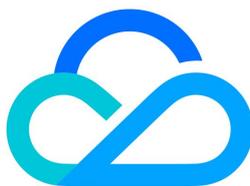


云数据库 Redis

性能排查与调优



腾讯云

【 版权声明 】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

性能排查与调优

无法连接 Redis 实例

CPU 使用率过高

出流量过高

内存使用率过高

总请求数过大

连接使用率过高

执行时延过高

执行错误

性能排查与调优

无法连接 Redis 实例

最近更新时间：2023-04-11 15:57:19

现象描述

- 现象1：使用云服务器 CVM 通过自动分配给云数据库的内网地址连接云数据库 Redis，连接失败。具体连接方式，请参见 [连接 Redis 实例](#)。
- 现象2：登录 [Redis 控制台](#)，在实例列表，在目标实例的操作列，单击登录，跳转至数据库管理 DMC 平台，连接云数据库 Redis，连接失败，如下图所示。



可能原因

- 首次连接数据库失败时，可能原因如下：
 - 端口问题。
 - 网络配置问题或安全组配置错误。
 - 密码问题。
- 实例运行过程中突发连接失败，可能原因如下：
 - 连接数已满。
 - 内存写满或者分片写满。
 - 发生 HA 切换、服务不可用、只读副本切换、只读副本服务不可用等。
- 客户端问题，可能原因如下：
 - 连接池参数设置不合理。
 - 连接泄露。

处理步骤

步骤1：使用 telnet 确认 Redis 端口是否可正常访问

大部分客户遇到的连接失败、无法连接等问题，可以通过命令行工具以及 `telnet` 缩小问题范围。

```
[root@VM-4-10-centos ~]# telnet 10.x.x.34 6379
Trying 10.x.x.34...
Connected to 10.x.x.34.
Escape character is '^]'.
```

如上述所示，提示连接成功代表 Redis 实例端口正常可访问。如果出现异常，请继续 [步骤2](#)，排查网络问题。

步骤2：排查是否为网络配置问题

云服务器和数据库，务必在同一账号同一个 VPC 内，或同在基础网络内，才能内网直接互通。如下情况，都可能导致连接失败。

- 云服务器（CVM）采用私有网络（VPC），Redis 采用基础网络。建议将 Redis 从基础网络切换为 VPC 网络，请参见 [配置网络](#)。
- CVM 采用基础网络，Redis 采用 VPC。建议将 CVM 从基础网络切换为 VPC 网络，请参见 [切换私有网络服务](#)。
- CVM 与 Redis 在同一地域内，但属于不同的 VPC 网络。建议将 Redis 迁移到 CVM 所在的 VPC 网络，请参见 [配置网络](#)。
- CVM 与 Redis 不在同一地域内，属于不同的 VPC 网络。建议在两个 VPC 网络之间建立 [云联网](#)。
- CVM 与 Redis 账号不同，属于不同的 VPC 网络。建议在两个 VPC 网络之间建立 [云联网](#)。

步骤3：确认是否为安全组问题

若 CVM 和 Redis 的安全组配置有误，则会导致 CVM 无法连接 Redis。

CVM 安全组配置有误

若想使用 CVM 连接 Redis，需在 CVM 安全组中配置出站规则，当出站规则的目标配置不为 0.0.0.0/0 且协议端口不为 ALL 时，需要把 Redis 的 IP 及端口添加到出站规则中。

1. 登录 [安全组控制台](#)，在安全组列表中，找到 CVM 所绑定的安全组，单击安全组名，进入 CVM 绑定的安全组详情页。
2. 选择出站规则页签，单击添加规则。
 - 类型选择自定义。
 - 目标填写您 Redis 的 IP 地址（段）。
 - 协议端口填写 Redis 内网端口
 - 策略选择允许。

Redis 安全组配置有误

若想指定的 CVM 连接 Redis 实例，需要在 Redis 安全组中配置入站规则，当入站规则的源端配置不为 0.0.0.0/0 且协议端口不为 ALL 时，需要把 CVM 的 IP 及端口添加到入站规则中。

1. 登录 [安全组控制台](#)，在安全组列表中，找到 Redis 实例所绑定的安全组，单击安全组名，进入 Redis 绑定的安全组详情页。
2. 选择入站规则页签，单击添加规则。

填写您允许连接的 IP 地址（段）及需要放通的端口信息（Redis 内网端口），选择允许放通。

 - 类型选择自定义。
 - 来源填写您 CVM 的 IP 地址（段）。
 - 协议端口填写 Redis 内网端口。
 - 策略选择允许。

⚠ 注意：

- Redis 内网默认端口为 6379，同时支持自定义端口，若修改过默认端口号，安全组中需放通 Redis 新端口信息。
- 安全组入站规则需要放通 Redis 实例的 6379 端口。

步骤4：确认是否为密码问题

执行 `info` 命令进行测试，如果执行提示如下，说明 Redis 密码没有问题。

```
[root@SNG-Qcloud /data/home/rickyu]# redis-cli -h 10.x.x.34 -p 6379 -a password
10.x.x.2:6379> info cpu
# CPU
used_cpu_sys:1623.176000
used_cpu_user:4649.572000
used_cpu_sys_children:0.000000
```

```
used_cpu_user_children:0.000000
```

如果执行提示 NOAUTH Authentication required. 代表密码错误。

```
10.x.x.31:6379> info memory
NOAUTH Authentication required.
10.x.x.31:6379>
```

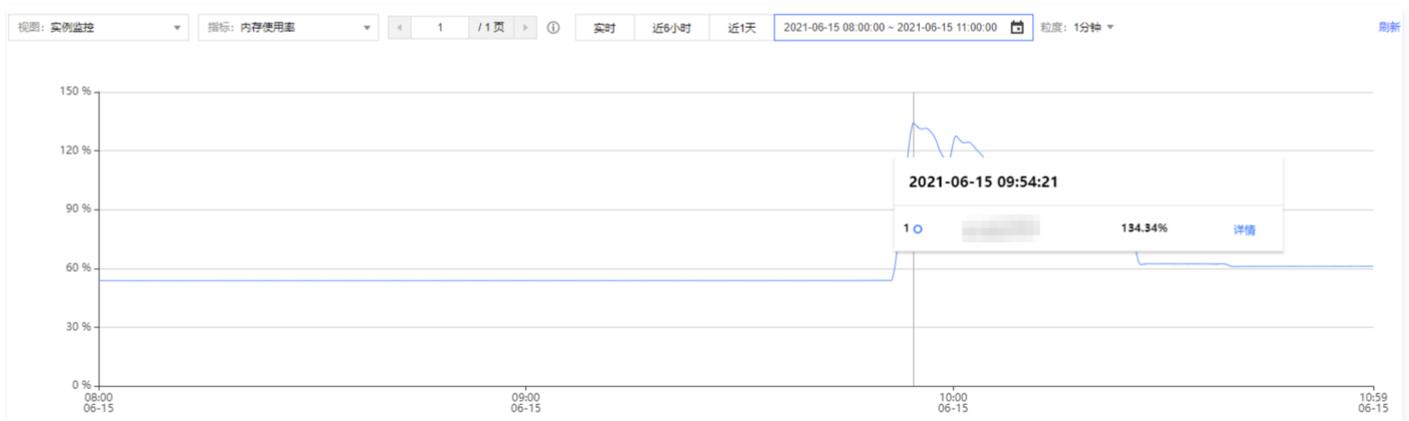
登录 [Redis 控制台](#)，单击实例 ID 进入详情页，进行密码重置即可，具体操作，请参见 [管理账号](#)。

步骤5：确认是否内存写满或分片写满导致连接失败

如果业务提示如下错误信息。

```
"-READONLY You can't write against a read only slave.\r\n"
```

登录 [Redis 控制台](#)，在实例列表找到目标实例，单击实例 ID 进入系统监控页面，指标下拉列表选择内存使用率，查看实例内存写满情况。



内存写满情况下，写入失败，需进行如下操作。

- 立即进行扩容。具体操作，请参见 [变更实例规格](#)。
- 修改数据库驱逐策略。在参数配置中设置参数 `maxmemory-policy` 调整为 `allkeys-lru` 或者 `volatile-lru`。具体操作，请参见 [管理实例参数](#)。

⚠ 注意：

驱逐策略调整为 `allkeys-lru`，可能损坏业务数据，请根据实际需求评估。

步骤6：确认是否连接数配额不足

监控指标连接使用率指客户端连接到实例的 TCP 连接数量与实例最大连接数的占比，该指标持续偏高，说明当前数据库连接数配额不足，需调整最大连接数。

问题现象

错误提示信息如下所示：

```
ERR max number of clients reached
```

解决方法

1. 登录 [Redis 控制台](#) 在，右侧实例列表页面上方，选择地域。在实例列表中，到目标实例。单击蓝色字体的实例 ID，进入实例详情页面，单击 [系统监控](#) 页签，再选择监控指标页签，查看监控数据。在视图下拉列表选择 [实例监控](#)，指标选择 [连接使用率](#)，查看其监控视图是否持续偏高。
2. 连接使用率持续偏高，调整最大连接数，修改连接数配置规格。具体操作，请参见 [调整连接数数量](#)。

步骤7：确认是否发生 HA 切换、服务不可用、只读副本切换、只读副本服务不可用等

如果在某个确定的时间点发现连接异常或者有大量的访问报错及慢查询，同时接收到腾讯云可观测平台事件告警，发生了异常事件，请联系 [在线客服](#) 获取帮助。

腾讯云可观测平台事件告警配置方法，请参见 [创建事件规则](#)。

步骤8：如果使用Jedis连接池，确认客户端连接池配置是否合理

问题现象

如果连接池可用连接数量耗尽，旧连接未及时释放，新创建连接会失败，客户端提示如下错误信息。

```
JedisConnectionException: Could not get a resource from the pool
```

解决方法

1. 在客户端使用如下指令，确认当前访问实例6379端口的连接数量，如果客户端连接数接近连接池配置的 maxTotal 值，将会出现连接失败的问题。

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

2. 参考 [Java 连接示例](#)，检查是否调用 jedis.close() 进行旧连接的释放，避免连接泄露。

3. 如果旧连接都已释放，业务并发量大的情况下，需增加 maxTotal 参数值。

说明：

每个客户端连接池 maxTotal 值 * 客户端数量 = 云数据库 Redis 的最大连接数。

更多参考

网络类型/ VPC 判断方法

使用内网地址连接云数据库时，CVM 和 Redis 须是同一账号，且同一个 VPC 内，或同在基础网络。

说明：

- 如果实例列表的网络处，均显示为基础网络或均显示为 VPC，则表示 CVM 和 Redis 是同一网络类型。
- 如果实例列表的网络处，均显示为同一个 VPC（保障同一个地域），则表示 CVM 和 Redis 是同一 VPC。

查看 CVM 网络类型

登录 [CVM 控制台](#)，在实例列表查看网络。

ID/名称	监控	状态	可用区	实例类型	实例配置	主IPv4地址	主IPv6地址	实例计费模式	网络计费模式	操作
搜索 '...' 找到 1 条结果										
...		运行中	广州七区	标准型S6	4核 8GB 5Mbps 系统盘：通用型SSD云 硬盘	...		按量计费 2022-10-21 15:11:31创建	按流量计费	登录 更多
网络: Default-VPC										

查看 Redis 网络类型

登录 [Redis 控制台](#)，在实例列表查看网络。

实例 ID / 名称	监控 / 状态 / 任务	所属项目	可用区	网络	计费模式	架构版本	产品版本	已使用 / 总容量	创建时间	标签	操作
搜索 [] , 找到 1 条结果 返回原列表											
crs-	 运行中	默认项目	广州六区 M	Default-VPC Default-Subnet	包年包月 2023-02-28 11:33:48	Redis 5.0标准架构	内存版	35.12MB/256MB (13.72%)	2022-08-29 11:33:48		登录 配置变更 更多

通过开启外网地址实现外网访问

云数据库 Redis 现已支持在控制台手动开启外网地址，实现外网访问 Redis 实例。具体操作，请参见 [配置外网地址](#)。

CPU 使用率过高

最近更新时间：2023-05-22 10:34:21

CPU 使用率升高会影响整个实例集群的吞吐量，甚至可能会导致应用阻塞，超时中断。当平均 CPU 使用率高于60%或者 CPU 平均峰值使用率高于90%且持续时间超过5分钟时，您需要及时排查具体原因，针对性快速解决问题，以保障业务稳定性和可用性。

现象描述

- 现象1: 收到 CPU 使用率过高的消息提醒:

```
【腾讯云】尊敬的用户，您好！您账号（账号ID：100
XXXXXXXXXX）的云监控告警已恢复。
查看告警详情：https://tcm.qq.com
告警内容：云数据库-Redis-内存版(5秒粒度)-Redis节点 | 平均CPU使用率 >85%
告警对象：crs-XXXXXXXXXX
当前数据：78.183%(平均CPU使用率)
项目|地域：默认项目|广州
告警策略：Redis-cache节点监控
触发时间：2021-07-20 13:57:00 (UTC+08:00)
恢复时间：2021-07-20 13:58:00 (UTC+08:00)
```

- 现象2: CPU 使用率的监控指标高。
- 现象3: 整体吞吐量变小、响应速度变慢。

排查与解决

序号	可能原因	原因分析	排查方法	解决方法
1	频繁建立短连接	实例大量资源消耗在处理频繁短连接上，引起 CPU 使用率较高，连接数较高，然而 QPS（集群每秒访问次数）未达到预期。	借助数据库智能管家 DBbrain 的 诊断优化 功能，分析数据库实例的实时会话统计视图及数据，确认是否存在连接数突增的现象。具体查询方式，请参见 实时会话 。	将短连接调整为长连接，例如使用 JedisPool 连接池连接。具体代码示例，请参见 Jedis 客户端 。
2	存在高时间复杂度的命令。例如：sort、sunion、zunionstore 等。	Redis 是单线程执行命令，执行复杂度高的命令，很可能阻塞其他命令。命令的时间复杂度越高，在执行时会消耗较多的资源，会引起 CPU 使用率上升。	执行高复杂度的命令，通常比较耗时，会相应产生慢日志。可借助数据库智能管家 DBbrain 的 诊断优化 功能进行 慢日志分析 ，在慢日志信息列表中，查看是否存在高复杂度的命令。具体操作，请参见 慢日志分析 。	使用复杂度较高命令，每次不要获取太多的数据，尽量操作少量的数据，让 Redis 可以及时处理返回。
3	读写负载过高，如存在对热点 Key 的大量访问或者存在大 Key。	热 Key 指的是那些在一段时间内访问频率特别高的键值。具体到业务场景，包括热点新闻、热门直播、秒杀活动等等。大量访问流量集中到一个实例中，达到单个实例的处理上限，引起 CPU 使用率上升。	可借助数据库智能管家 DBbrain 的 诊断优化 功能进行 热 Key 分析 ，快速发现访问频次高的热 Key。具体操作，请参见 热 Key 分析 。	热 Key : 拆分复杂数据结构，将热点 Key 拆分为若干个新的 Key 分布到不同 Redis 节点上，从而减轻压力。以哈希类型为例，该热 Key 的类型是一个二级数据结构，该哈希元素个数可能较多，可以考虑将当前 hash 进行拆分。
4		大 Key 指某个 Key 对应的 value 很大，占用	可借助数据库智能管家 DBbrain 的 诊断优化 功能进行 大 Key 排查 ，	大 Key : 如果是 value 过大，您可以将对象拆分成多个 key-

		<p>的 Redis 空间很大。执行大 Key 相关读取或者删除操作时，会严重占用带宽和 CPU。</p>	<p>针对数据库大 key 占用内存的情况进行监控分析。具体信息，请参见 内存分析。</p>	<p>value，将操作压力平摊到多个 Redis 实例；如果是 Key 过多，您可以参考 hash 结构存储，将多个 Key 存储在一个 hash 结构中。</p>
5		<p>读负载过高，达到资源上限。</p>	<p>借助数据库智能管家 DBbrain 的诊断优化功能，查看数据库实例性能趋势视图，分析 QPS、读写请求指标，确认是否由于读负载或者写负载过大导致 CPU 使用率偏高。具体信息，请参见 性能趋势。</p>	<p>读负载过高：通过 增加副本数量 来分摊读负载。开启副本只读，把当前实例的读请求转移在副本只读节点上，实现读取能力的弹性扩展，提升读写性能。具体操作，请参见 开关读写分离。</p>
6		<p>写负载过大，内存规格不足。</p>		<p>写负载过大：您可以通过 增加分片数量 的方式来分摊写负载。若实例为标准架构，请优先升级标准架构为集群架构，提升 CPU 处理能力。具体操作，请参见 升级实例架构。升级集群架构之前需要检测兼容性，请参见 标准架构迁移集群架构检查。</p>
7	<p>同一时间点大量的 Key 过期。</p>	<p>Key 过期时间设置在同一时间点，到达过期时间点，Redis 将占用较大 CPU 资源处理 Key 的淘汰线程，引起短暂卡顿。</p>	<p>在控制台 系统监控 页面，排查指标 key 过期数，确认是否某个时间点有大量的 Key 过期。具体操作，参见 5秒监控更新说明。</p>	<p>在业务逻辑中，分散过期时间，避免 Key 集中过期。</p>
8	<p>频繁切换 DB，即频繁执行 select 命令进行 DB 切换。</p>	<p>频繁切换 DB，带来资源的过度开销。</p>	<p>借助数据库智能管家 DBbrain 的 延迟分析 功能，对 命令字分析 中 select 命令的监控数据进行确认，确认是否存在 select 请求比较多的现象。具体查询方式，请参见 命令字分析。</p>	<ul style="list-style-type: none"> 如果存储不同业务，针对频繁切换 DB 的业务，建议分开存储。 如果存储相同业务，评估 Key 名不冲突的前提下，考虑将数据存储在不同的 DB，减少 select 请求操作次数。

出流量过高

最近更新时间：2023-03-01 10:57:19

现象描述

- 现象1：出流量指标达到上限。用户收到出流量限流触发告警的消息。

```
【腾讯云】尊敬的用户，您好！您
账号（账号ID：1[REDACTED]，
昵称：6[REDACTED]）的云监控告
警持续触发。查看告警详情：✔
https://tcm.qq.com/[REDACTED]
告警内容：云数据库-Redis-内存版
(5秒粒度)-实例汇总 | 出流量使用率
[REDACTED]
告警对象：ID:crs-[REDACTED]
Instance Name: [REDACTED]
Port: 1[REDACTED]:2:6379
当前数据：[REDACTED](出流量使用率)
项目|地域：[REDACTED]|成都
告警策略：[REDACTED]
触发时间：2021-06-25 10:53:05
(UTC+08:00)
持续时间：10分钟
```

- 现象2：响应时延变大。

可能原因

- 由大 Key 引发。
若请求 Key 的 Value 值较大，则易造成出流量过高的问题。
- pipelines请求。
Pipeline 技术是把多次请求打包成一次请求发送给数据库服务端处理，在接收完所有命令执行结果后再返回给上层业务。如果一次请求较多，便会造成出流量过高。更多信息，请参见 [Redis pipeline 官网介绍](#)。
- 批量查询，如：mget、hmget 或 hgetall 等。
单次批量查询 Key 的数量较多，引起出流量过高。
- 实例配置规格不足。
业务量上涨，数据量增大，实例配额出流量达到上限，无法满足当前业务流量需求。

解决方法

步骤1：排查大 Key 问题

- 登录 [Redis 控制台](#)，通过数据库智能管家 DBbrain 的 [诊断优化](#) 功能进行大 Key 排查，并创建大 Key 分析任务，根据分析报告，优化大 Key。具体操作，请参见 [内存分析](#)。
- 若存在大 Key，在不影响业务的前提下，减少对大 Key 访问。如果是 value 过大，可以将对象拆分成多个 key-value，将操作压力平摊到多个 Redis 实例；如果是 Key 过多，可以参考 hash 结构存储，将多个 Key 存储在一个 hash 结构中。

❗ 说明：

为防止产生大 Key，设计 Value 时，建议参考如下建议。

- 建议 String 类型控制在10KB以内，hash、list、set、zset 元素个数不要超过5000。
- 对于非字符串的大 Key，建议使用 hscan、sscan、zscan 渐进式删除。

步骤2：检查业务是否使用了 pipelines

排查业务代码逻辑，建议不使用 pipeline 或者减少每次 pipelines 中的命令请求个数。如果无法确认，请 [提交工单](#) 联系腾讯云助手协助进行排查。

步骤3：排查单次查询 Key 的数量

- 登录 [Redis 控制台](#)，通过数据库智能管家 DBbrain 的 [诊断优化](#) 功能，分析性能指标 [Key 请求数量](#) 及 [mget 请求数](#) 的变化趋势，排查是否存在 Key 请求突增的现象。具体操作，请参见 [性能趋势](#)。
- 通常，建议单次操作的 Key 的个数或者元素个数的大小不超过50个。如果 value 比较大，则需要继续减少。

步骤4：升级实例规格

1. [增加副本数量](#)，并开启副本只读，分摊读负载，把当前实例的读请求转移到副本只读节点上，实现读取能力的弹性扩展，改善网络流量性能问题。具体操作，请参见 [开关读写分离](#)。
2. [增加分片数量](#)，系统将分配更多的标准带宽。若实例为标准架构，请优先升级标准架构为集群架构，提升 CPU 处理能力。具体操作，请参见 [升级实例架构](#)。升级集群架构之前需要检测兼容性，请参见 [标准架构迁移集群架构检查](#)。

步骤5：调整带宽

如果升级实例规格之后，出流量依然很高，则将带宽调至最大。增加带宽目前免费，具体操作，请参见 [带宽调整](#)。

内存使用率过高

最近更新时间：2023-08-04 16:11:32

现象描述

- 现象1: 收到内存使用率过高的告警提醒消息。

```
【腾讯云】您账号（账号 ID: 1[REDACTED]7）的云监控告警已经触发。查看告警详情：https://tcm.qq.com/\[REDACTED\]
告警内容：云数据库-Redis-内存版(5秒粒度)-实例汇总 | 内存使用率 > 80%
告警对象：ID:crs-[REDACTED] Instance Name: [REDACTED]_秒级监控单az_可删除|Ip Port: [REDACTED]:6379
当前数据：80.15%(内存使用率)
项目|地域：默认项目|广州
告警策略：默认
触发时间：2021-07-23 09:34:00 (UTC+08:00)
```

- 现象2: 在 [Redis 控制台](#) 的系统监控页面，查看到内存使用率、Key 驱逐数、P99响应时延监控指标均突增。
- 现象3: 程序写入数据提示 `command not allowed when used memory > 'maxmemory'` 错误信息。

Redis 内存占用

Redis 通常占用内存的数据包含以下几类：

- 对象内存：用户数据区，即实际存储的 Value 信息。
- 缓冲内存：包括客户端输入和输出缓冲区，以及主从同步复制缓冲区。

说明：

当执行客户端 Range 类操作或大 Key 时，Input buff 与 Output buff 占用的内存会增大，影响缓冲区，甚至导致内存溢出 OOM (Out Of Memory)。

- 内存碎片：大量的更新操作，例如 append、setrange；大量的过期键删除，释放的空间无法得到有效利用。
- 链路内存：创建子进程内存的消耗，一般这部分的消耗会比较小。

处理步骤

序号	可能原因	排查方式	解决方法
1	<ul style="list-style-type: none">写入数据量增多引起内存占用升高。未设置 Key 的 TTL 策略。	<ol style="list-style-type: none">在 登录 Redis 控制台，单击实例 ID 进入实例详情页面。选择系统监控页签，查看 Redis 实例监控指标内存使用率、Key 总个数、Key 过期数、Key 驱逐数对应的监控视图，分析内存占用与 Key 数量波动趋势是否一致。	<p>如果内存占用率与 Key 总个数是正比例增加。</p> <ul style="list-style-type: none">内存可能为业务正常写入的数据占用而增加。请评估业务需求并对 Redis 实例及时扩容。具体操作，请参见 变更实例规格。内存可能因未设置合理的过期驱逐策略以及过期 Key 删除频率而导致 Key 数量堆积而增加。<ul style="list-style-type: none">过期驱逐策略设置不合理，容易导致无效 Key 占用过多的空间。修改 Key 的过期时间、驱逐策略及过期删除频率相关参数，请在控制台 配置页面 重新配置 maxmemory-policy 参数与 hz 参数。具体操作，请参见 管理实例参数。Key 过期删除频率设置过高，即 hz 数值增大，将会占用较多 CPU 资源。请根据业务实际情况分析调整，hz 值不宜过大。
2	<ul style="list-style-type: none">写入大 Key 引起输入缓冲区溢出。	借助数据库智能管家 DBbrain 的诊断优化功能分析内存倾斜问题。具体信	针对异常大 Key 进行 Value 值拆分和优化。具体方法，请参见 热 Key 与大 key 。

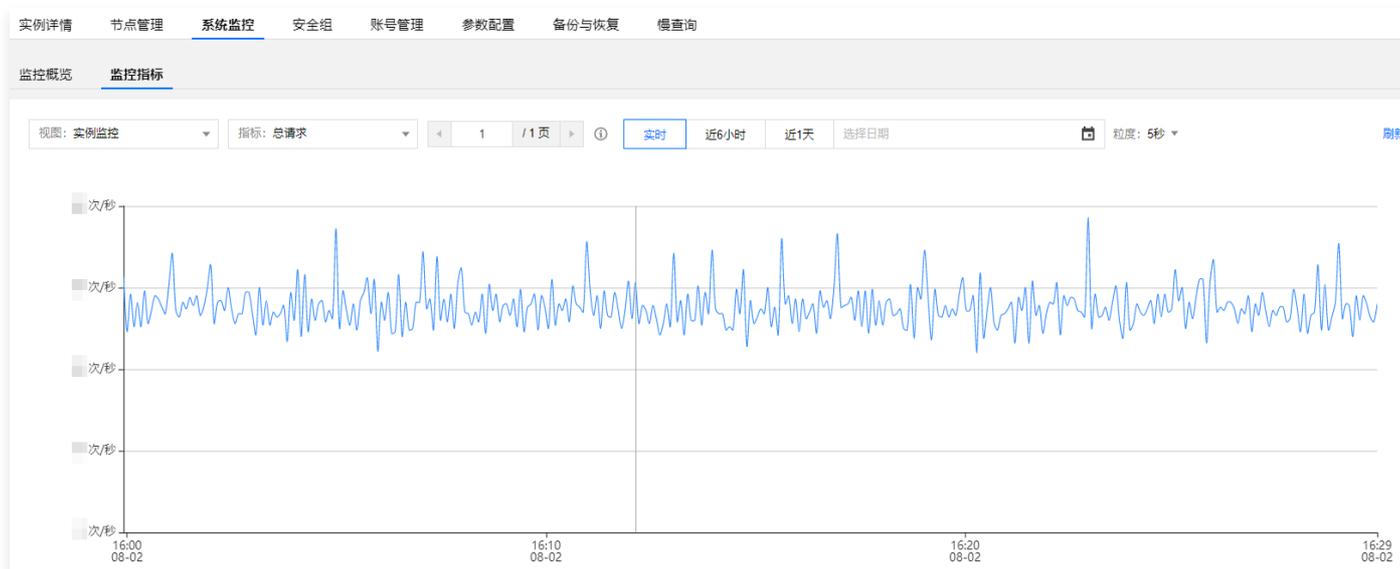
- 读取大 key、请求大量命令返回结果、或者执行 monitor 命令引起输出缓冲区溢出。息，请参见 [内存分析](#)。

总请求数过大

最近更新时间：2023-02-06 16:36:56

现象描述

- 现象1: QPS 指标较高。



- 现象2: 响应时延变大。
- 现象3: 可能引起连接超时。

可能原因

- 自身业务需要优化。
- 实例配置需要升级。

解决思路

- 查看节点负载：对于集群架构，查看节点负载情况，如果只有一个或少量节点的 QPS 超过告警值，则说明可能存在热 Key；如果大部分节点都超过，则说明 Redis 整体负载高，这种情况考虑升级实例配置。
- 查看慢查询：您可以打开控制台查看是否存在慢查询，如果存在并且时间与发生问题的时间匹配，那么可能是存在大 Key 的问题。
- 查看 CPU 使用率：您可以查看 CPU 使用率是否过高，如果过高则可能是机器资源不足，可以考虑升级实例配置。

经过判断，如果需要优化业务，您可以针对热 Key、大 Key 进行优化；如果需要升级实例配置，您可以通过开启读写分离和增加分片来满足当前的业务需求。

处理步骤

业务优化

- 登录 [Redis 控制台](#)，在实例列表，单击实例 ID，进入实例管理页面。
- 在实例管理页面，选择**系统监控**页，确认 QPS 是否过高或者有突发的热 Key。
- 在排查异常访问后，您可以对业务逻辑进行优化：
 - 对于热 Key，您可以拆分复杂数据结构，将热点 Key 拆分为若干个新的 Key 分布到不同 Redis 节点上，从而减轻压力。以哈希类型为例，该热 Key 的类型是一个二级数据结构，该哈希元素个数可能较多，可以考虑将当前 hash 进行拆分。
 - 对于大 Key，如果是 value 过大，您可以将对象拆分成多个 key-value，将操作压力平摊到多个 Redis 实例；如果是 Key 过多，您可以参考 hash 结构存储，将多个 Key 存储在一个 hash 结构中。

实例升级

读负载过大场景

读负载过大时，您可以 [开启读写分离](#)，通过 [增加副本数](#) 来分摊读负载。

⚠ 注意

开启读写分离可能会导致数据读取不一致（副本节点数据滞后于主节点），请先确认业务是否允许数据不一致的问题，请参见 [变更实例规格](#)。

1. 登录 [Redis 控制台](#)，在实例列表，选择操作列的 [配置变更](#) > [增加副本](#)。

架构版本 ▾	产品版本 ▾	已使用/总容量	创建时间 ↕	操作
Redis 4.0标准架构	内存版	34.87MB/1GB	2021-05-26 16:56:44	登录 配置变更 ▾ 更多 ▾ <ul style="list-style-type: none"> 扩容节点 缩容节点 增加副本 带宽调整

2. 在弹出的配置变更对话框，选择需更改的配置，单击**确定**。

3. 返回实例列表，待实例状态变更为**运行中**，即可正常使用。

写负载过大场景

集群架构

登录 [Redis 控制台](#)，在实例列表，选择操作列的 [配置变更](#) > [增加分片](#)。

ⓘ 说明

- 配置变更后，实例将按照新的规格计费。
- 新增分片操作，系统将自动均衡 Slot 配置，并且迁移数据，迁移操作可能会失败，建议在业务低峰期进行操作，避免迁移操作对业务访问造成影响。
- 每分片能提供的 QPS 为 8W - 10W，请按需增加。

架构版本 ▾	产品版本 ▾	已使用/总容量	创建时间 ↕	操作
Redis 4.0标准架构	内存版	34.95MB/4GB	2020-07-16 21:29:50	登录 配置变更 ▾ 更多 ▾
Redis 4.0集群架构	内存版	34.99MB/4GB	2020-07-16 11:17:41	登录 配置变更 ▾ 更多 ▾
Redis 4.0集群架构	内存版	14.54MB/64GB	2020-06-01 11:45:15	登录 <ul style="list-style-type: none"> 增加分片 删除分片 扩容节点 缩容节点 增加副本

标准架构

通过升级为集群版提升 CPU 处理能力，升级集群前需要检测兼容性，请参见 [标准架构迁移集群架构检查](#)。

1. 登录 [Redis 控制台](#)，在实例列表，单击实例 ID，进入实例详情页面。

2. 在实例详情页的规格信息处，单击架构升级。

基本信息		规格信息	
实例名称	crs- XXXXXXXXXX  	产品版本	内存版
实例ID	crs- XXXXXXXXXX 	兼容版本	Redis 4.0 版本升级
实例状态	 运行中	架构版本	标准架构 架构升级
可用区	广州四区	内存容量	4GB, 已用 0MB
所属项目	默认项目 分配至项目	内存配置	1 分片/4GB/1 副本
读写状态	读写	副本只读	未开启

3. 升级集群架构后，返回实例列表，选择操作列的[配置变更](#) > [增加分片](#)。

ⓘ 说明

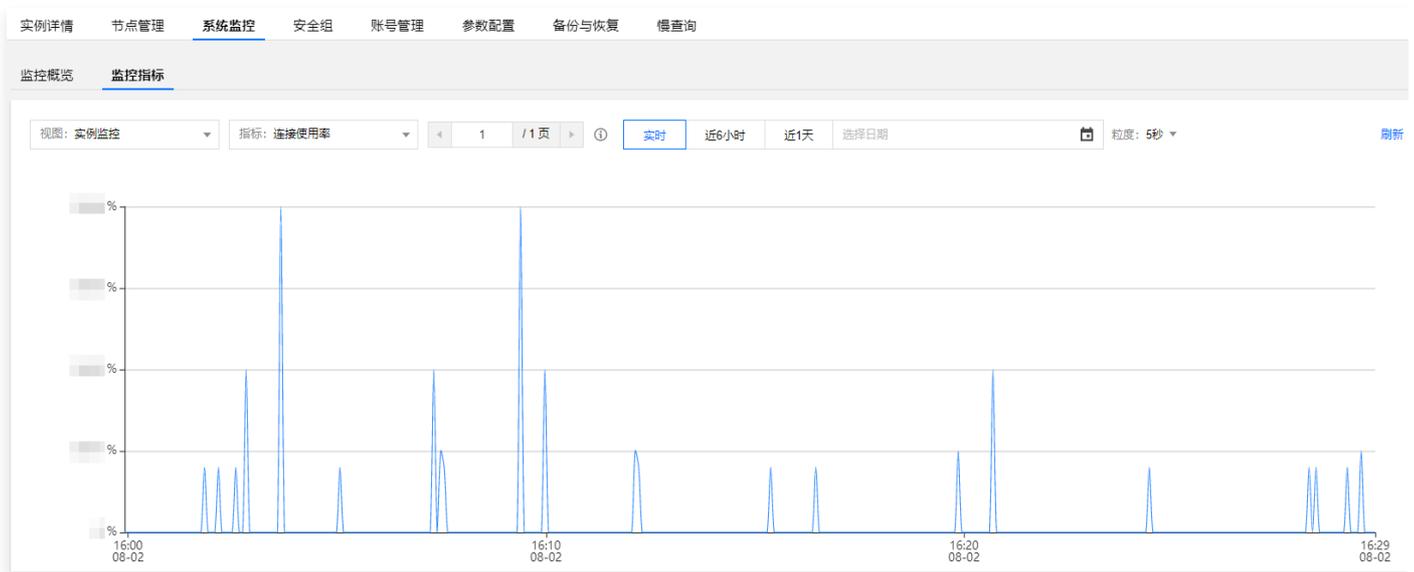
如果以上方法仍未解决问题，您还可以 [在线咨询](#) 联系售后。

连接使用率过高

最近更新时间：2022-01-26 16:17:08

现象描述

- 现象1：连接使用率指标过高。



- 现象2：无法连接。

可能原因

- 连接泄露。
- 最大连接数设置不足。

解决思路

您可以检查是否存在连接泄露。对于第三方网络连接，必须要有 finally 块执行，否则后续很容易由于不规范的编码，出现连接未正常释放的问题。

处理步骤

修改代码逻辑

检查是否存在连接泄露，如果存在泄露，您可以释放无效连接，并修正导致连接泄露的不规范代码。

修改连接配置

- 登录 [Redis 控制台](#)，在实例列表，单击实例 ID，进入实例详情页面。
- 在实例详情页面的网络信息 > 最大连接数栏，单击调整调整最大连接数。



- 在弹出的对话框，调整最大连接数，单击确定。

! 说明

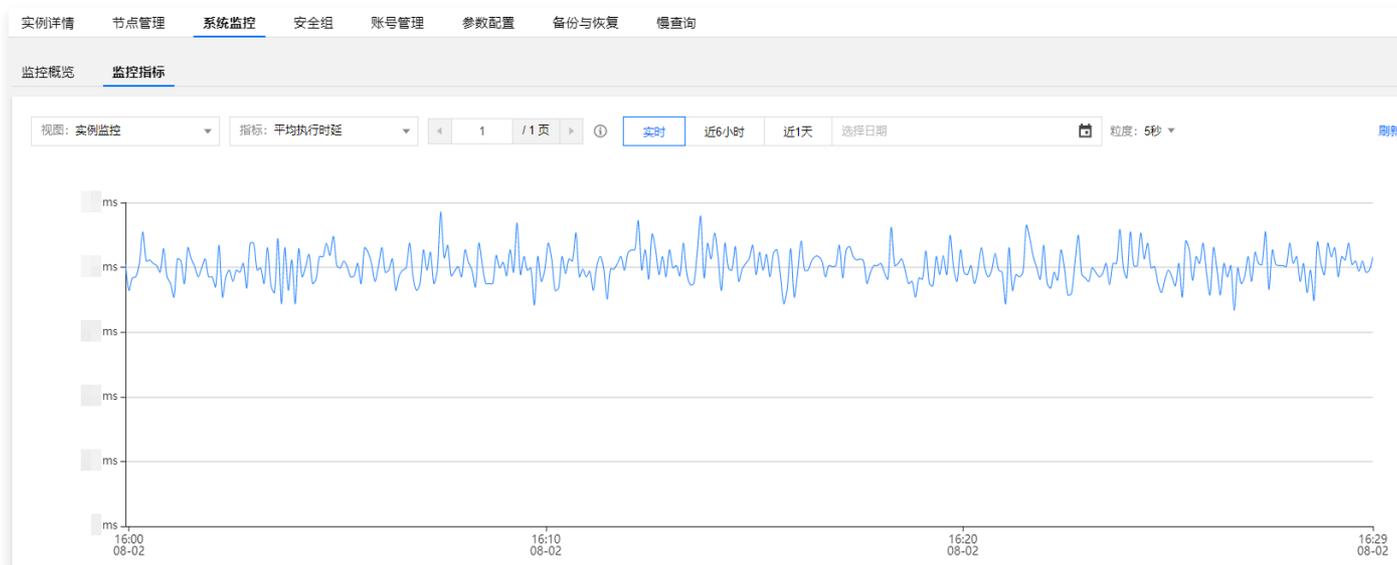
如果以上方法仍未解决问题，您可以通过 [在线支持](#) 联系售后。

执行时延过高

最近更新时间：2022-04-24 15:57:14

现象描述

- 现象1：执行时延的监控指标过高。



- 现象2：业务受到影响。

可能原因

- 总请求数过大。
- 流量、连接数受限。
- 有类似 `keys *`、`mget` 等复杂命令。

解决思路

您可以根据控制台的监控和自身的业务逻辑，来查看是否存在上述问题，并进行有针对性的优化。

处理步骤

- 若总请求数过大，请参考 [总请求数过大](#)。
- 若流量受限请参考 [出流量过高](#)，连接数受限请参考 [连接使用率过高](#)。
- 复杂命令优化，如果有 `keys *` 命令，您可以考虑使用 `scan` 命令进行分批遍历，替代更高时间复杂度的 `keys`；如果有 `mget` 命令，您可以考虑拆分为 `n` 次操作来获取 `n` 个 Key 等。

说明

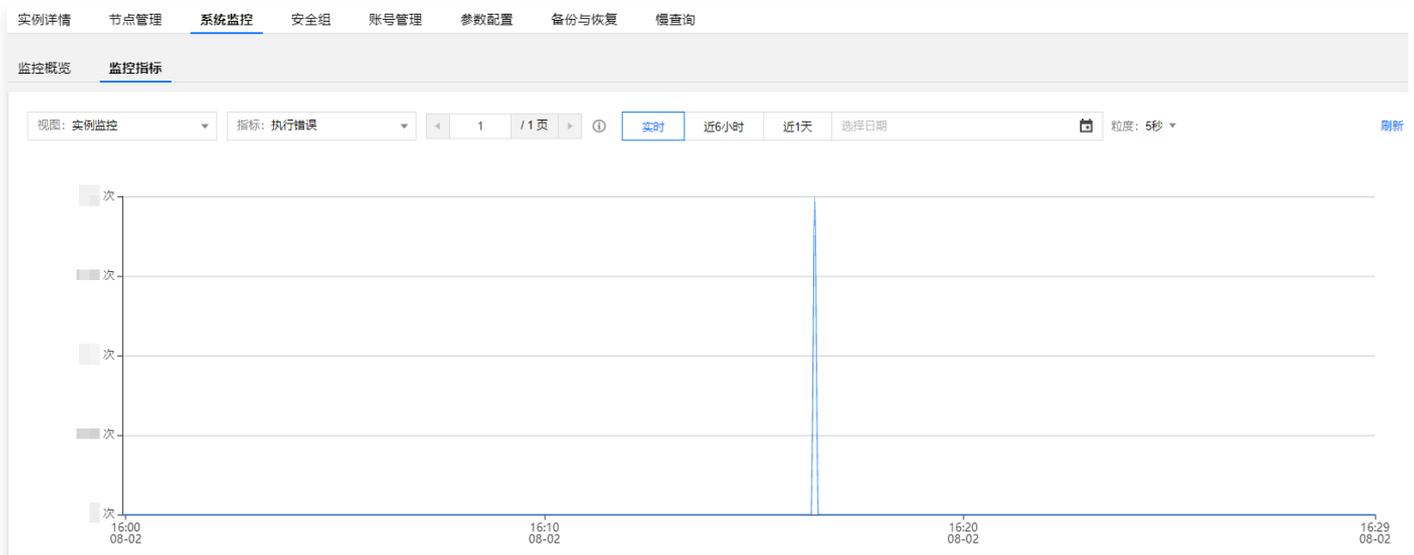
如果以上方法仍未解决问题，您可以通过 [在线支持](#) 联系售后。

执行错误

最近更新时间：2023-02-08 16:44:50

现象描述

- 现象1：出现执行错误。



- 现象2：效果不同于预期。

可能原因

- 命令输入有误。
- 在命令输入正确的情况下，可能存在：
 - 业务逻辑理解有偏差。
 - 内存已写满。

解决思路

- 若命令输入有误，请重新输入正确的命令。
- 若命令输入正确，您可以查看腾讯云 [Redis 错误码列表](#)，进行问题定位。常见问题有：
 - 执行 lua 脚本失败。
 - 执行集群命令没有附带正确的节点 ID。
 - 存在内存写满。

处理步骤

- 对于命令使用导致的错误，您可以根据对应说明和业务逻辑进行修正。
- 对于内存已写满导致的执行错误，您可以参考 [内存使用率过高](#) 进行相应处理。

说明

如果以上方法仍未解决问题，您可以通过 [在线支持](#) 联系售后。