# Redis集群在大街网的应用实践

代晓磊@大街网

DTCC

2016中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2016

数 据 定 义 未 来

SequeMedia 盛拓传媒 IT168.com ChinaUnix ITPUB

# 目录

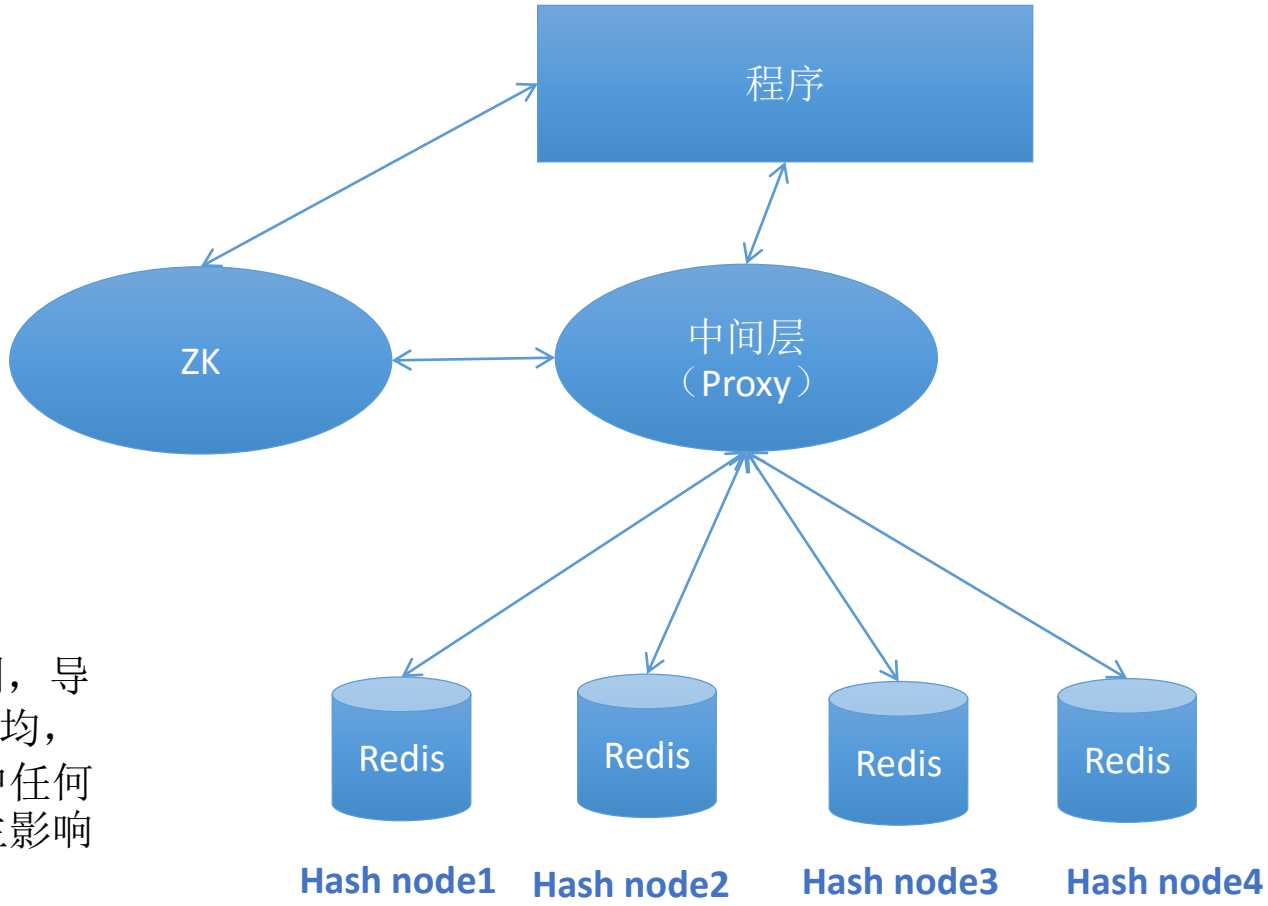
- 大街Redis缓存架构之路
- 遇到的坑
- 大街Redis自动化


## （1）Redis主从

架构：LVS+Keepalive+Redis主从

程序通过VIP+端口来访问Redis

问题：部署为单机多实例，基于服务器级别的故障切换，只有当部署在主服务器上的Redis全部挂掉或者主服务器宕机的情况下，才能切换到备服务器上的redis。




优点：避免单节点的瓶颈

缺点：中间层控制hash路由规则，导致程序比较重，而且keys分布不均，仍然无法解决高可用问题，其中任何一个节点宕机，都会对程序产生影响

1、优点:

（1）、高可用

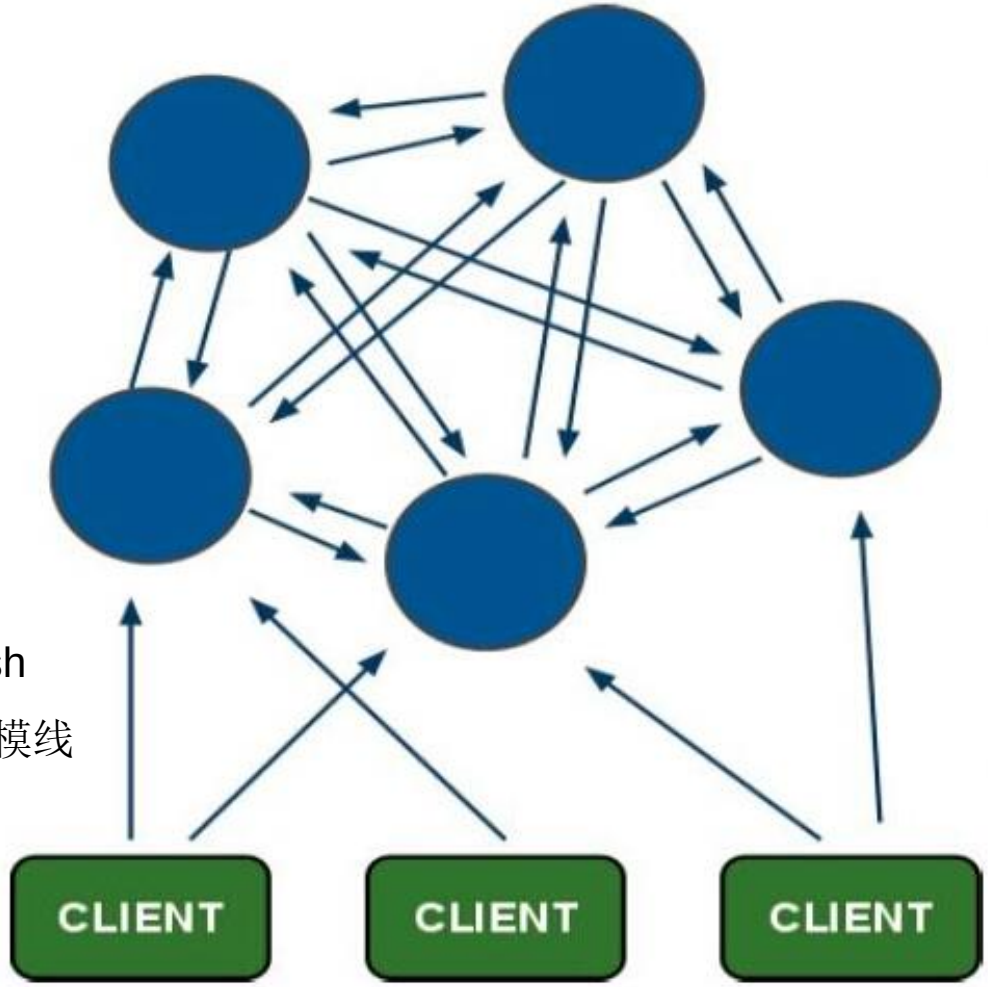
（2）、高性能

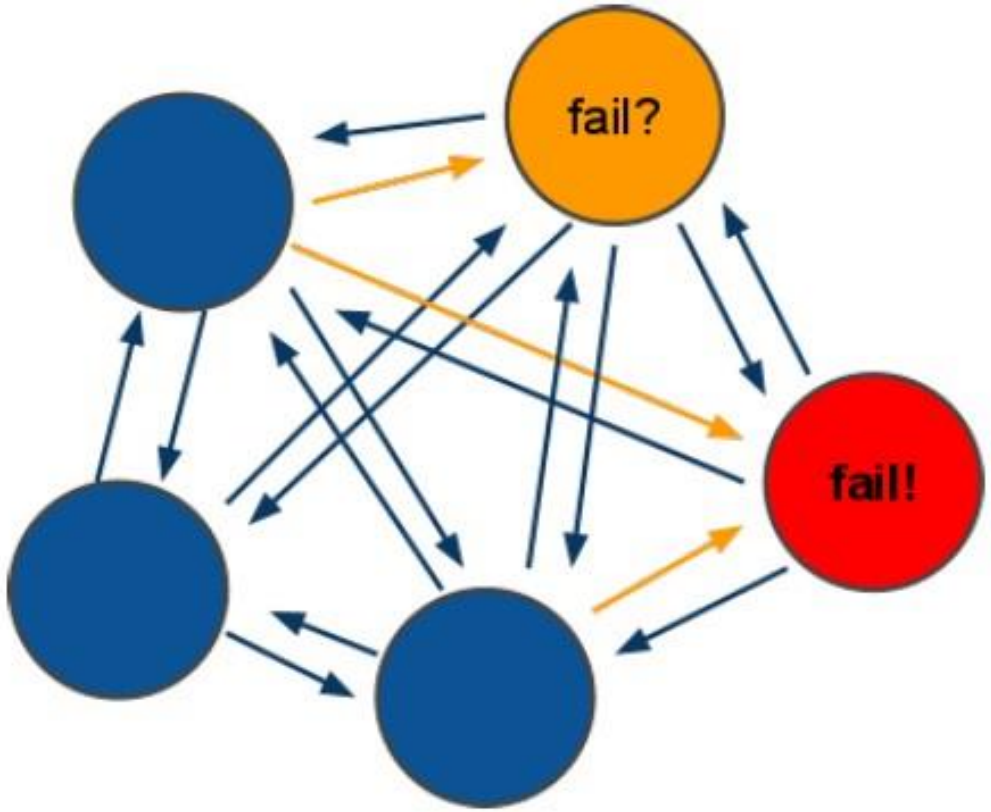
（3）、扩展性好、支持在线分片

（4）、丰富集群管理命令:cluster xxx

2、缺点

不支持多keys操作(如果keys在同一个hash slot是可以的)；只能用0号数据库；缺少大规模线上使用，不知道有多少坑。






```
108755:S 20 Apr 10:23:51.051 * Background AOF rewrite finished successfully
108755:S 21 Apr 09:46:45.590 * FAIL message received from 3a689f2d508690b4cb3a5a86dd991471ae070a74 about 7c1808daac01b818b4a3a
3fb74eda05c26af8247
108755:S 21 Apr 09:46:45.590 # Cluster state changed: fail
108755:S 21 Apr 09:46:45.615 # Start of election delayed for 796 milliseconds (rank #0, offset 35797569949).
108755:S 21 Apr 09:46:46.415 # Starting a failover election for epoch 16.
108755:S 21 Apr 09:46:46.459 # Failover election won: I'm the new master.
108755:S 21 Apr 09:46:46.459 # configEpoch set to 16 after successful failover
108755:M 21 Apr 09:46:46.459 # Connection with master lost.
108755:M 21 Apr 09:46:46.459 * Caching the disconnected master state.
108755:M 21 Apr 09:46:46.459 * Discarding previously cached master state.
108755:M 21 Apr 09:46:46.459 # Cluster state changed: ok
~
~
```

Command/day: 20亿+

Instance: 300+

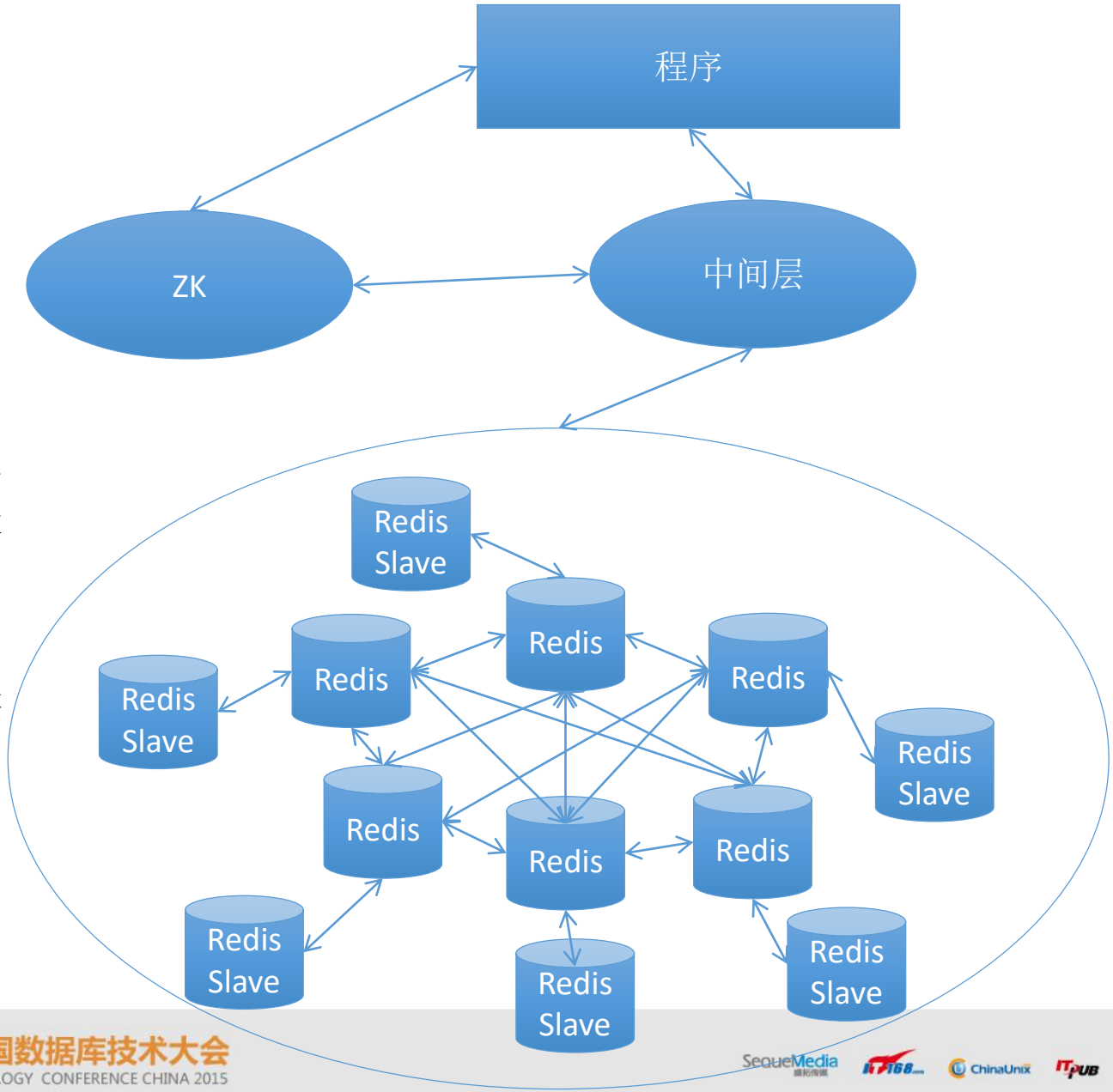
Servers： 几十台服务器

Memory used: 1T

特点：

1、基于配置中心，对程序透明，并且保证数据源只要有一个可用链接，集群访问就不受影响。

2、根据业务来划分集群，避免不同业务公用同一集群带来的相互影响。



（1）最大内存、内存过期策略、 keys过期时间设定

（2）内存碎片率(mem_fragmentation_ratio)

（3）Redis Cluster核心参数配置

（4）keys *、flushdb危险命令

（5）连接周期性异常

（6）bgsave导致的集群阻塞问题（ AOF or Rdb？ ）

# （1）最大内存、keys过期时间设定、内存过期策略

1、最大内存没有设定

config set maxmemory=?

2、Keys过期时间

根据需求：存储OR Cache

Cache：设定ttl避免内存超限

存储：不设定，必须监控内存使用并自动调整内存

3、合理配置Redis的过期策略

volatile-lru：使用LRU算法从已设置过期时间的数据集合中淘汰数据。

volatile-ttl：从已设置过期时间的数据集合中挑选即将过期的数据淘汰。

volatile-random：从已设置过期时间的数据集合中随机挑选数据淘汰。

allkeys-lru：使用LRU算法从所有数据集合中淘汰数据。

allkeys-random：从数据集合中任意选择数据淘汰

no-enviction：禁止淘汰数据。

mem_fragmentation_ratio=used_memory_rss/used_memory

1、mem_fragmentation_ratio>1

问题原因：
Redis没有内存回收机制，如果批量过期数据或者删除数据，内存的碎片可能就比较大。

解决方式：
重启Redis实例

```
[root(  )@  ~]# redis-cli -c -p
127.0.0.1:  > info memory
# Memory
used_memory:3506713360
used_memory_human:3.27G
used_memory_rss:10736922624
used_memory_peak:10107829136
used_memory_peak_human:9.41G
used_memory_lua:35840
mem_fragmentation_ratio:3.06
mem_allocator:jemalloc-
```

2、mem_fragmentation_ratio<1

问题产生原因:
一般发生在内存紧张的服务器，就是本身内存剩余不多，redis申请不到足够的内存，这样就会使用swap

解决方式:
增加物理内存，或者减少redis内存占用(删除一些keys)

```
used_memory:10891866632
used_memory_human:10.14G
used_memory_rss:3313442816
used_memory_peak:10897787944
used_memory_peak_human:10.15G
mem_fragmentation_ratio:0.30
mem_allocator:jemalloc-[redacted]
```

1、cluster-node-timeout 5000

2、cluster-require-full-coverage no

1、问题描述

线上Service访问Redis异常，大量的5XX

2、问题分析

查看慢日志：执行keys *时间跟Service Error时间一致

```
127.0.0.1:XXXX> SLOWLOG get 10
 1) 1) (integer) 545
    2) (integer) 1446775933
    3) (integer) 8038973
    4) 1) "keys"
       2) "*visit_uid”
mysql> select from_unixtime(1446775933);
+---------------------------+
| from_unixtime(1446775933) |
+---------------------------+
| 2015-11-06 10:12:13       |
+---------------------------+
1 row in set (0.01 sec)
```

3、问题解决

在配置文件中禁用危险命令：keys rename ‘’

使用Redis自带客户端：redis-cli -h -p –scan –pattern ‘*visit’实现同样的功能

# （5）Redis连接周期性异常

1、问题描述

某个redis节点，监控发现每隔半个小时连接数突增10倍以上

2、问题分析

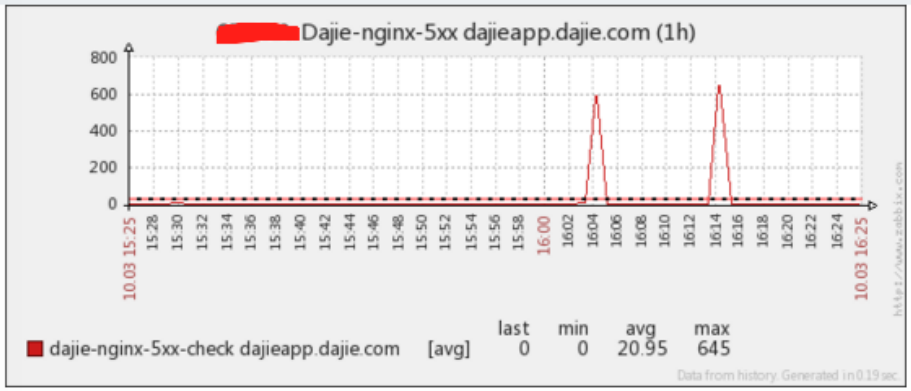
统计一段时间的redis连接发现，redis瞬时连接都正常，初步判定是某个ip频繁创建和关闭连接导致

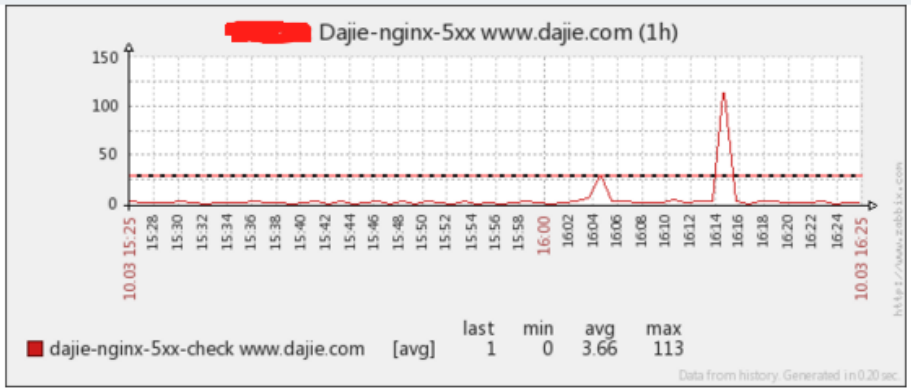
3、问题解决

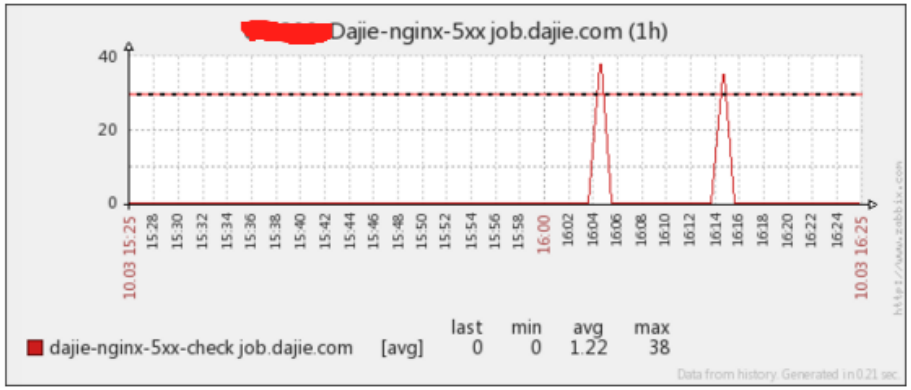
通过tcpflow来抓取问题时间段的连接数据包情况，最终定位到具体的ip，并且找到原因。

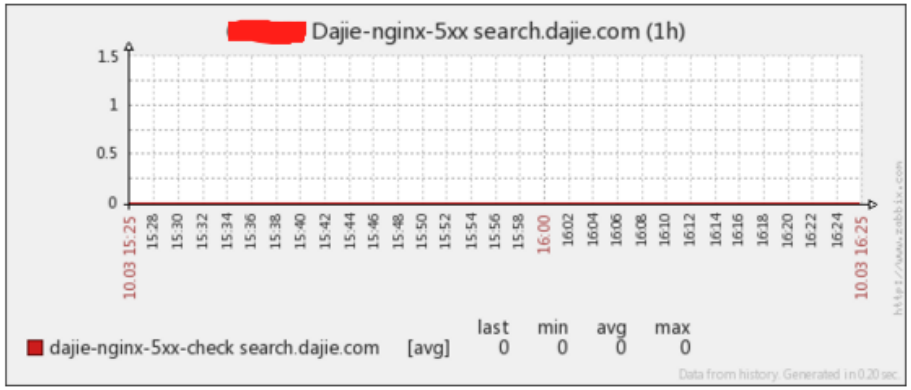
tcpflow -c -p -i em1 dst 192.168.1.30 and port 6379 > dxl2.log

1、问题现象：线上官客、job大量的5XX



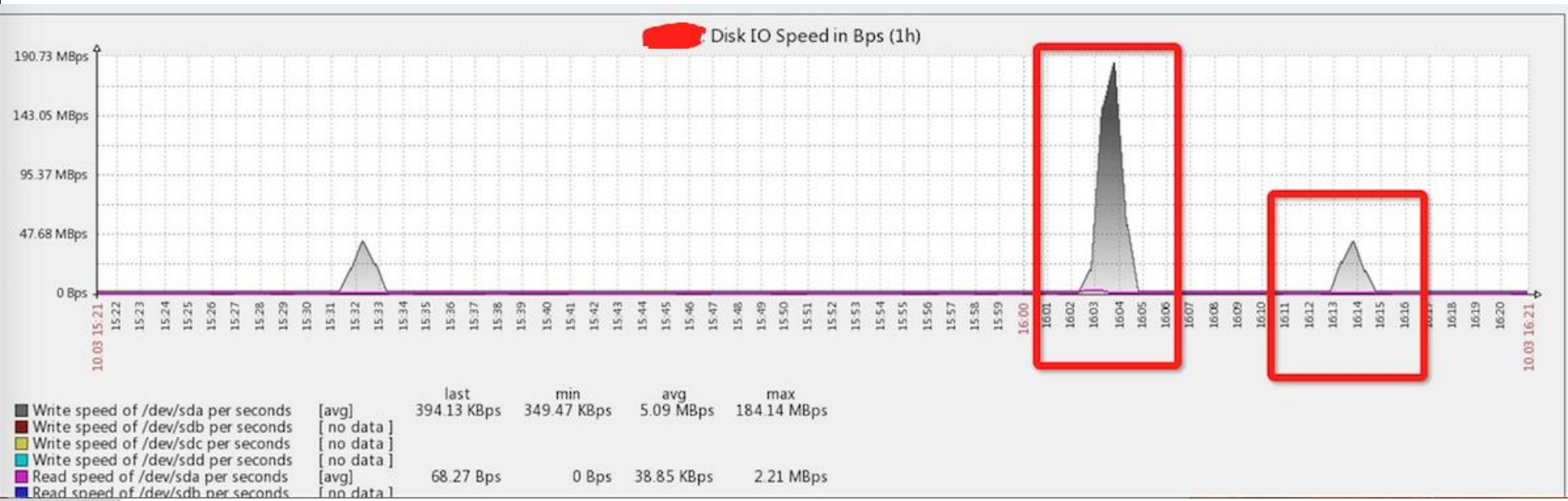



2、问题排查：

查看job集群的Redis log，发现aof写入异常：disk is busy？

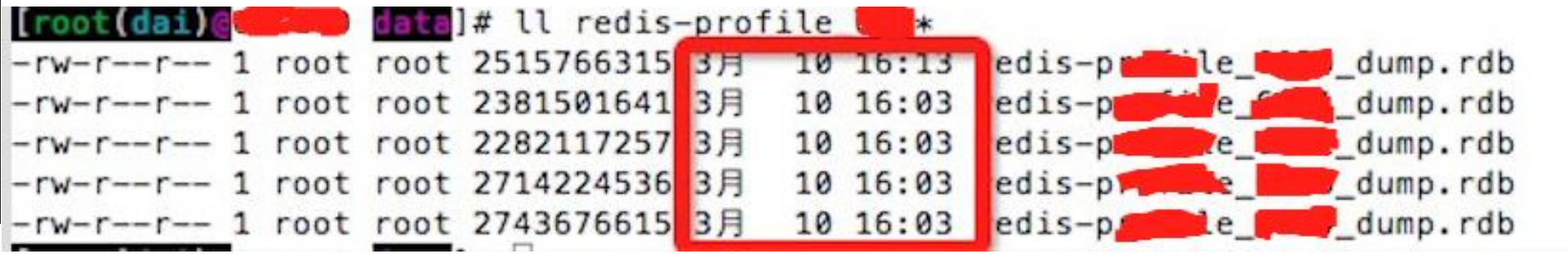
```
[root(  @      0 logs]# grep "M 10 Mar 16" rc-job-base-    .log
65686:M 10 Mar 16:03:35.095 * Asynchronous AOF fsync is taking too long (disk is busy?). Writing the AOF buffer without waiting for
 fsync to complete, this may slow down Redis.
65686:M 10 Mar 16:12:58.059 * Asynchronous AOF fsync is taking too long (disk is busy?). Writing the AOF buffer without waiting for
 fsync to complete, this may slow down Redis.
```

查看服务器的IO负载情况，发现在16点03和10分有2个IO峰值，跟job抖动的时间重合



3、问题定位：

查看这个时间段的高IO操作，因为这个是redis服务器，导致高IO只能是rdb数据落地。最终发现是profile业务的bgsave导致。



**4、问题解决：AOF or RDB？**

AOF：

优点：aof文件是一个只　　加操作　　日志文件(append only file)　　。　　并且有序的保存了对数据库执　　所有写入操作，如果有类似flushdb操作，只需要删除文件中flushdb的操作，重启redis即可

缺点：aof文件比较大，根据不同的fsync策略，aof的速度可能比rdb慢

RDB：

优点：文件紧凑，非常适合备份以及快速恢复

缺点：只是某一时刻的内存快照，即数据不是最新，适合缓存集群，不适合存储集群。

总结：

根据需求选择，缓存集群使用RDB备份，存储集群使用AOF，采用SSD设备；迁移存储集群到新服务器

（1）自动化之规范

（2）自动化部署、配置(修改内存等配置)

（3）自动化监控

（4）redis自动化迁移工具

（5）集群扩容

（6）自动化备份

（7）分析(slowlog分析、keys分布)

1、keys命名规范：

redis的key命名尽量简单明确，最好根据子业务名(或者缩写)命名，keys命名禁止出现各种复杂符号。

比如下面的keys，下面的key首先过长、然后有空格、还有&符号

test_guangzhou spengler automated vending technologies reaserch & development co., ltd

2、Redis使用规范：

禁止将大量的成员存储到一个hash key中

禁止连接线上redis执行keys *dxl这种方式来过滤keys

keys建议设定过期时间，除非是把redis当存储用。

合理使用Redis的数据类型


dajie大街
DB自动化运维系统
核心表管理
监控配置管理
备份管理
Redis管理
Redid实例表
Reids详情
Reids命中率
Reids内存使用
Reids链接使用
Redis慢日志查询
Cobar管理
统计分析
Redid实例表 - Add
主机ip: *
Redis实例端口号: *
实例类型(1:主 2:备): *
状态0:停 1:启 2:初始化: *
Version: *
最大内存: *
是否备份:0 不 1 是: *
集群名: *
访问域名: *
业务组:
集群组号(单实例为0): *
实例添加时间: *
修改时间: *
Save and close
Cancel

缘由：由于人工设定内存失误导致集群节点不可用
目的：自动化脚本来智能配置内存

```
[root(dai)@[redacted] [redacted]scripts]# python redis_add_memory.py 2016041401
argv_nums: 2
Redis cluster num:2016041401
Redis:[redacted] max memory = redis_now_maxmemory,dont need add memory!!
Redis:[redacted] max memory = redis_now_maxmemory,dont need add memory!!
Redis:[redacted] max memory = redis_now_maxmemory,dont need add memory!!
Redis:[redacted] max memory = redis_now_maxmemory,dont need add memory!!
Redis:[redacted] max memory = redis_now_maxmemory,dont need add memory!!
Redis:[redacted] max memory = redis_now_maxmemory,dont need add memory!!
```

dajie大街 | DB自动化运维系统

Logged in as: administrator

- 核心表管理
- 监控配置管理
- 备份管理
- Redis管理
  - Redid实例表
  - Reids详情
  - Reids命中率
  - Reids内存使用
  - Reids链接使用
  - Redis慢日志查询
- Cobar管理
- 统计分析

## Reids命中率 - Administration

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Next

| 主机ip | Redis实例端口号 | keys数 | 命中率 | 阶段命中数 | 阶段丢失数 | 阶段命令数 | 累计命令数 | 每秒ops | 累计命中 |
|---|---|---|---|---|---|---|---|---|---|
| 10.10.[redacted] | [redacted] | 1621[redacted] | 1.00 | 2077[redacted] | 7[redacted] | 2078[redacted] | 1901[redacted]269 | 3[redacted] | 1631[redacted] |
| 10.10.[redacted] | [redacted] | 1620[redacted] | 1.00 | 2401[redacted] | 12[redacted] | 24[redacted] | 1880[redacted] | 4[redacted] | 161[redacted] |
| 10.10.[redacted] | [redacted] | 1620[redacted] | 0.97 | 28[redacted] | 10[redacted] | 3[redacted] | 1901[redacted] | 5[redacted] | 163[redacted] |
| 10.10.[redacted] | [redacted] | 1620[redacted] | 0.96 | 278[redacted] | 12[redacted] | 3[redacted] | 1902[redacted] | 4[redacted] | 163[redacted] |
| 10.10.[redacted] | [redacted] | 1620[redacted] | 1.00 | 2403[redacted] | 10[redacted] | 24[redacted] | 1856[redacted] | 4[redacted] | 158[redacted] |
| 10.10.[redacted] | [redacted] | 1620[redacted] | 0.99 | 1746[redacted] | 14[redacted] | 177[redacted] | 1904[redacted] | 2[redacted] | 162[redacted] |
| 10.10.[redacted] | [redacted] | 1620[redacted] | 1.00 | 230[redacted] | 15[redacted] | 230[redacted] | 1605[redacted] | 3[redacted] | 133[redacted] |
| 10.10.[redacted] | [redacted] | 1620[redacted] | 1.00 | 213[redacted] | 28[redacted] | 213[redacted] | 1579[redacted] | 40[redacted] | 131[redacted] |
| 10.10.[redacted] | [redacted] | 1619[redacted] | 0.95 | 291[redacted] | 14[redacted] | 320[redacted] | 1904[redacted] | 1[redacted] | 163[redacted] |
| 10.10.[redacted] | [redacted] | 1619[redacted] | 1.00 | 2309[redacted] | 12[redacted] | 2312[redacted] | 1832[redacted] | 39[redacted] | 156[redacted] |
| 10.10.[redacted] | [redacted] | 1619[redacted] | 0.80 | 760[redacted] | 18[redacted] | 120[redacted] | 1904[redacted] | 2[redacted] | 1634172 |
| 10.10.[redacted] | [redacted] | 1619[redacted] | 1.00 | 2304[redacted] | 15[redacted] | 23[redacted]823 | 1626[redacted] | 3838[redacted] | 1356219 |

dajie大街 DB自动化运维系统

Logged in as: **administrator**

- 核心表管理
- 监控配置管理
- 备份管理
- Redis管理
  - Redid实例表
  - Reids详情
  - Reids命中率
  - Reids内存使用
  - Reids链接使用
  - Redis慢日志查询
- Cobar管理
- 统计分析

**Reids内存使用 - Administration**

1 | 2 | 3 | 4 | 5 | 6 | Next

| 主机ip | Redis实例端口号 | keys数 | 阶段命令数 | 分配内存 | 常驻系统内存 | 使用过内存的峰值 | 每秒ops | Sta |
|---|---|---|---|---|---|---|---|---|
| | | | | G | | | | 2016-05- |
| [redacted] | [redacted] | [redacted] | [redacted] | 5.74G | 7507316736 | 6.78G | [redacted] | 2016-0 |
| [redacted] | | [redacted] | [redacted] | 5.74G | 7507316736 | 6.78G | [redacted] | 2016-0 |
| [redacted] | | [redacted] | [redacted] | 5.74G | 7507316736 | 6.78G | [redacted] | 2016-0 |
| [redacted] | | [redacted] | [redacted] | 3.27G | 10736922624 | 9.41G | [redacted] | 2016-0 |
| [redacted] | | [redacted] | [redacted] | 3.27G | 10736922624 | 9.41G | [redacted] | 2016-0 |

1 | 2 | 3 | 4 | 5 | 6 | Next

dajie大街 DB自动化运维系统

Logged in as: ad

- 核心表管理
- 监控配置管理
- 备份管理
- Redis管理
  - Redid实例表
  - Reids详情
  - Reids命中率
  - Reids内存使用
  - Reids链接使用
  - Redis慢日志查询
- Cobar管理
- 统计分析

**Reids链接使用 - Administration**

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Next

Show: 5

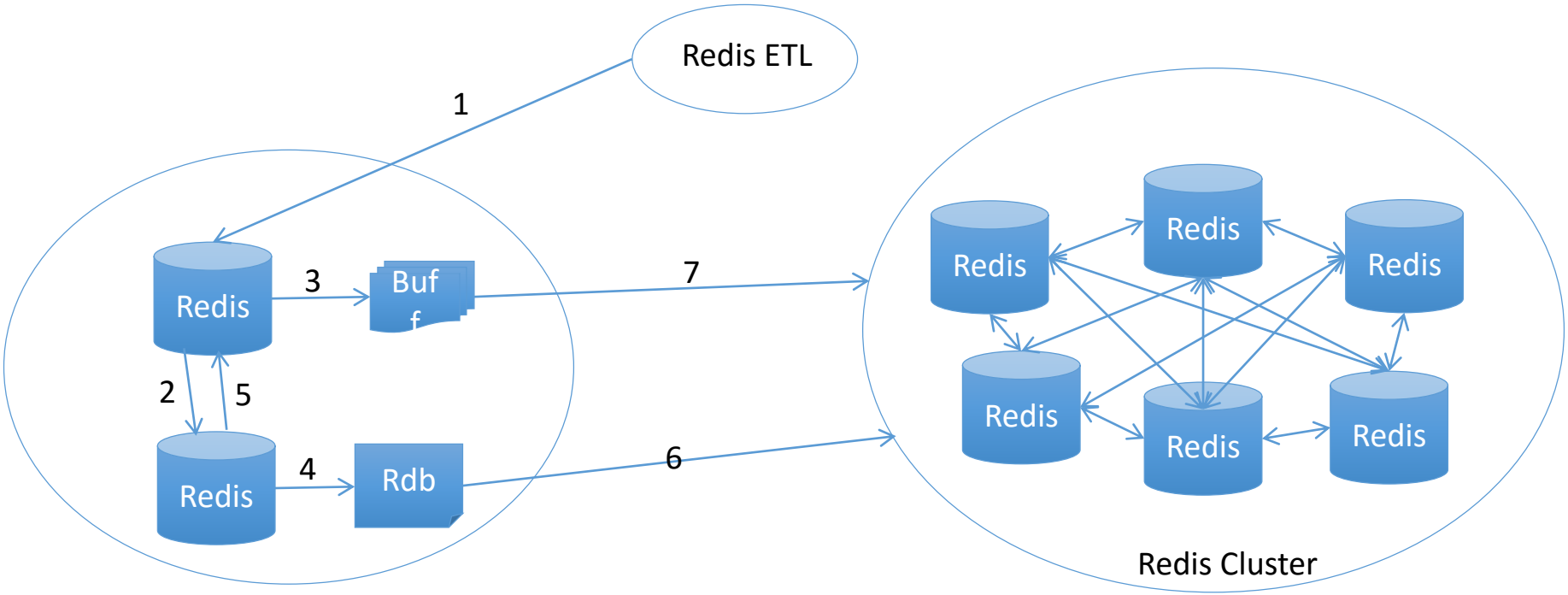
| 主机ip | Redis实例端口号 | keys数 | 当前链接数 | 阶段链接数 | 累计链接数 | Stats date | |
|---|---|---|---|---|---|---|---|
| | | | | | | | Search (Extended) |
| [redacted] | [redacted] | [redacted] | 27 | 196 | [redacted] | 2016-05-03 16:20:01 | |
| [redacted] | [redacted] | [redacted] | 81 | 553 | [redacted] | 2016-05-03 16:20:01 | |
| [redacted] | [redacted] | [redacted] | 81 | 558 | [redacted] | 2016-05-03 16:20:01 | |
| [redacted] | [redacted] | [redacted] | 21 | 158 | [redacted] | 2016-05-03 16:20:01 | |
| [redacted] | [redacted] | [redacted] | 21 | 154 | [redacted] | 2016-05-03 16:20:01 | |

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Next

(Records 1 - 5 of 38745)

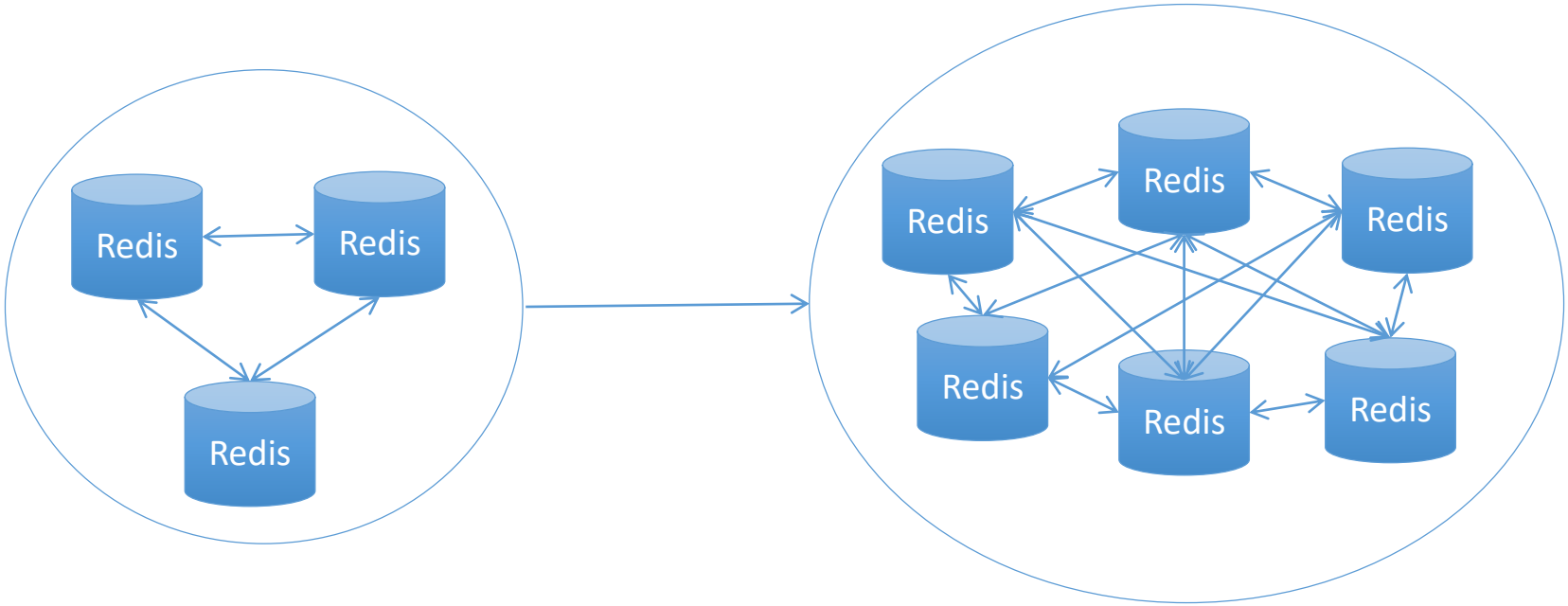
核心：基于redis port
功能：

实现单节点中部分keys的迁移
单redis迁移到redis cluster
自建hash集群迁移到redis cluster



可以使用2种方式对集群扩容：

1、为集群所有节点扩展内存
2、增加新节点
　直接扩展1倍的节点
http://redis.io/commands/cluster-setslot



dajie.com大街 | DB自动化运维系统

Logged in as: **administrator**

- 核心表管理
- 监控配置管理
- 备份管理
- **Redis管理**
  - Redid实例表
  - Reids详情
  - Reids命中率
  - Reids内存使用
  - Reids链接使用
  - Redis慢日志查询
- Cobar管理
- 统计分析

**Redid实例表 - Administration**

Redid实例表 Add

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Next

| 主机Ip | Redis实例端口号 | 实例类型(1:主 2:备) | 状态0:停 1:启 2:初始化 | Version | 最大内存 | 是否备份:0 不 1 是 | 集群名 | |
|---|---|---|---|---|---|---|---|---|
| [redacted] | [redacted] | 1 | 1 | [redacted] | 4gb | 0 | [redacted] | |
| [redacted] | [redacted] | 1 | 1 | [redacted] | 4gb | 0 | [redacted] | |
| [redacted] | [redacted] | 1 | 1 | [redacted] | 4gb | 0 | [redacted] | |
| [redacted] | [redacted] | 1 | 1 | [redacted] | 12gb | 0 | [redacted] | |
| [redacted] | [redacted] | 1 | 1 | [redacted] | 12gb | 0 | [redacted] | |

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Next

# （7）性能分析之slowlog

dajie.com 大街 | DB自动化运维系统

Logged in as: [redacted]

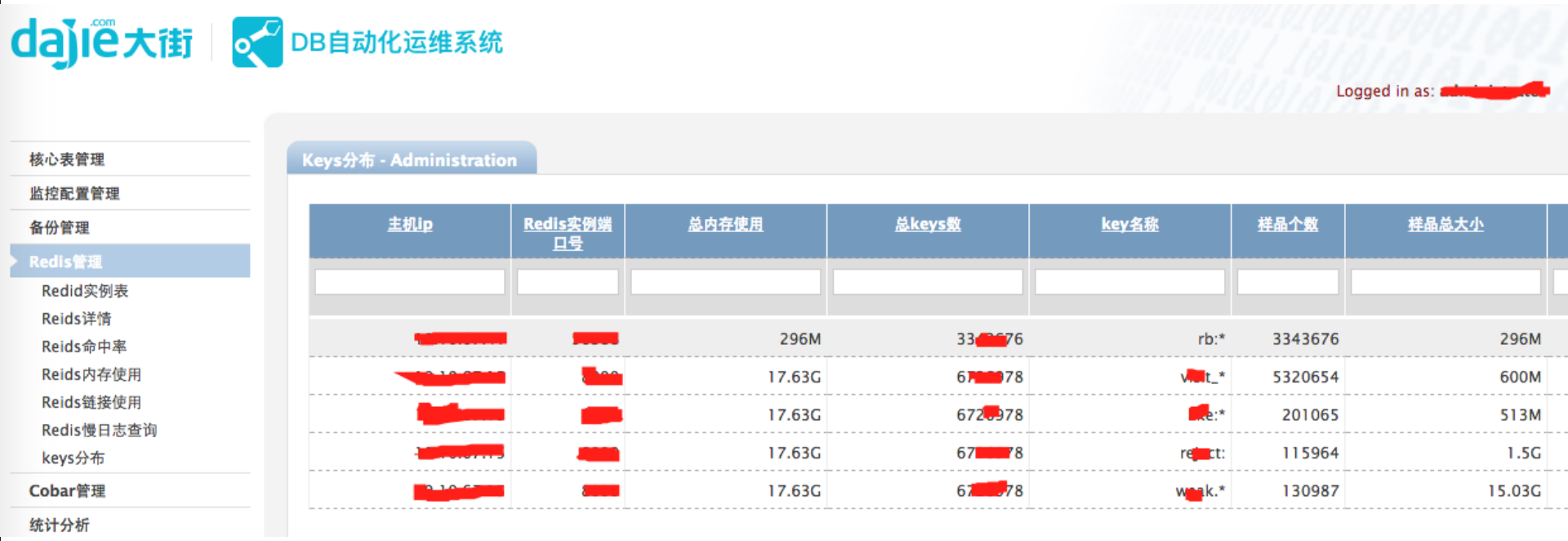
- 核心表管理
- 监控配置管理
- 备份管理
- Redis管理
  - Redid实例表
  - Reids详情
  - Reids命中率
  - Reids内存使用
  - Reids链接使用
  - Redis慢日志查询
- Cobar管理
- 统计分析

## Redis慢日志查询 - Administration

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Next

Show: 25

| 主机ip | Redis实例端口号 | 命令产生时间 | 命令执行时间(微秒) | 命令详情 | Stats date | |
|---|---|---|---|---|---|---|
| | | | | | | Search (Extended) |
| [redacted] | [redacted] | 2016-05-03 16:29:04 | 319742 | KEYS [redacted]:* | 2016-05-03 16:30:23 | |
| [redacted] | [redacted] | 2016-05-03 16:29:03 | 334743 | KEYS [redacted]:* | 2016-05-03 16:30:23 | |
| [redacted] | [redacted] | 2016-05-03 16:29:01 | 262044 | KEYS [redacted]:* | 2016-05-03 16:30:23 | |
| [redacted] | [redacted] | 2016-05-03 16:29:00 | 400251 | KEYS [redacted]:* | 2016-05-03 16:30:23 | |
| [redacted] | [redacted] | 2016-05-03 16:28:59 | 252729 | KEYS [redacted]:* | 2016-05-03 16:30:23 | |
| [redacted] | [redacted] | 2016-05-03 16:28:58 | 396656 | KEYS [redacted]:* | 2016-05-03 16:30:23 | |
| [redacted] | [redacted] | 2016-05-03 16:15:30 | 16274 | CONFIG GET maxmemory | 2016-05-03 16:30:23 | |
| 10[redacted] | 8[redacted] | 2016-05-03 16:17:52 | 23575 | ZRANGEBYSCORE [redacted] 1.1179544E8 2.147483647E9 | 2016-05-03 16:30:20 | |


dajie大街
DB自动化运维系统
Logged in as:
核心表管理
监控配置管理
备份管理
Redis管理
Redid实例表
Reids详情
Reids命中率
Reids内存使用
Reids链接使用
Redis慢日志查询
keys分布
Cobar管理
统计分析
Keys分布 - Administration
主机Ip
Redis实例端口号
总内存使用
总keys数
key名称
样品个数
样品总大小
296M
rb:*
3343676
296M
17.63G
5320654
600M
17.63G
201065
513M
17.63G
115964
1.5G
17.63G
130987
15.03G


大街网-招聘启事
1、NoSQL DBA(Redis or Mongodb)
2、MySQL DBA
3、运维系统开发人员(Python)
联系邮箱：542369334@qq.com

# Thank You

dajie.com 大街