slot的key；也可以把目标Redis用主备实例代替proxy集群实例。
3. 错误信息中包含“read: connection reset by peer”，源Redis为主备，且迁移过程中，频繁发生主备倒换，请**分析源Redis是否存在大Key**，如果源Redis存在大

key，建议将大key打散成多个小key后再迁移；也可强行关闭主备倒换，等数据迁移完毕后再开启主备倒换，命令：**config set slave-priority 0**。如果目标Redis为proxy集群，请排查pipeline阈值大小，建议调整proxy节点pipeline阈值为5W，命令： **proxy.config set client-max-pipeline 50000**。

导入备份文件失败

导入备份文件失败的处理建议：提供错误信息，联系客服。

7.20 数据迁移失败问题排查

在使用控制台进行数据迁移时，如果出现迁移方案选择错误、在线迁移源Redis没有放通SYNC和PSYNC命令、源Redis和目标Redis网络不连通等问题，都会导致迁移失败。

本章节主要介绍使用DCS控制台进行数据迁移时迁移失败的问题排查和解决。

排查步骤

步骤1 单击已创建的迁移任务名称，进入迁移任务页面。

步骤2 查看迁移日志。参照**迁移故障处理**，根据对应的错误日志信息，做对应排查。

步骤3 检查迁移方案是否选择正确。

根据自建Redis迁移至DCS、DCS实例间迁移、其他云厂商Redis服务迁移至DCS的不同场景，选择合适的迁移方案，例如，DCS实例间迁移，高版本不支持迁移到低版本。

迁移方案选择不正确，会导致迁移失败，具体迁移方案，请查看**迁移方案介绍**。

步骤4 检查源Redis是否放通SYNC和PSYNC命令，迁移任务底层资源与源Redis、目标Redis网络是否连通。

如果是在线迁移，才涉及该操作。

在线迁移，必须满足源Redis和目标Redis的网络相通、源Redis已放通SYNC和PSYNC命令这两个前提，否则，会迁移失败。

- 网络

  检查源Redis、目标Redis、迁移任务所需虚拟机是否在同一个VPC，如果是同一个VPC，则检查安全组（Redis 3.0/Redis 6.0企业版实例）或白名单（Redis 4.0及以上基础版实例）是否放通端口和IP，确保网络是连通的；如果不在同一个VPC，则需要**建立VPC对等连接**，打通网络。

  源Redis和目标Redis必须允许迁移任务底层虚拟机访问。实例安全组或白名单配置，请参考**配置安全组**、**配置白名单**。

  迁移任务的安全组“出方向规则”需放通端口和IP，确保迁移任务底层虚拟机可以访问源Redis和目标Redis，请参考**配置安全组**。

  源Redis和目标Redis属于不同的云厂商，请参考**云专线**打通网络。

  说明

  Redis 4.0及以上版本的基础版实例需要放通白名单的目的，是允许迁移任务底层虚拟机可以访问源Redis和目标Redis（迁移任务底层虚拟机会占用一个IP）。同样，如果是Redis 3.0/Redis 6.0企业版实例，需要配置实例安全组的入方向规则对迁移任务底层虚拟机放通。

- 命令

  默认情况下，一般云厂商都是禁用了SYNC和PSYNC命令，如果要放通，需要联系云厂商运维人员放通命令。

  - 华为云内部进行迁移：
    - 自建Redis迁移至DCS，默认没有禁用SYNC和PSYNC命令；
    - 华为云DCS服务之间进行迁移，如果是同一账号相同Region进行在线迁移，在执行迁移时，会自动放通SYNC和PSYNC命令；
    - 如果是不同Region或相同Region不同账号进行在线迁移，不会自动放通SYNC和PSYNC命令，无法使用在线迁移。推荐使用备份文件导入方式迁移。
  - 其他云厂商迁移到华为云：

    一般云厂商都是禁用了SYNC和PSYNC命令，如果使用在线迁移功能，需要联系源端的云厂商运维人员放通此命令，离线迁移，推荐使用备份文件导入方式。

步骤5 检查源Redis是否存在大Key。操作请参见**分析源Redis是否存在大Key**。

如果源Redis存在大key，建议将大key打散成多个小key后再迁移。

步骤6 检查目标Redis的规格是否大于迁移数据大小、是否有其他任务在执行。

如果目标Redis的实例规格小于迁移数据大小，迁移过程中，内存被占满，会导致迁移失败。

如果目标Redis存在正在执行的主备倒换，建议联系客服关闭主备倒换后，重新执行数据迁移。待迁移完成后，重新开启主备倒换。

步骤7 如果是单机/主备实例迁移到集群实例，请检查：

- 如果是单机/主备实例迁移到Proxy集群实例，Proxy集群默认不开启多DB，仅有一个DB0，请先确保单机/主备实例DB0以外的DB是否有数据，如果有，请先参考**开启多DB操作**开启Proxy集群多DB设置。
- 如果是单机/主备实例迁移到Cluster集群实例，Cluster集群不支持多DB，仅有一个DB0，请先确保单机/主备实例DB0以外的DB是否有数据，如果有，请将数据转存到DB0，否则会出现迁移失败，将数据转存到DB0的操作请参考**使用Rump在线迁移**。

步骤8 检查迁移操作是否正确。

检查填写的IP地址、实例密码是否正确。

步骤9 排查白名单。

步骤10 如果无法解决，请联系客服。

----结束

7.21 Memcached 如何迁移?

由于Memcached本身没有命令可遍历查询所有数据，因此无法从正在使用的Memcached中直接导出数据、并迁移到云缓存Memcached实例。

建议通过应用程序写日志的方式记录缓存key、同时提取key-value，写入云缓存Memcached，以逐步完成数据的迁移。

说明

部分开源工具能够利用Memcached的stats cachedump命令结合get操作查询出Memcached中的部分key-value，但由于stats cachedump命令限制最多只能查出不超过2MB的key（查询出的key总长度和不超过2MB，且该长度限制包含每个key 20余字节的辅助信息），因此也不能通过此类工具或类似方法执行数据迁移。

7.22 是否支持 Memcached 和 Redis 之间实例数据的迁移?

不支持，Memcached和Redis属于不同的缓存数据库，不支持互相迁移数据。

8 大 Key/热 Key 分析/过期 Key 扫描

8.1 什么是大 Key/热 Key?

名词	定义
大Key	大Key可以分为两种情况： • Key的Value占用存储空间较大。一般单个String类型的Key大小达到10KB，或者集合类型的Key总大小达到50MB，则被定义为大Key。 • Key的元素较多。一般集合类型的Key中元素超过5000个，则被定义为大Key。
热Key	通常当一个Key的访问频率或资源占用显著高于其他Key时，则称之为热Key。例如： • 某个集群实例一个分片每秒处理10000次请求，其中有3000次都是操作同一个Key。 • 某个集群实例一个分片的总带宽使用（入带宽+出带宽）为100Mbits/s，其中80Mbits是由于对某个Hash类型的Key执行HGETALL所占用。

8.2 存在大 Key/热 Key，有什么影响?

类别	影响
大Key	**造成规格变更失败。** Redis集群变更规格过程中会进行数据rebalance（节点间迁移数据），单个Key过大的时候会触发Redis内核对于单Key的迁移限制，造成数据迁移超时失败，Key越大失败的概率越高，大于512MB的Key可能会触发该问题。

类别	影响
	造成数据迁移失败。 数据迁移过程中，如果一个大Key的元素过多，则会阻塞后续Key的迁移，后续Key的数据会放到迁移机的内存Buffer中，如果阻塞时间太久，则会导致迁移失败。
	容易造成集群分片不均的情况。 • 各分片内存使用不均。例如某个分片占用内存较高甚至首先使用满，导致该分片Key被逐出，同时也会造成其他分片的资源浪费。 • 各分片的带宽使用不均。例如某个分片被频繁流控，其他分片则没有这种情况。
	客户端执行命令的时延变大。 对大Key进行的慢操作会导致后续的命令被阻塞，从而导致一系列慢查询。
	导致实例流控。 对大Key高频率的读会使得实例出方向带宽被打满，导致流控，产生大量命令超时或者慢查询，业务受损。
	导致主备倒换。 对大Key执行危险的DEL操作可能会导致主节点长时间阻塞，从而导致主备倒换。
热Key	**容易造成集群分片不均的情况。** 造成热Key所在的分片有大量业务访问而同时其他的分片压力较低。这样不仅会容易产生单分片性能瓶颈，还会浪费其他分片的计算资源。
	使得CPU冲高。 对热Key的大量操作可能会使得CPU冲高，如果表现在集群单分片中就可以明显地看到热Key所在的分片CPU使用率较高。这样会导致其他请求受到影响，产生慢查询，同时影响整体性能。业务量突增场景下甚至会导致主备切换。
	易造成缓存击穿。 热Key的请求压力过大，超出Redis的承受能力易造成缓存击穿，即大量请求将被直接指向后端的数据库，导致数据库访问量激增甚至宕机，从而影响其他业务。

8.3 为了减少大 Key 和热 Key 过大，有什么使用建议?

- **string类型控制在10KB以内**，hash、list、set、zset**元素尽量不超过5000个**。
- Key的命名前缀为业务缩写，禁止包含特殊字符(比如空格、换行、单双引号以及其他转义字符)。
- Redis事务功能较弱，不建议过多使用。

- 短连接性能差，推荐使用带有连接池的客户端。
- 如果只是用于数据缓存，容忍数据丢失，建议关闭持久化。
- 大Key/热Key的优化方法，请参考下表。

类别	方法
大Key	**进行大Key拆分。** 分为以下几种场景： • **该对象为String类型的大Key：**可以尝试将对象分拆成几个Key-Value， 使用MGET或者多个GET组成的pipeline获取值，分拆单次操作的压力。如果是集群实例，由于集群实例包含多个分片，拆分后的Key会自动平摊到集群实例的多个分片上，从而降低对单个分片的影响。 • **该对象为集合类型的大Key，并且需要整存整取：**在设计上严格禁止这种场景的出现，因为无法拆分。有效的方法是将该大Key从Redis去除，单独放到其余存储介质上。 • **该对象为集合类型的大Key，每次只需操作部分元素：**将集合类型中的元素分拆。以Hash类型为例，可以在客户端定义一个分拆Key的数量N，每次对HGET和HSET操作的field计算哈希值并取模N，确定该field落在哪个Key上，实现上类似于Redis Cluster的计算slot的算法。
	将大Key单独转移到其余存储介质。 无法拆分的大Key建议使用此方法，将不适用Redis能力的数据存至其它存储介质，如**SFS**或者其余NoSQL数据库，并在Redis中删除该大Key。 **注意** 禁止使用DEL直接删除大Key，可能会造成Redis阻塞，甚至主备倒换。Redis 4.0及以上版本建议采用UNLINK命令删除大Key。
	合理设置过期时间并对过期数据定期清理。 合理设置过期时间，避免历史数据在Redis中大量堆积。由于Redis的惰性删除策略，过期数据可能并不能及时清理，如果发现Redis过期Key清理较慢，建议配置**过期Key扫描**。
热Key	**使用读写分离。** 如果热Key主要是读流量较大，则可以在客户端**配置读写分离**，降低对主节点的影响。还可以增加多个副本以满足读需求，但是备机较多也有相应的影响，DCS主备节点之间使用的是星型复制，即所有的备节点都直接和主节点保持同步，这样能保证备节点之间相互独立，且复制延迟较小。缺点是在备节点数量较多的情况下，主节点的CPU和网络负载会较高。
	使用客户端缓存/本地缓存。 该方案需要提前了解业务的热点Key有哪些，设计客户端/本地和远端Redis的两级缓存架构，热点数据优先从本地缓存获取，写入时同时更新，这样能够分担热点数据的大部分读压力。缺点是需要修改客户端架构和代码，改造成本较高。

类别	方法
	设计熔断/降级机制。 热Key极易造成缓存击穿，高峰期请求都直接透传到后端数据库上，从而导致业务雪崩。因此热Key的优化一定需要设计系统的熔断/降级机制，在发生击穿的场景下进行限流和服务降级，保护系统的可用性。

8.4 如何分析 Redis 3.0 实例的热 Key?

由于Redis 3.0本身不提供热Key能力，您可以参考以下方法进行分析。

- 方法1：进行业务结构和业务实现分析，找到可能的热Key。
  例如，某商品在秒杀，或者用户登录，对业务代码分析，很容易找到热Key。
  优点：简单易行。
  缺点：需要对业务代码比较了解，另外对于一些复杂的业务场景，不太容易分析。
- 方法2：在客户端代码中，调用Redis的函数中，进行访问Key的记录，进而统计出热Key。
  缺点：需要代码进行侵入式修改。
- 方法3：抓包分析。
  优点：简单易行。

8.5 如何提前发现大 Key 和热 Key?

方法	说明
使用DCS自带的大Key和热Key分析工具进行分析	请参考**分析Redis实例大Key和热Key**。
通过redis-cli的bigkeys和hotkeys参数查找大Key和热Key	• Redis-cli提供了bigkeys参数，能够使redis-cli以遍历的方式分析Redis实例中的所有Key，并返回Key的整体统计信息与每个数据类型中Top1的大Key，bigkeys仅能分析并输入六种数据类型（STRING、LIST、HASH、SET、ZSET、STREAM），命令示例为：**redis-cli -h <实例的连接地址> -p <端口> -a <密码> --bigkeys**。 • 自Redis 4.0版本起，redis-cli提供了hotkeys参数，可以快速帮您找出业务中的热Key，该命令需要在业务实际运行期间执行，以统计运行期间的热Key。命令示例为：**redis-cli -h <实例的连接地址> -p <端口> -a <密码> --hotkeys**。热Key的详情可以在结果中的summary部分获取到。

方法	说明
通过Redis命令查找大Key	如果有已知的大Key模式，例如知道其前缀为cloud:msg:test，那么可以通过一个程序，SCAN符合该前缀的Key，然后通过查询成员数量和查询Key大小的相关命令，来判断具体的大Key。 ● 查询成员数量的相关命令：LLEN，HLEN，XLEN，ZCARD，SCARD ● 查询Key占用内存大小的命令：DEBUG OBJECT，MEMORY USAGE **注意** 该方法会大量消耗计算资源，请知晓并评估其风险，不要在业务压力较大的实例使用该方法，否则可能会对正常业务造成影响。
通过redis-rdb-tools工具找出大Key	**redis-rdb-tools**是分析Redis RDB快照文件的开源工具。可以根据需求自定义分析Redis实例中所有Key的内存占用情况。 使用此方法需要在DCS实例备份与恢复页签中**导出实例的rdb文件**。 **注意** 该方法时效性相较于在线分析来说较差，优势在于完全不影响现有业务。

对于Redis 3.0实例，由于Redis 3.0本身不支持热Key分析，推荐可以使用**配置告警**的方法，帮助您发现热Key。

- 配置节点级别的**内存利用率**监控指标的告警。

  如果某个节点存在大key，这个节点比其他节点内存使用率高很多，会触发告警，便于用户发现潜在的大key。
- 配置节点级别的**入网最大带宽**、**出网最大带宽**、**CPU利用率**监控指标的告警。

  如果某个节点存在热key，这个节点的带宽占用、CPU利用率都比其他节点高，该节点会容易触发告警，便于用户发现潜在热key。

8.6 DCS 删除过期 key

问题现象

分布式缓存服务每天定时清理一次过期key是根据什么规则清理的？清理规则可以自己调整么？

过期 key 删除机制

- 惰性删除：Redis的删除策略由主循环中的判断逻辑进行控制，所有Key读写命令执行之前都会调用函数对其进行检查，如果过期，则删除该键，然后返回Key不存在的结果；未过期则不做操作，继续执行原有的命令。
- 定期删除：由Redis的定时任务函数实现，该函数以一定的频率运行，每次运行时，都从键空间中取出一定数量的随机Key进行检查，并删除其中的过期键。

> **说明**
>
> 不是每次定时任务都会检查所有的Key，而是随机检查一定数量的Key，该机制旨在防止阻塞Redis主进程太久而造成业务阻塞，所以会造成已过期的Key释放内存速度较慢。

解决方案

- 配置一个定时的热key扫描，具体操作可参考**热key扫描**或写一个用scan命令扫描全局key的定时任务把key全部遍历一遍，触发已过期的key从内存中删除。
- 通过自行配置定时任务，在任务执行期间，会对所有缓存实例的主节点进行扫描操作，扫描操作会遍历整个实例的键空间，触发Redis引擎中对Key过期的判断，从而释放已过期的Key，具体操作可参考**过期Key扫描**。

如何查询删除了哪些过期 Key?

暂不支持查询删除的过期Key记录。

8.7 Key 的保存时间是多久？如何设置 Key 的过期时间?

- Key的保存时间是多久?
  - 如果没有设置过期Key，数据会一直存在。
  - 如果设置了过期Key，过期Key的删除机制请参考**过期Key扫描**。
  - 如果已经设置了过期Key，希望移除设定的过期时间，可使用Redis **PERSIST** 命令。
- 如何设置过期Key?

  可使用**expire**或**pexpire**命令设置某个key过期时间，例如执行**expire key1 100**命令后，则key1在100秒后将过期；执行**pexpire key2 1800**后，则key2在1800毫秒后将过期。

  expire是以秒作为key过期时间，**pexpire**是以毫秒作为key过期时间。

8.8 Redis 执行大 Key 分析后内存使用率降低的原因

Redis执行大Key分析，只会查询占用空间过大的Key，并不会删除Key。如果Redis执行大Key分析后内存使用率降低，可能是因为原Redis中存在较多过期Key，因为过期Key的惰性删除机制，Key过期后如果未被访问和识别到，不会立即被删除从而积压，在实例进行大Key分析过程中，会遍历Redis实例中的所有Key，使实例中的过期Key被识别到过期，因而被删除。

过期Key的删除机制，以及如何手动执行过期Key或设置自动过期Key扫描的操作，请参考**过期Key扫描**。

9 Redis 命令

9.1 Redis 命令是否支持审计?

Redis是高性能读写，如果命令支持审计，性能会受到影响。目前仅Redis 4.0及以上版本的Proxy集群实例支持在部分区域支持命令审计日志，参考**查看Redis实例的命令审计日志**。其他实例类型和区域暂不支持命令审计。

9.2 如何清空 Redis 数据?

注意数据清空功能为高危操作，请谨慎执行。

- Redis 3.0实例

  Redis 3.0实例不支持在DCS控制台上执行“数据清空”功能。需要使用Redis-cli客户端连接实例，执行**flushdb**或者**flushall**命令进行清空。

  flushall：清空整个实例的数据。

  flushdb：清空当前DB中的数据。

- Redis 4.0及以上版本的实例

  Redis 4.0及以上版本的实例数据清空，可以使用Redis-cli客户端或通过管理控制台的Web CLI功能连接实例，执行**flushdb**或者**flushall**命令清空数据，或使用DCS控制台上的“更多>数据清空”功能一次全量清空Redis数据。

  如果是集群实例，集群实例默认不支持多DB，由分片组成，如果使用命令清空，需要对集群每个分片都执行**flushdb**或者**flushall**命令，否则容易出现数据清空不彻底的问题。

  说明

  - 目前只有Redis 4.0及以上版本的实例支持在DCS控制台上执行“数据清空”功能及通过管理控制台的Web CLI功能连接Redis实例。
  - 在Web CLI界面使用flushdb命令，一次只会清理一个分片，如果有多个分片，需要用命令行连接到每个分片的主节点上，分别执行flushdb。
  - Web CLI方式不支持清空Cluster集群的数据。

9.3 如何在 Redis 中查找匹配的 Key 和遍历所有 Key?

查找匹配 Key

在大Key和热Key分析中，不支持按照指定格式分析，如果需要查找指定前缀或者后缀格式的Key，您可以使用scan命令，根据指定格式进行匹配查找。

例如，需要查找Redis实例中包含a关键字的Key，可以使用Redis-cli工具，执行以下命令：

./redis-cli -h *{redis_address}* -p *{port}* [-a *password*] --scan --pattern '*a*'

遍历所有 Key

由于keys命令复杂度高，容易导致Redis无响应，所以禁止使用keys命令遍历实例所有的Key。如果需要在Redis实例中遍历所有的Key，可以使用Redis-cli工具，执行以下命令可以遍历Redis实例的所有key。

./redis-cli -h *{redis_address}* -p *{port}* [-a *password*] --scan --pattern '*'

scan命令的使用方法，可以参考**Redis官方网站**。

9.4 Redis 命令执行失败的可能原因

Redis命令执行失败，一般有以下可能原因：

- 命令拼写不正确

  如下图所示，命令拼写有误，Redis实例返回“ERR unknown command”，删除key的正确命令为**del**。

  ```
  192.168.0.244:6379> delete hellokitty
  (error) ERR unknown command 'delete'
  192.168.0.244:6379> del hellokitty
  (integer) 1
  192.168.0.244:6379>
  ```

- 在低版本Redis实例运行高版本命令

  如下图所示，在Redis 3.0版本运行Redis 5.0新增的Stream相关命令，Redis实例返回命令出错信息。

  ```
  192.168.0.244:6379> xadd stream01 * field01 teststring
  (error) ERR unknown command 'xadd'
  192.168.0.244:6379> info server
  # Server
  redis_version:3.0.7.9
  redis_git_sha1:10fba618
  ```

- DCS Redis不支持的部分命令。

  出于安全原因，DCS禁用了部分命令，具体参考**Redis命令的兼容性**，查看禁用命令与受限使用命令。

- 在控制台提供的Web CLI界面执行命令失败。

  Web CLI工具除了同样不支持上述列出的禁用命令与受限命令，对keys命令也有一定的使用限制。

- 执行lua脚本失败。

  例如报错：ERR unknown command 'EVAL' ，说明您的Redis实例属早期创建的低版本Redis实例，不支持lua脚本，这种情况请提工单联系技术支持，升级您的Redis实例。

- 执行setname和getname失败。

  说明您的Redis实例属早期创建的低版本Redis实例，不支持这两个命令，这种情况请提工单联系技术支持，升级您的Redis实例。

- 2018年7月10日前创建的Redis**集群**实例，以下命令被禁用，客户端执行时也会收到命令出错信息。如果需要支持，请提工单联系技术支持，升级集群实例。

  SINTER、SDIFF、SUNION、PFCOUNT、PFMERGE、SINTERSTORE、SUNIONSTORE、SDIFFSTORE、SMOVE、BLPOP、BRPOP、BRPOPLPUSH、ZUNIONSTORE,、ZINTERSTORE、EVAL、EVALSHA、BITOP、RENAME、RENAMENX、RPOPLPUSH、MSETNX、SCRIPT LOAD、SCRIPT KILL、SCRIPT EXISTS、SCRIPT FLUSH。

9.5 在 Web Cli 执行 keys 命令报错“permission denied”

Web Cli已禁用**keys**命令，请使用**Redis-cli**执行。

9.6 高危命令如何重命名?

当前支持重命名的高危命令为command、keys、flushdb、flushall、hgetall、scan、hscan、sscan、和zscan，Proxy集群实例还支持dbsize和dbstats命令重命名，其他命令暂不支持重命名。高危命令重命名的操作方式，请参考**命令重命名**。

说明

- 目前Redis不支持直接禁用命令，涉及到以上高危命令，可以使用命令重命名。关于DCS实例支持和禁用的命令请参考**开源命令兼容性**。
- 提交命令重命名操作后，系统会自动重启该实例，重命名操作完成后立即生效。因为涉及安全性，页面不支持查询重命名后的命令。
- 同一个命令支持多次重命名，每次新的重命名操作会覆盖之前的重命名命令。
- Redis 4.0及以上版本的实例支持命令重命名，Redis 3.0不支持。

9.7 是否支持 pipeline 命令?

支持。

注意： Redis Cluster集群实例使用pipeline时，要确保管道中的命令都能在同一分片执行。

9.8 Redis 是否支持 INCR/EXPIRE 等命令?

支持。

命令兼容性相关说明请参考产品简介的“**命令兼容性说明**”章节。

9.9 Redis 命令执行不生效

如果客户端代码业务异常，怀疑是Redis命令不生效，可以通过**Redis-cli**执行命令和查看数据，判断Redis命令执行是否异常。

以下列举两个场景：

- 场景一：通过设置key值和查看key值，即可判断该命令是否生效。

  Redis通过set命令写String类型数据，但是数据未变化，则可以使用Redis-cli命令访问Redis实例，执行如下命令：

  ```
  192.168.2.2:6379> set key_name key_value
  OK
  192.168.2.2:6379> get key_name
  "key_value"
  192.168.2.2:6379>
  ```

- 场景二：通过expire命令设置过期事件，但是怀疑过期时间不对，则可以执行如下操作:

  设置10秒过期时间，然后执行ttl命令查看过期时间，如下图表示，执行ttl命令时，过期时间剩下7秒。

  ```
  192.168.2.2:6379> expire key_name 10
  (integer) 1
  192.168.2.2:6379> ttl key_name
  (integer) 7
  192.168.2.2:6379>
  ```

说明

Redis客户端和服务端通过二进制协议进行通信，使用Redis-cli、Jedis、Python客户端并没有差异。

因此如果怀疑Redis有问题，但是使用Redis-cli排查没问题，那就很可能是业务代码存在问题，如果日志没有明显错误信息，则建议在代码添加日志支撑进一步分析。

9.10 Redis 命令执行是否有超时时间? 超时了会出现什么结果?

Redis超时分为客户端超时和服务端超时。

- 客户端命令超时时间一般由客户端代码自行控制，业务侧需要根据自己的业务特点选择合适的超时时间（例如Java的Lettuce客户端，该参数名为timeout）。

  客户端如果发生命令执行超时，根据不同客户端的逻辑控制，可能会发生超时报错、命令堵塞、客户端连接重试等情况。

- Redis服务端Timeout默认配置为0，不会主动断开连接，如果需要修改配置，可以参考**修改实例配置参数**。

如果实例配置了该Timeout参数值（不为0），当客户端与服务端空闲连接超过该参数值时，连接会断开。

9.11 Redis 的 Key 是否能设置为大小写不敏感?

DCS Redis和开源Redis保持一致，key对大小写敏感，且不支持设置大小写不敏感功能。

9.12 WebCli 的常见报错

1. ERR Wrong number of arguments for 'xxx' command

   该报错代表执行的Redis命令存在参数错误（语法错误），可以参考开源Redis命令协议介绍进行命令构造。

2. ERR unknown command 'xxx'

   该报错代表此命令为未知命令或者非redis协议定义的合法命令，可以参考开源Redis命令协议介绍进行命令构造。

3. ERR Unsupported command: 'xxx'

   该报错代表命令在DCS的Redis实例场景下禁用，可以参考**支持和禁用的Web CLI命令**。

10 监控告警

10.1 Redis 实例 CPU 使用率达到 100%的原因

问题现象

Redis实例CPU使用率短时间内冲高。CPU过高可能会导致连接超时，影响业务。CPU过高也可能触发主备倒换。

可能原因

1. 客户的业务负载过重，qps过高，导致CPU被用满，排查方法请参考**排查QPS是否过高**。
2. 使用了keys等消耗资源的命令，排查及处理措施请参考**查找并禁用高消耗命令**。
3. 发生Redis的持久化重写操作，排查及处理措施请参考**是否存在Redis的持久化重写操作**。

排查 QPS 是否过高

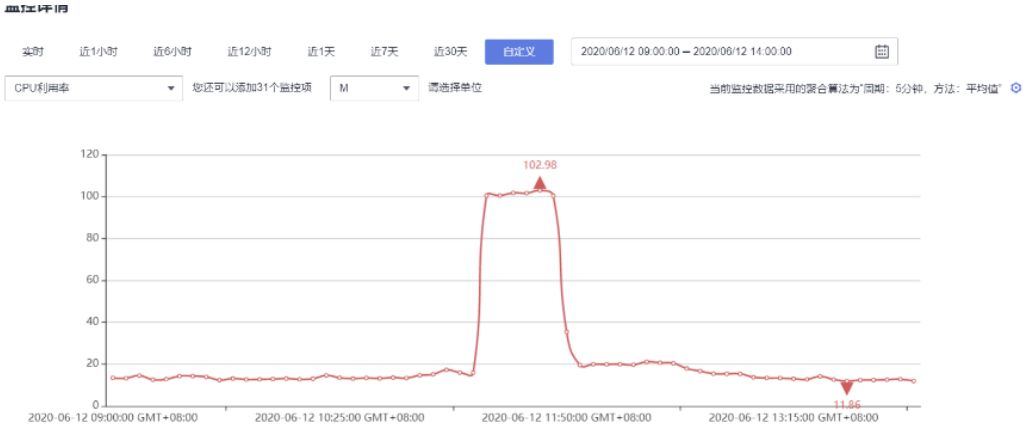
在分布式缓存服务控制台的缓存管理页面，单击实例进入实例详情界面，单击左侧的性能监控，进入性能监控页面，查询实例级别的每秒并发操作数（ QPS ）。

如果QPS过高，建议优化客户业务或者**变更实例规格**。不同实例规格支持的QPS请参考**实例规格**。

查找并禁用高消耗命令

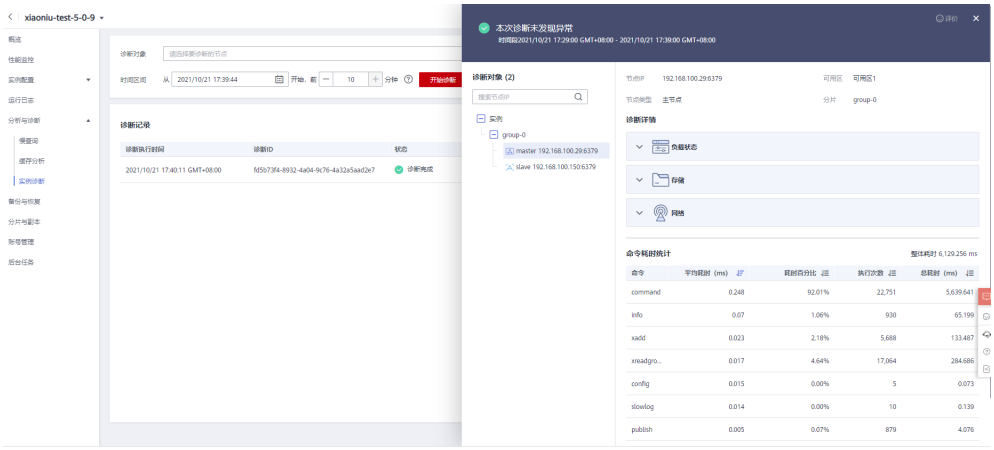
使用了keys等消耗资源的命令，高消耗资源的命令即时间复杂度为O(N)或更高的命令，通常情况下，命令时间复杂度越高，在执行时消耗的资源越高，这会导致CPU使用率超高，容易触发主备倒换。关于各命令对应的时间复杂度信息请参见**Redis官网**。例如，使用了keys等消耗资源的命令，导致CPU超高，建议客户改成scan命令或者禁用keys命令。

步骤1 通过性能监控功能，确认CPU使用率高的具体时间段。



步骤2 通过下述方法，找出高消耗的命令。

- 慢查询功能会记录执行超过指定时间阈值的命令，通过分析慢查询的语句和执行时长可帮助您找出高消耗命令，具体操参见**慢查询**。
- 通过实例诊断功能，选择CPU冲高的时间点进行诊断后，可以看到报告中的对应时间段命令的执行情况以及CPU耗时百分比，具体操作参见**实例诊断**。



步骤3 处理措施。

- 评估并禁用高风险命令和高消耗命令，例如**FLUSHALL**、**KEYS**、**HGETALL**等。
- 优化业务，例如避免频繁执行数据排序操作。
- **可选：**根据业务情况，选择下述方法对实例进行调整：
  - 调整实例为读写分离实例，对高消耗命令或应用进行分流。
  - 扩容实例增强实例处理能力。

----结束

是否存在 Redis 的持久化重写操作

除单机及单副本Cluster集群实例外，华为云其他Redis实例默认开启AOF数据落盘，实例开启了AOF持久化功能后，会定期进行AofRewrite的磁盘整理，AOF磁盘持久化整理一般在以下2种场景执行：

- 数据量写入不大，AOF文件不大时，固定在每天的凌晨1-4点进行AOF持久化重写。所以容易出现这个时间点实例CPU使用率超高的现象。

- 数据量写入过大，AOF文件大小超过阈值（缓存实例容量的3-5倍）时，不论当前的所处的时间，会自动触发后台AOF持久化重写。

Redis的持久化重写操作（Bgsave或Bgrewriteaof）比较消耗CPU资源（请参考**为什么使用Fork执行Bgsave和Bgrewriteaof**），Bgsave和Bgrewriteaof会调用系统的Fork机制，造成CPU短暂时间冲高。

如果客户没有需要用到持久化功能，建议将该功能关闭（请根据实际业务慎重操作，关闭持久化功能会导致极端故障场景下恢复时，由于没有落盘造成的数据丢失）。关闭操作：在实例详情页面，选择“配置参数”页签，将“appendonly”修改为“no”。

10.2 如何查看 Redis 实例的实时并发连接数和最大连接数

查看 Redis 实例实时并发连接数

当您需要查看DCS实例收到的实时连接数时，在控制台缓存管理页面，单击需要查看的实例右侧的“查看监控”，进入云监控页面。

进入监控页面后，找到“**活跃的客户端数量**”监控项。您可以单击该监控项的右上角的查看按钮  ，使用大图模式查看。

在弹出的“活跃的客户端数量”页面，根据需要选择查看的时间段，例如，需要查看10分钟内的连接数，您可以将时间自定义为10分钟。由于监控数据采集的是周期内增加的连接数，您可以通过监控图表，查看这个时间段的连接数的走势，并统计10分钟内的连接总数。

说明

监控指标“活跃的客户端数量”用于统计已连接的客户端数量，包括系统监控、配置同步和业务相关的连接数，不包括来自从节点的连接。

您还可以通过控制台**会话管理**，查看连接实例的客户端会话信息。

查看或修改实例最大连接数

您可以在控制台实例创建页面或通过**实例规格**查看实例默认及最大可配的最大连接数。

创建实例后，您可以通过DCS控制台实例详情页“实例配置>参数配置”页面，查看或修改**maxclients**参数值，即最大连接数。（读写分离实例暂不支持该参数）

如果连接数达到上限，超出的请求会被拒绝，连接超时。

10.3 Redis 监控数据异常处理方法

当对Redis监控数据存在疑问或异议时，可以使用Redis-cli访问Redis实例，执行**info all**命令，查看进程记录的指标。info all输出详解可参考：**https://redis.io/docs/latest/commands/info/**。

10.4 监控数据出现实例已使用内存略大于实例可使用内存是什么原因?

DCS单机和主备实例已使用内存为redis-server进程统计的已使用内存。集群是基于分片机制实现的，集群的已使用内存为各个分片redis-server的已使用内存的总和。

由于开源redis-server内部机制的原因，有时会出现DCS缓存实例已使用内存略大于可使用内存的情况，此为正常现象。

Redis通过zmalloc来分配内存，不会在每一次分配内存时都检查是否会超过max_memory，而是在周期任务以及命令处理的开头处等地方，判断一次当前的used_memory是否超过max_memory，如果超过就触发逐出操作。所以，对于max_memory策略的限制实施并不是实时、刚性的，会出现某个时间used_memory大于max_memory的情形。

10.5 为什么带宽使用率指标会超过 100%

带宽使用率基本信息如下:

指标ID	指标名称	含义	取值范围	测量对象&维度	监控周期（原始指标）
bandwidth_usage	带宽使用率	当前流量带宽与最大带宽限制的百分比	0-200%	测量对象： Redis 4.0及以上版本 主备、读写分离、集群实例数据节点 测量维度： dcs_cluster_node	1分钟

其中，带宽使用率的计算公式为：带宽使用率=（网络瞬时输入流量+网络瞬时输出流量）/（2*最大带宽限制）* 100%。

该公式中同时计算了网络瞬时输入流量和网络瞬时输出流量，这两个指标值是有统计主从同步的流量的。所以统计的总流量使用量会比正常的业务流量大一些，会发生带宽使用率指标超过100%的情况。

判断当前是否被限流，请使用**流控次数**这个指标，这个指标值大于0时，表示当前已使用的带宽超过最大带宽限制，产生流控。

限流时，流控次数指标是不统计主从同步流量的，所以有时候会出现带宽使用率指标超过100%，但流控次数为0的情况。

10.6 监控指标中存在已拒绝的连接数是什么原因?

当监控指标中出现“已拒绝的连接数”时，请确认客户端连接数是否已经超过实例的最大连接数限制。

说明

Redis 4.0/5.0/6.0版本的实例，仅在主备、集群和读写分离实例的数据节点中支持查看“已拒绝的连接数”。

- 查看最大连接数：单击实例名称，进入实例详情页面，选择“配置参数”页签，查看maxclients参数的值（读写分离实例暂不支持该参数，可通过**DCS实例规格**查询实例最大连接数）。
- 查看实际连接数：单击实例名称，进入实例详情页面，选择“性能监控”页签，找到“活跃的客户端数量”监控项查看。

如果客户端连接数已到达连接上限，可以根据需要调整maxclients参数，如果maxclients参数已经是最大可配连接数，仍不满足需求，则需要升级规格。

10.7 触发限流（流控）的原因和处理建议

Redis产生流控，说明redis在周期内的使用流量超过该实例规格的最大带宽。流控会导致连接被丢弃，从业务角度可能会造成业务的延迟增大，客户端连接异常等问题。

说明

实例规格对应的最大带宽，可以查看**实例规格**中对应实例类型的“基准/最大带宽”。

带宽使用率不高时，也有可能有限流，因为带宽使用率是上报周期实时值，一个上报周期检查一次。而流控检查是秒级的，有可能存在上报周期间隔期间，流量有秒级冲高，然后回落，待上报带宽使用率指标时已恢复正常。

对于主备实例：

- 如果实例一直有流控但是带宽使用率不高，这说明可能存在业务微突发问题，或者大Key热Key问题，建议对实例进行自动诊断分析，优先排除大Key热Key问题。
- 如果带宽使用率居高不下，说明带宽可能存在超限风险，需要扩容处理。

对于集群实例：

- 仅有单个或少量几个分片出现流控，则多数为该分片存在大Key热Key问题。
- 所有或大多数分片同时出现流控或者带宽使用率高的问题，这说明实例的带宽达到了瓶颈，建议扩容实例。

说明

- DCS控制台提供了大Key和热Key的分析功能，请参考**分析Redis实例大Key和热Key**减少大key和热key。
- 如果用户执行了keys等消耗资源的命令，也可能会导致CPU和带宽使用率增加，从而出现流控。

11 主备倒换

11.1 发生主备倒换的原因有哪些?

主备倒换有以下几种可能的场景：

- 用户自行从DCS控制台界面发起“主备倒换”操作，切换主实例。
- DCS检测到主备实例的主节点存在故障后，触发实例“主备倒换”操作。
  例如，使用了keys等消耗资源的命令、日志老化批量删除日志，导致CPU超高，都会触发主备倒换。
- 用户在DCS界面上执行重启操作，可能触发备节点升主节点，即主备倒换。
- Redis实例在扩容过程中，可能会发生主备倒换。
  扩容过程中，实例会创建新规格的节点作为备节点，主节点数据全量+增量同步到备节点后进行主备切换并删除原节点，完成扩容。

如果您需要对实例主备倒换进行监控，可以在云监控服务中创建事件监控，具体操作请参考**创建事件监控的告警通知**。创建事件监控告警后，如果发生主备倒换，系统会上报主备倒换事件，收到该事件通知后，请查看客户端业务是否存在异常。如果业务不正常，则需要确认客户端连接是否正常，是否支持在主备倒换后进行客户端重连恢复业务，如果不支持客户端重连，则需要重启客户端。

11.2 主备倒换的业务影响

DCS主备、读写分离、或者集群实例发生异常时，会触发内部主备倒换，并自动恢复，在异常检测和恢复期间，可能会影响业务，时间在半分钟内。

11.3 主备实例发生主备倒换后是否需要客户端切换 IP?

不需要。当主备倒换或主节点故障后，IP地址会自动绑定到正常的备节点，绑定后，原备节点升级为主节点。

11.4 Redis 主备节点的数据如何同步?

一般情况下，Redis主节点数据更新后会自动复制到关联的备节点。但由于Redis异步复制的技术，特殊情况下，备节点更新可能会落后于主节点。例如，当主节点的I/O写入

速度超过了备节点的同步速度，或者因异常原因导致主节点和备节点数据同步网络延迟，使得备节点与主节点存在滞后或者部分数据不一致，若此时进行主备切换，未及时完成同步的少量数据可能会丢失。

12 创建实例和权限

12.1 Redis 实例创建失败的可能原因

- 子网IP不足

  分析：单机实例需要绑定1个子网IP地址，主备实例需要绑定2个子网IP地址，集群实例有多个节点，需要绑定多个IP地址。

  解决方案：如果所选子网IP资源不足，可以更换子网创建实例或者释放当前子网下其他IP地址。

- IAM用户（子用户）没有创建权限

  分析：用户所属组需要拥有“DCS FullAccess”策略或“DCS Administrator”角色或者拥有创建DCS实例的权限。

  解决方案：使用管理员用户创建DCS实例。

12.2 创建 DCS 实例时页面无法自动获取子网和安全组等信息

创建DCS缓存实例时，如果无法查看虚拟私有云、子网、安全组、弹性IP，可能原因是该用户无Server Administrator和VPC Administrator权限，增加权限的详细步骤请参考**如何修改用户权限**。

12.3 创建 DCS 时选择不到需要的企业项目

现象

创建DCS时，选择不到需要的企业项目。

原因

选择不到需要的企业项目的原因是企业项目下没有添加DCS的权限。

解决方案

1. 登录分布式缓存服务控制台。

2. 单击“企业 > 项目管理”中，单击企业项目名称后操作列的“查看资源”进入该企业项目的详细信息页面。
3. 单击“权限管理 > 用户组授权”进入选择用户页面。

   说明

   选择“用户组授权”，给用户所在的用户组统一添加授权，如果选择“用户授权”，也可以给单用户添加授权。
4. 单击需要赋权的用户/用户组后的“授权”进入授权页面。
5. 查询并选中“DCS FullAccess”策略名称，单击“下一步”添加DCS的项目权限后，单击“确定”，完成DCS项目权限的添加。

   如需了解更多DCS权限策略，请参考**权限策略**。

   说明

   如果您同时配置了DCS UserAccess和DCS FullAccess这两个系统策略，由于DCS UserAccess策略存在Deny，根据Deny优先原则，您无法执行实例创建、修改、删除、扩容和缩容操作。所以，如需使用DCS FullAccess策略，需要删除DCS UserAccess策略。

12.4 IAM 子用户无法看到新买的 Redis

问题现象

子账户看不到新创建的Redis资源。

问题原因

新购Redis所在的企业项目没给这个iam子用户添加权限。

解决方案

1. 登录分布式缓存服务控制台。
2. 单击“企业 > 项目管理”中，单击企业项目名称后操作列的“查看资源”进入该企业项目的详细信息页面。
3. 单击“权限管理 > 用户组 > 添加授权”进入选择用户组页面。
4. 选择需要赋权的用户组，单击“下一步”进入设置策略页面。
5. 选中“DCS UserAccess”策略名称，单击“确定”添加DCS的权限完成。

13 Memcached 使用

13.1 Memcached 实例的数据能否 dump 出来分析?

不支持将数据导出分析。

13.2 DCS 的 Memcached 兼容的版本号是多少?

DCS的Memcached是基于Redis 3.0版本引擎实现的，兼容memcache1.5.1版本。

13.3 DCS 的 Memcached 支持哪些数据结构?

DCS的Memcached目前仅支持Key-Value的数据结构，暂不支持List等数据结构。

13.4 Memcached 实例支持公网访问么?

Memcached实例暂不支持公网访问。

未开启公网访问的DCS缓存实例，本地环境不能直接连接DCS缓存实例。DCS采用虚拟私有云（VPC）管理各服务的网络安全，用户创建的DCS缓存实例，必须通过与DCS缓存实例相同虚拟私有云（VPC）的弹性云服务器（ECS）来访问。

如果您在应用开发调试阶段，可以通过网络代理转发的方式，用一台能与DCS缓存实例网络互通的弹性云服务器（ECS）做中转，实现本地环境连接DCS缓存实例。具体操作参考**使用SSH隧道代理机制实现公网访问DCS实例**。

13.5 Memcached 实例是否支持修改配置参数?

处于“运行中”状态的Memcached实例支持修改配置参数。

具体修改操作，请参考**配置运行参数**。

13.6 DCS 的 Memcached 与自建 Memcached 的区别是什么?

DCS的Memcached与本地自建Memcached的区别如下表13-1所示。

表 13-1 DCS Memcached 与自建 Memcached 的区别

比较项	DCS Memcached	自建Memcached
部署	DCS的Memcached简单易用，创建后缓存服务即时可用，实现快速部署，无需关心硬件及软件。	自己搭建Memcached的操作和设置较复杂。
可用性	DCS的Memcached主备实例支持双机热备，能够持续提供稳定的服务，在主节点故障时，备节点秒级自动升级为主节点，避免单点故障。	自建的Memcached服务需要自行实现高可用。
安全	DCS的Memcached采用公有云的VPC和安全组进行网络访问安全控制。	自建的Memcached服务需要用户自行设计并实现安全机制。
扩容	DCS的Memcached支持在线扩容，在控制台简单操作，即可完成扩容。	自建的Memcached服务的扩容稍复杂，需要自己添加硬件配置，重启服务。

13.7 DCS 的 Memcached 过期数据清除策略是什么?

DCS的Memcached作为缓存产品是允许用户根据业务需求设置在其中存放数据的过期时间的。例如在执行**add**操作的时候可以设置**expire**过期时间。

DCS的Memcached默认策略为不逐出（noeviction）。支持通过修改Memcached实例配置参数（maxmemory-policy）修改实例的数据逐出策略。

Memcached实例支持的数据逐出策略和配置参数的方式，请参考**修改DCS实例配置参数**。

13.8 创建 Memcached 实例时如何选择可用区?

简单来说，只要是在同一区域（Region）内，选择任意一个可用区内DCS的Memcached都没有功能上的本质区别。

一般来讲，同一可用区比跨可用区的网络时延更有优势，但是跨可用区从容灾的角度比同可用区更有优势。当应用内部需要更低的网络时延，可以将应用实例组部署在同一个可用区中。

DCS的Memcached目前已支持跨可用区部署，在管理控制台创建DCS的Memcached时进行配置。只要与您的ECS在同一个区域（Region）内，无论选择创建哪个可用区的Memcached都可以实现与ECS正常的连通使用。若您希望获得更低的网络时延，请根据您ECS的可用区选择创建对应可用区的Memcached。

例如，您有一台**中国-香港**的ECS，其属于**中国-香港**可用区B，那么当您在创建DCS的Memcached时，选择**中国-香港**任何一个可用区的实例都是可以正常使用的。您选择**中国-香港**可用区B的实例可以获得与这台ECS更低的网络时延。

注意：由于资源库存因素，可能出现您在创建DCS的Memcached时仅有一个可用区资源供选择的情况，这不会影响您的正常使用。