

百度分布式redis集群平台 开发与运维

胡东旭

向上草根交流群组织交流：www.eeqee.com

Mongodb精英qq总群 99404458

Redis精英qq总群 243375697

Neo4j精英qq总群 368193928

2000人NOSQL总qq群 224718662

向上交流群 第4年

草根交流，1月1次，面对面

www.eeqee.com

公众微信号 eegee_com

始于2011年5月

学习，工作，友谊，共赢

负责人向上电话微信：18010082538

活动预告1(技术类):

分布式对象存储体系CEPH原理及应用

百度大侠 解读MySQL内核

通过案例深入研究Goldengate问题解决思路

细说垂直型网络爬虫技术

用 Excel+SQL玩转BI（含APP）

BI的 Excel 解决方案

郑保卫博士：数据架构之道 + 数据建模之道

数据科学家之道1讲：知识体系地图

数据科学家之道2讲：企业数据治理与元数据

数据科学家之道3讲：企业数据架构常见方案及应用形式

数据科学家之道4讲：数据产品的设计方法及数据驱动的企业

R统计分析2讲：R统计分析快速入门

R统计分析3讲：R语言以及特点

HR猎头必备IT知识第1讲：关系数据库的开发

唯一报名地址 www.eegee.com

活动预告2(非IT技术类):

如何搞定面试官 PK 如何看懂求职者

如何建立亲子关系 & 如何帮助孩子长大

个人成长：带着孩子跨越人生的5个栏杆

婚恋之道1：如何选择一生的伴侣

婚恋之道2：男女有“别”

婚恋之道3：性、爱、责任

必备金融知识：社保及常备保险深入分析

必备金融知识：银行与证券

理财实操：外汇保证金实战

创业交流：App展示与合作洽谈

互联网医疗现状及未来可能

有机食品PK转基因食品

唯一报名地址 www.eeqee.com

Redis简述和介绍

1 Redis的概念以及特性

2 Redis常用命令

3 Redis应用场景

4 为何引入BDRP

BDRP

1. BDRP简述与介绍

2. BDRP的开发

3. BDRP的运维

4. Q & A

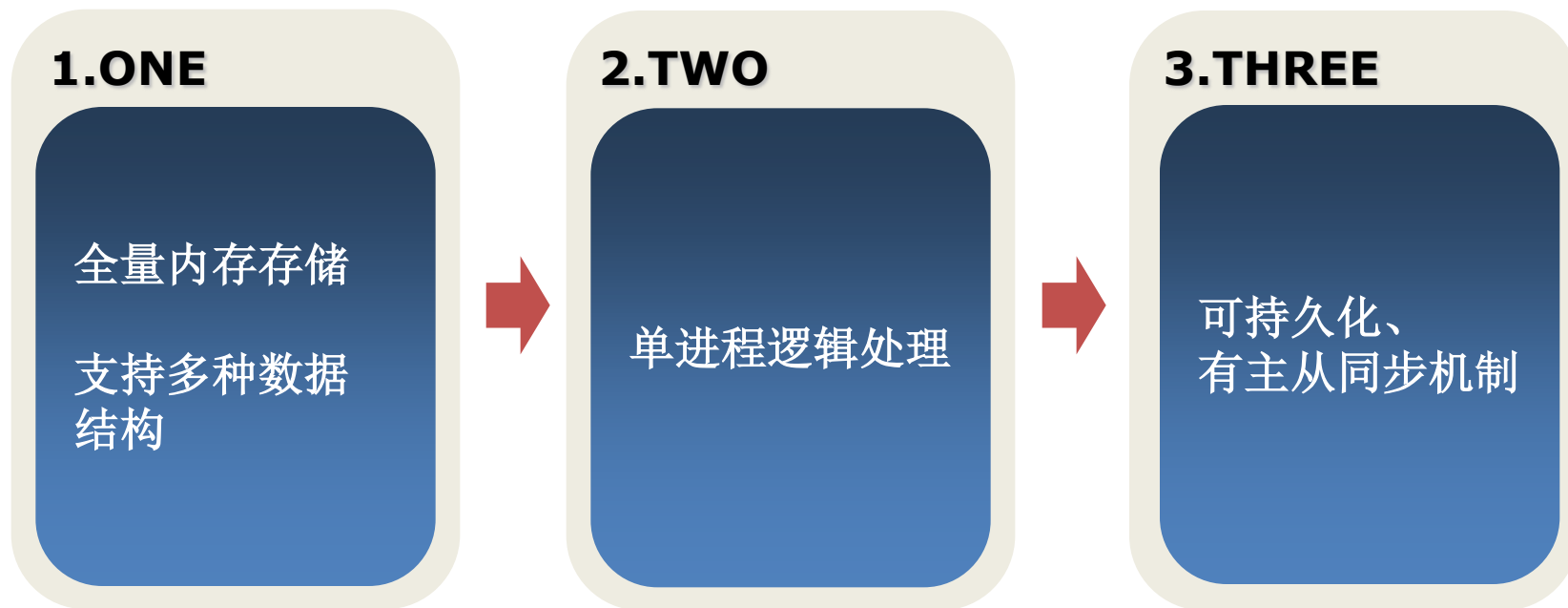
redis基本概念

◆BDRP（百度分布式redis平台）

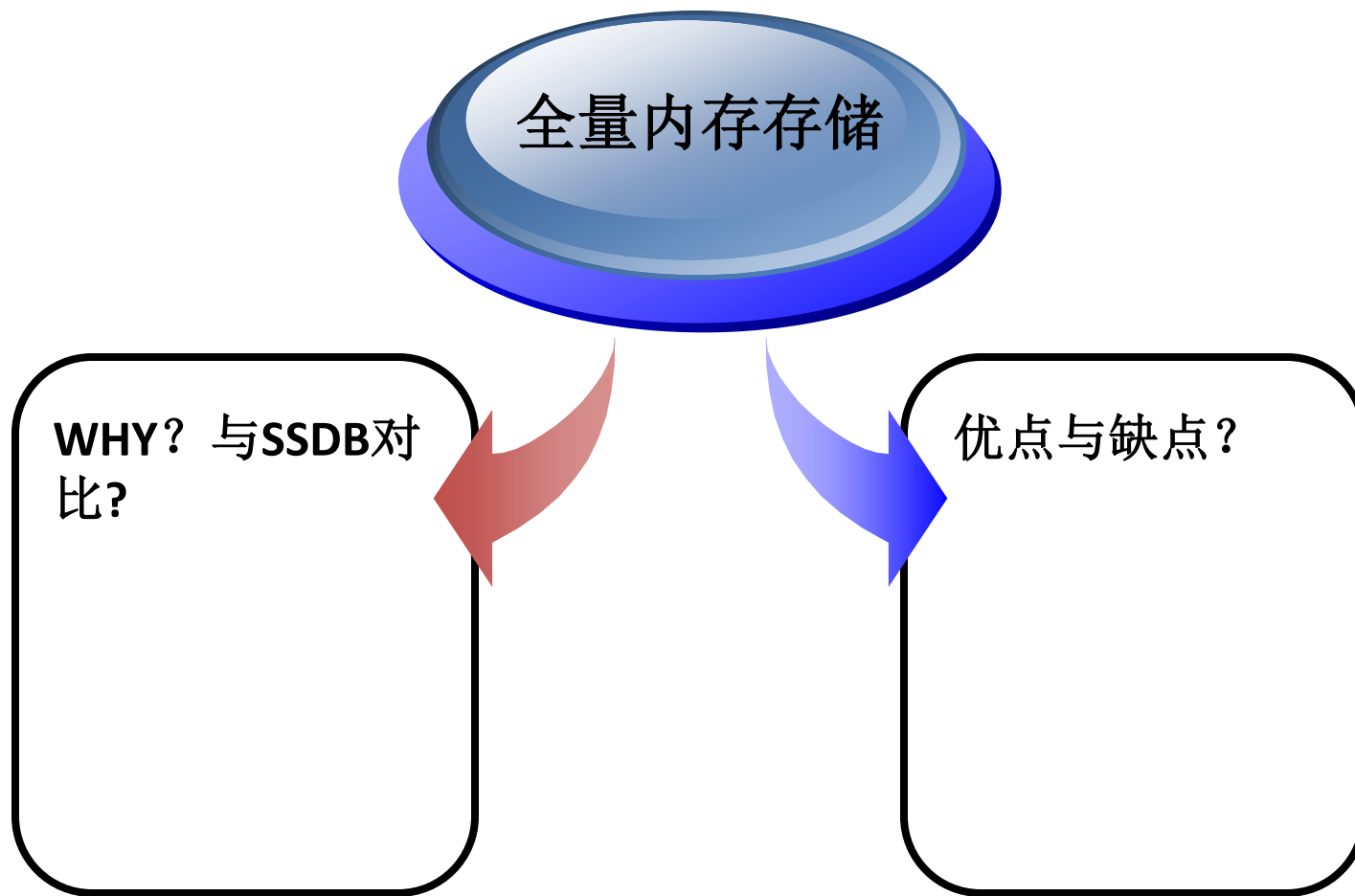
Redis的定义与特性：

- 1、纯C编写**
- 2、K/V存储型Nosql**

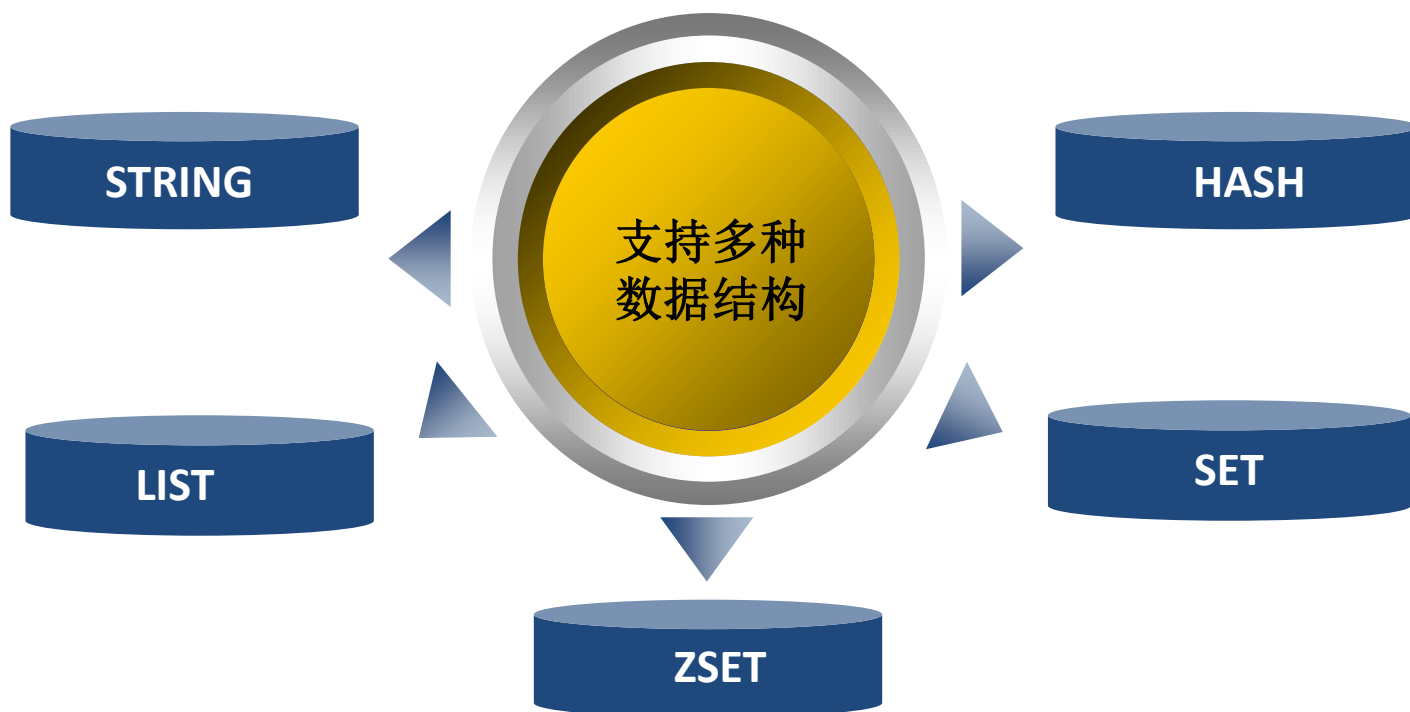
Redis基本特性



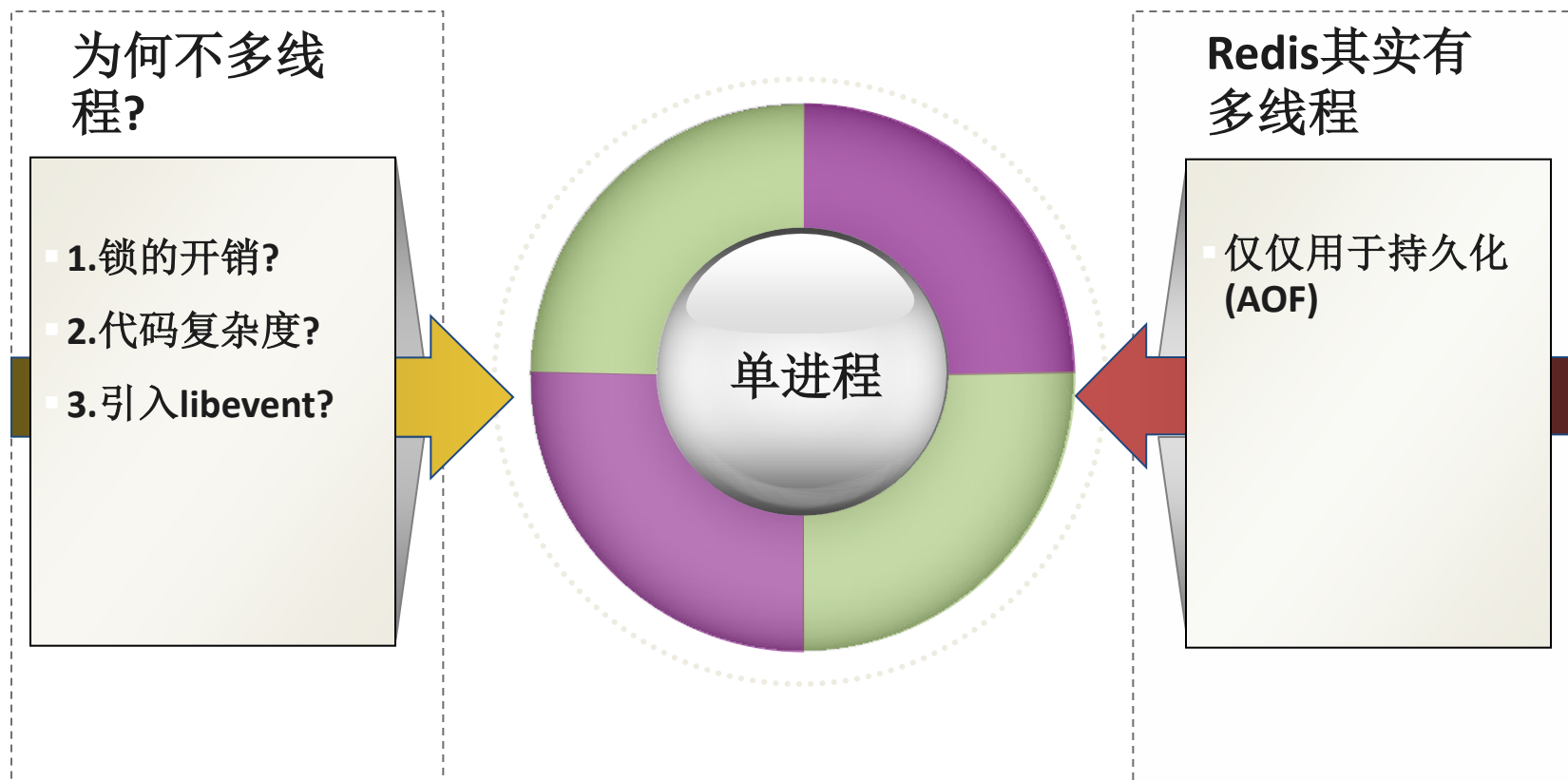
Redis特性(一)



Redis特性(二)



Redis特性(二)



Redis特性(三)

AOF & RDB

- 1.何为AOF
- 2.何为RDB

同步机制

- 1.异步同步

Redis命令

• Set, get, mget, mset 等

• Hset, hget, hgetall 等

是否对key设置TTL

Redis协议

**Example:
SET hello world**

```
*3\r\n$3\r\nSET\r\n$5\r\nHello\r\n$5\r\nWorld\r\n
```

+OK\r\n (状态回复)

:100\r\n (整形回复)

-ERR\r\n(错误回复)

**\$5\r\nHELLO\r\n(BULK
回复)**

Redis应用场景

1

QPS 高，数据库不能全部hold住

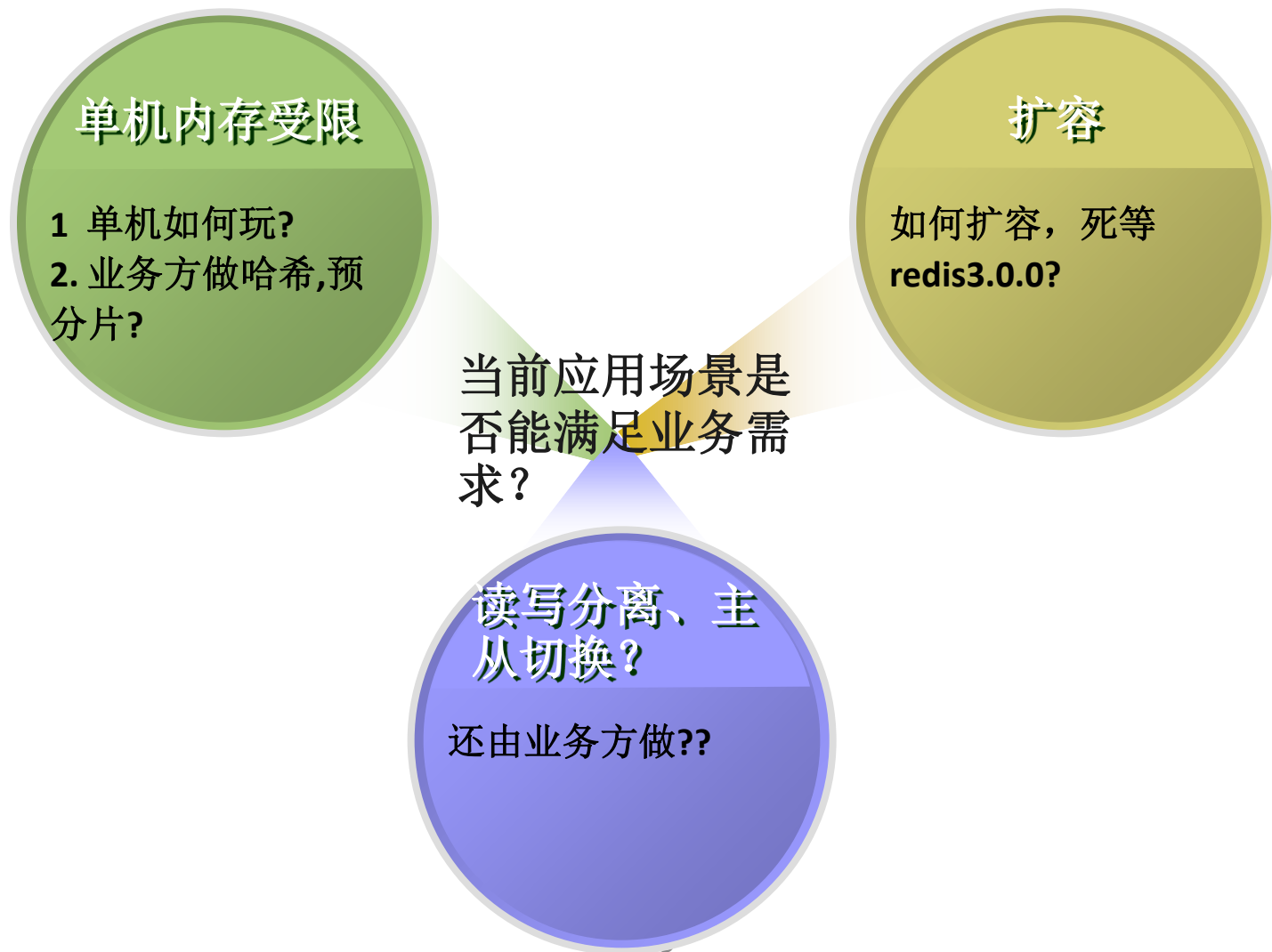
2

需要用到复杂数据结构

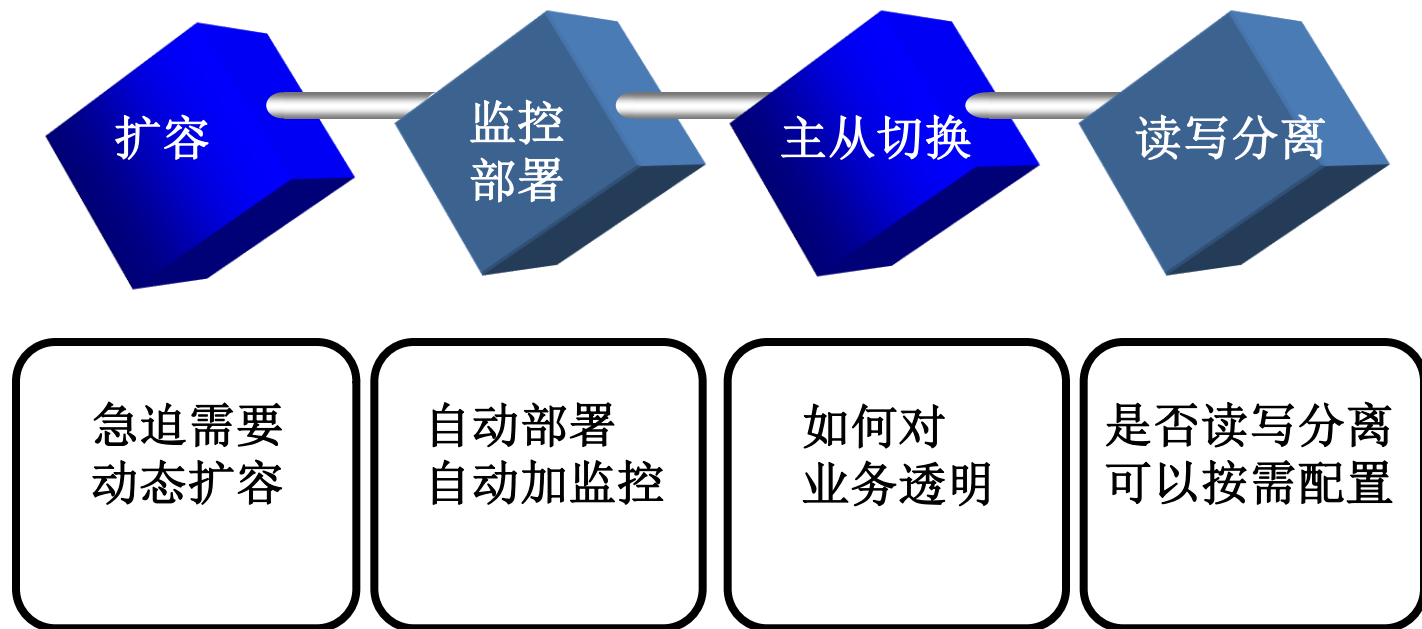
3

简易的消息推送
(发布订阅)

Question 1



引入BDRP



BDRP开发

自动化部署模块

A

Proxy模块

B

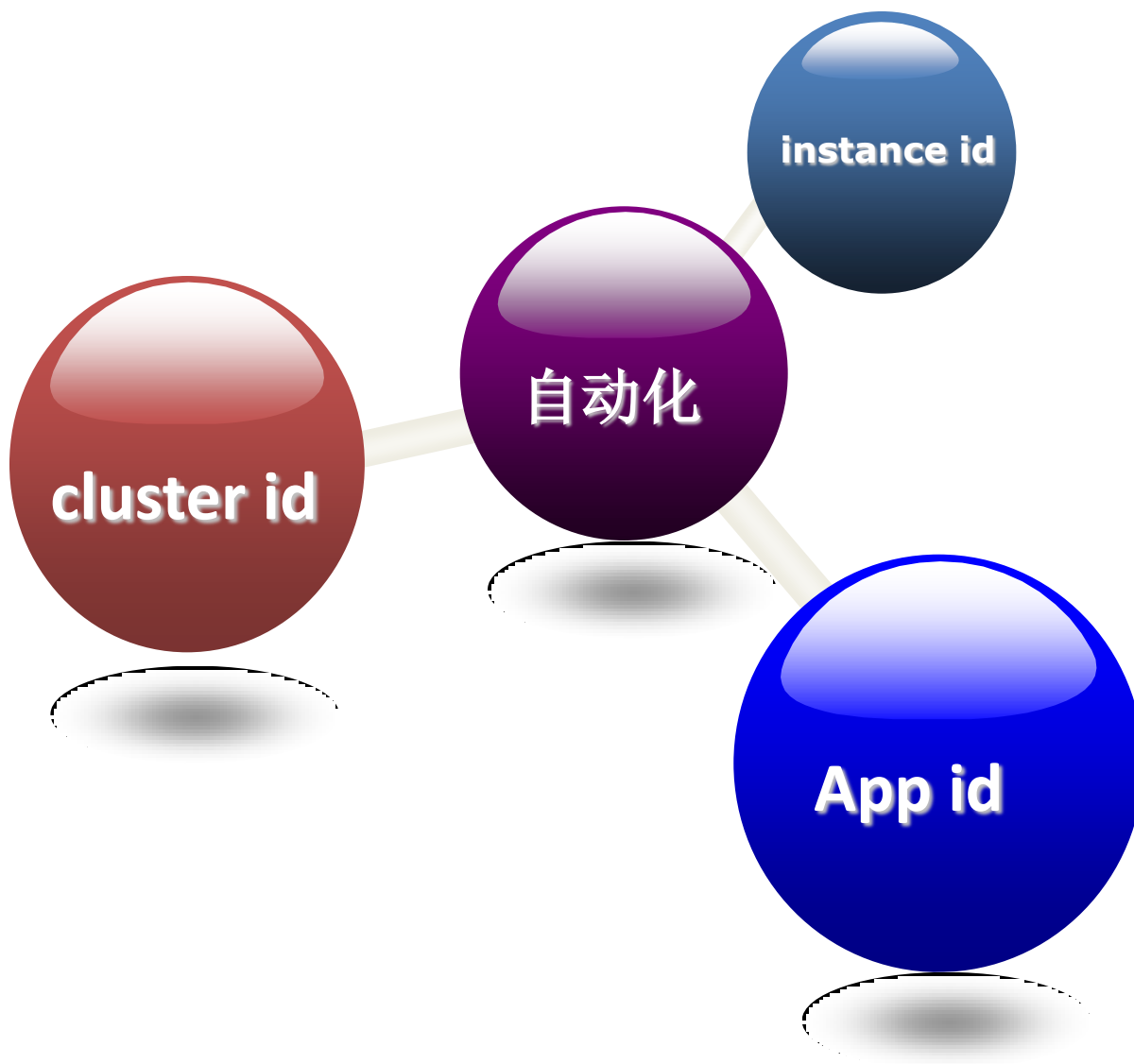
Redis-server实例

C

metaserver

D

自动化部署部署与监控的依赖



Proxy模块

- 注：基于twemproxy修改,增加以下功能
- 1. 读写分离可配置化
- 2. 连接数控制
- 3. 流量配额控制
- 4. server的配置变更无需重启proxy
- 5. 增加流量数据采集端口
- 6. 后端的redis slave宕掉以及发生主从切换，都无需人为手动修改配置

Proxy模块配置说明

- hash/distribution
- Auto_eject_hosts
- Client_connections
- Quota
- Flow_control
- Proxy_id
- Proxy_ins_id
- Server拓扑结构变化
- 白名单/权限控制

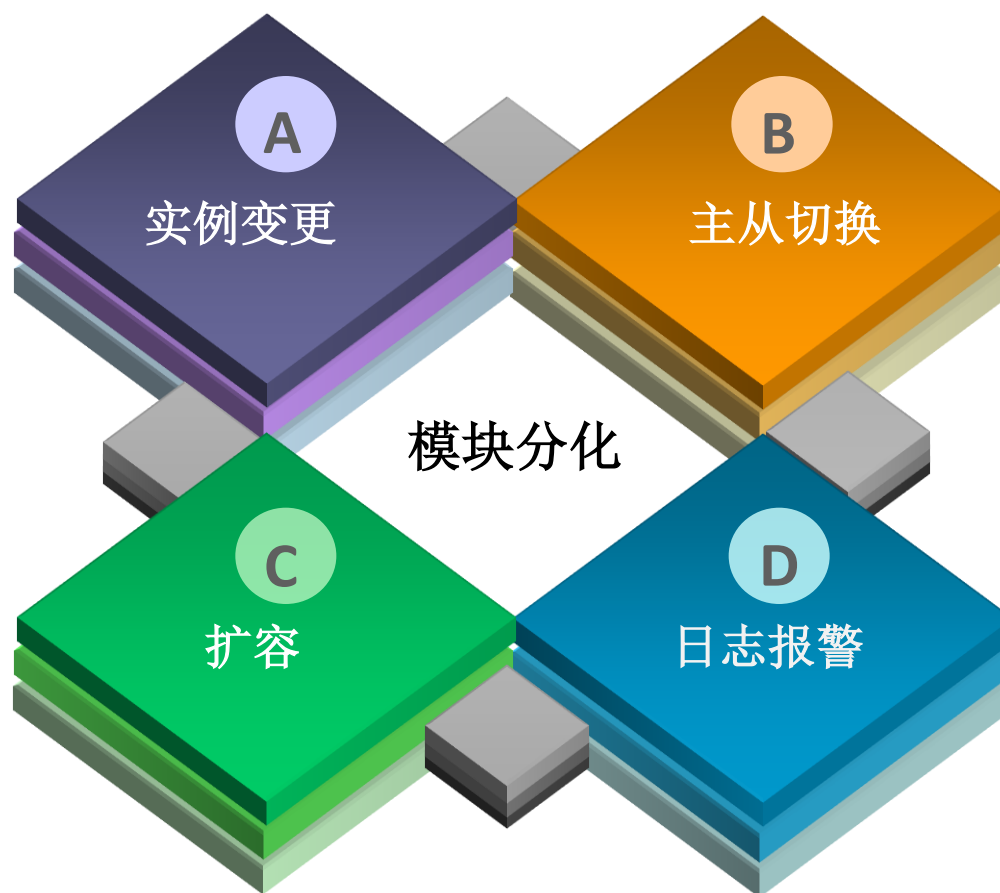
```
323:
  listen: 0.0.0.0:8333
  hash: crc16
  distribution: slot
  preconnect: true
  auto_eject_hosts: false
  redis: true
  backlog: 512
  client_connections: 5000
  server_connections: 1
  server_retry_timeout: 2000
  server_failure_limit: 2
  quota: 100
  flow_control: false
  proxy_id: 232
  proxy_ins_id: 458
  servers:
    - server1
      IP1:8333:1
      IP2:8333:1
      IP2:8333:1
~
```

Proxy配置变更与说明

- 1. 数据采集端口

```
{
  "proxy_qps":1, "proxy_read_qps":1, "proxy_write_qps":0, "323": {"client_eof":58, "cl
s":1, "real_read_qps":1, "real_write_qps":0, "consuming_token":0, "server_ejects":
rver_eof":0, "server_err":0, "server_timeout":0, "server_connections":3, "request
80, "response_bytes":2730237996, "in_queue":0, "in_queue_bytes":0, "out_queue":0,
```

metaserver



metaserver

- 1.实例变更（添加/删除）
- 新增Mpset,mpget,mpiset,mpiget,mriset等命令

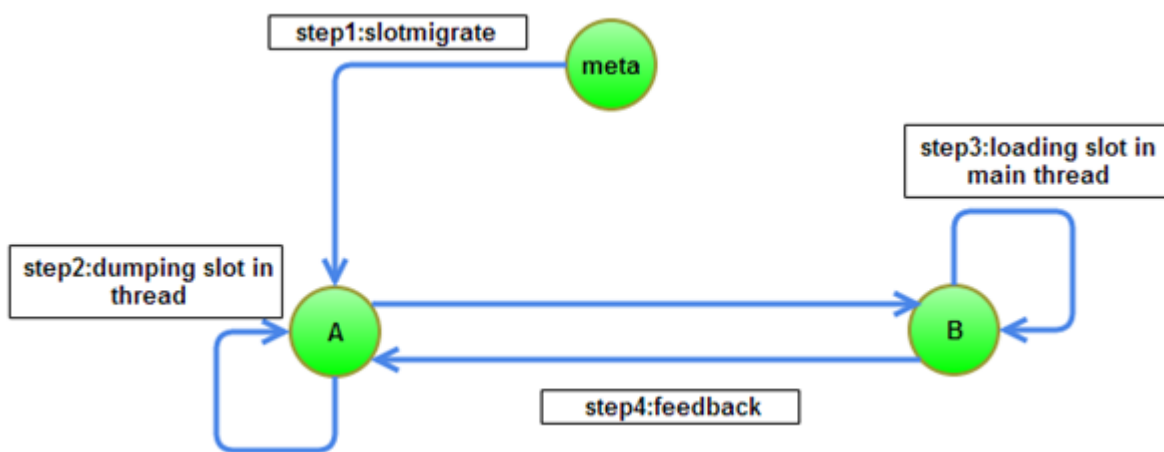
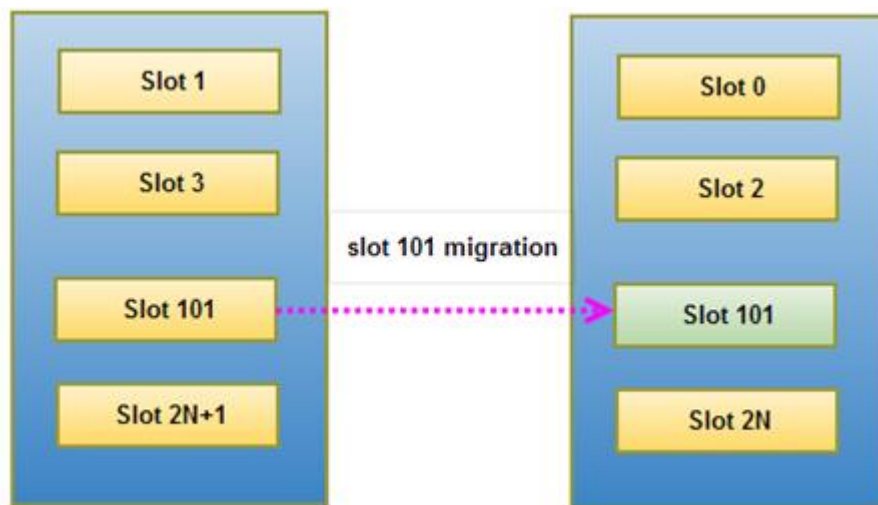
```
{ "mpset", proxySetCommand, 2, "wE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mpdel", proxyDelCommand, 2, "wE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mpget", proxyGetCommand, 2, "rE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mpiset", proxyInstSetCommand, 5, "wE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mpidel", proxyInstDelCommand, 2, "wE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mpiget", proxyInstGetCommand, 2, "rE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mriset", redisInstSetCommand, 8, "wE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mridel", redisInstDelCommand, 2, "wE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mriget", redisInstGetCommand, 2, "rE", 0, NULL, -1, 1, 1, 0, 0 },  
{ "mriupdate", redisInstUpdateCommand, 3, "wE", 0, NULL, -1, 1, 1, 0, 0 }
```


metaserver

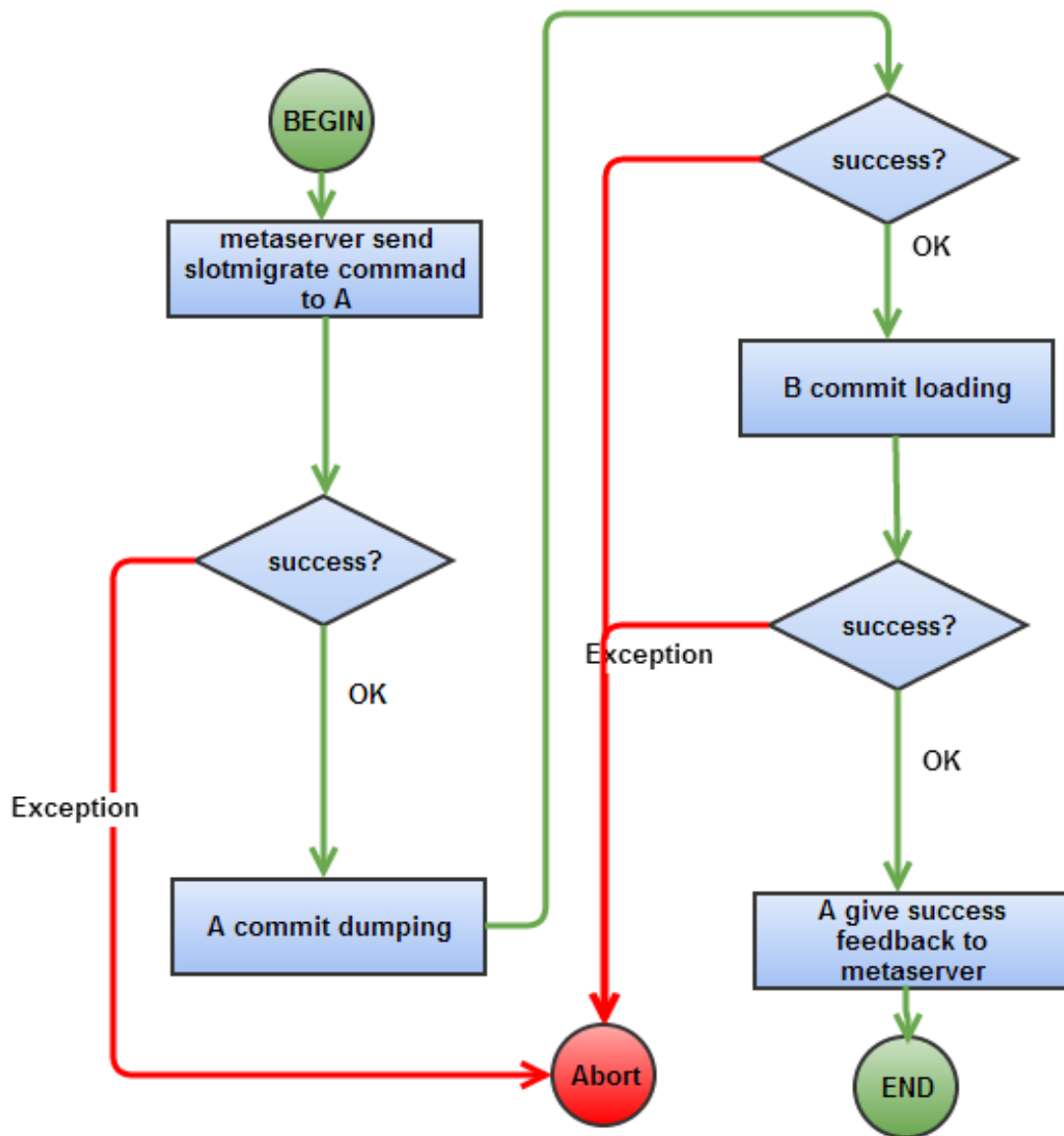
- 如何完成心跳检测、主从切换、扩容

```
void metaserverTimer() {  
    if (server.masterhost == NULL) {  
        /// master run timer  
        checkInstStatus();  
        if (metaserver.is_master_failover) {  
            checkMasterStatus();  
            proccessMasterdownTask();  
            processSlaveResyncTask();  
        }  
        proccessSlotMigrateTask();  
    }  
}
```

如何扩容(一)



如何扩容(二)



如何扩容(三)

- 与redis3.0.0 RC版本的有何区别?
- 1.migrate/dump command---
>slotdump/slotload command(响应速度更快，)
- 2.对业务方透明(迁移期间不影响读，写会失败)

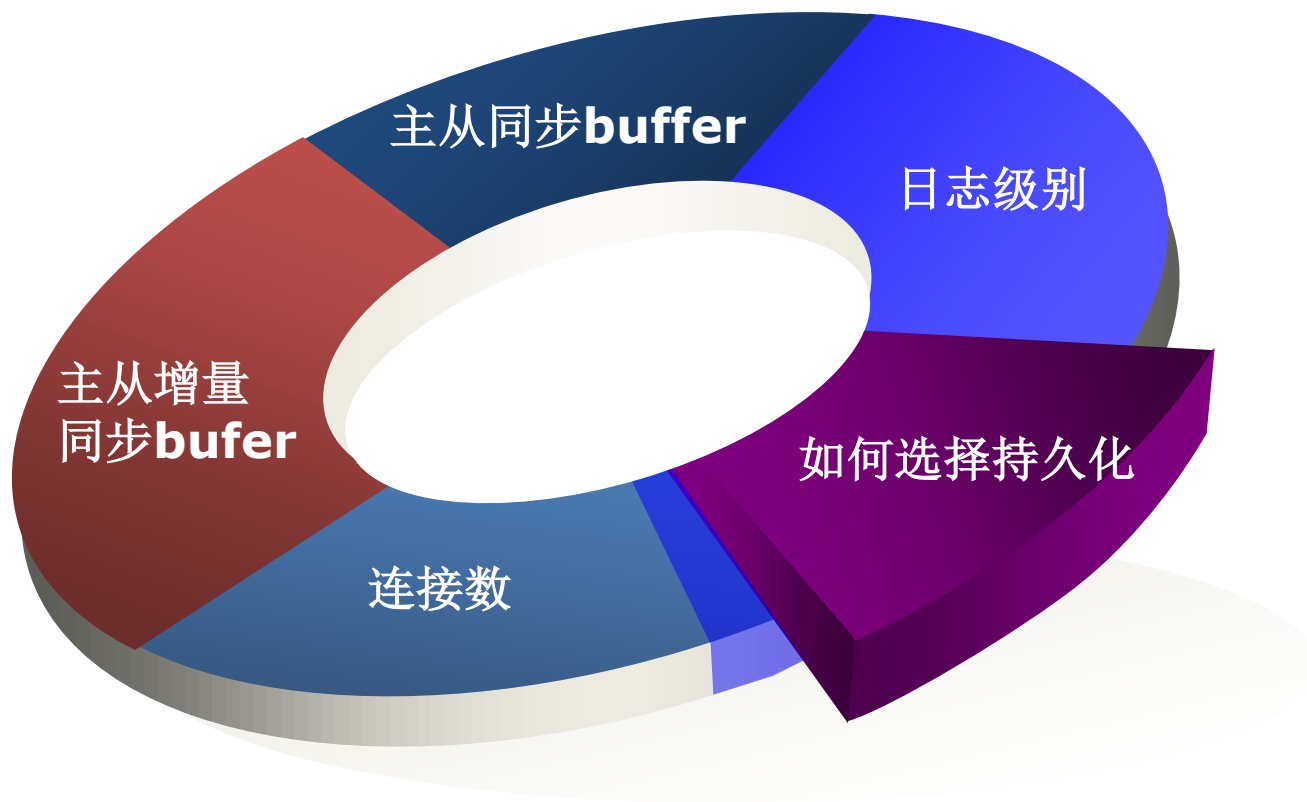
Metaserver 日志报警

- 1.发生主从切换[WARNING]
- 2.发生实例宕机[WARNING]
- 3.数据迁移失败[WARNING]
- 4.从metaserver宕机/假死[WARNING]

BDRP(redis以及proxy)运维

- 1. 小心redis以及proxy的配置
- 2. 关于redis命令的雷区
- 3. redis及proxy监控应该采集哪些项目
- 4. 关于redis运维简要准入说明
- 5. 关于系统参数的一些调优
- 6. 关于问题定位

小心redis的配置(一)



小心redis的配置(二)

- 1. daemonize and logfile
- 2.aof与rdb， 改如何选择
- 3. slave-serve-stale-data yes/no
- 4. maxclients N
- 5. client-output-buffer-limit slave 250mb
64mb 60
- 6. repl-backlog-size
- 7. maxmemory ? maxmemory-policy?

小心proxy(twemproxy)的配置

- 1. 启动参数
- `./bin/nutcracker -c ./conf/nutcracker.yml -w ./conf/whitelist -v 4 -A 10.202.6.87 -S 9000 -d -s 9066 -i 1000 -o ./log/nutcracker.log`
- 2. proxy运维注意事项
- 1. proxy存在OOM的可能性(极端case，高并发请求触发，导致proxy处理不过来，造成请求堆积)
- 2. proxy存在连接数打爆的case(社区版)

关于redis命令的雷区

- 1. 无论如何，K/V别太大（比如平均value有100K，QPS还非常高，redis的性能可能会大幅下降，受限于网卡以及单进程瓶颈），建议平均10K以下，业务上能压缩则压缩
- 2. 小心mget, del, hgetall, smembers等批量操作命令，可能会导致redis阻塞(过多的K/V组)，建议拆成多个分组
- 3. 持久化带来影响(fork的系统开销)?

Redis及proxy应该采集哪些监控项

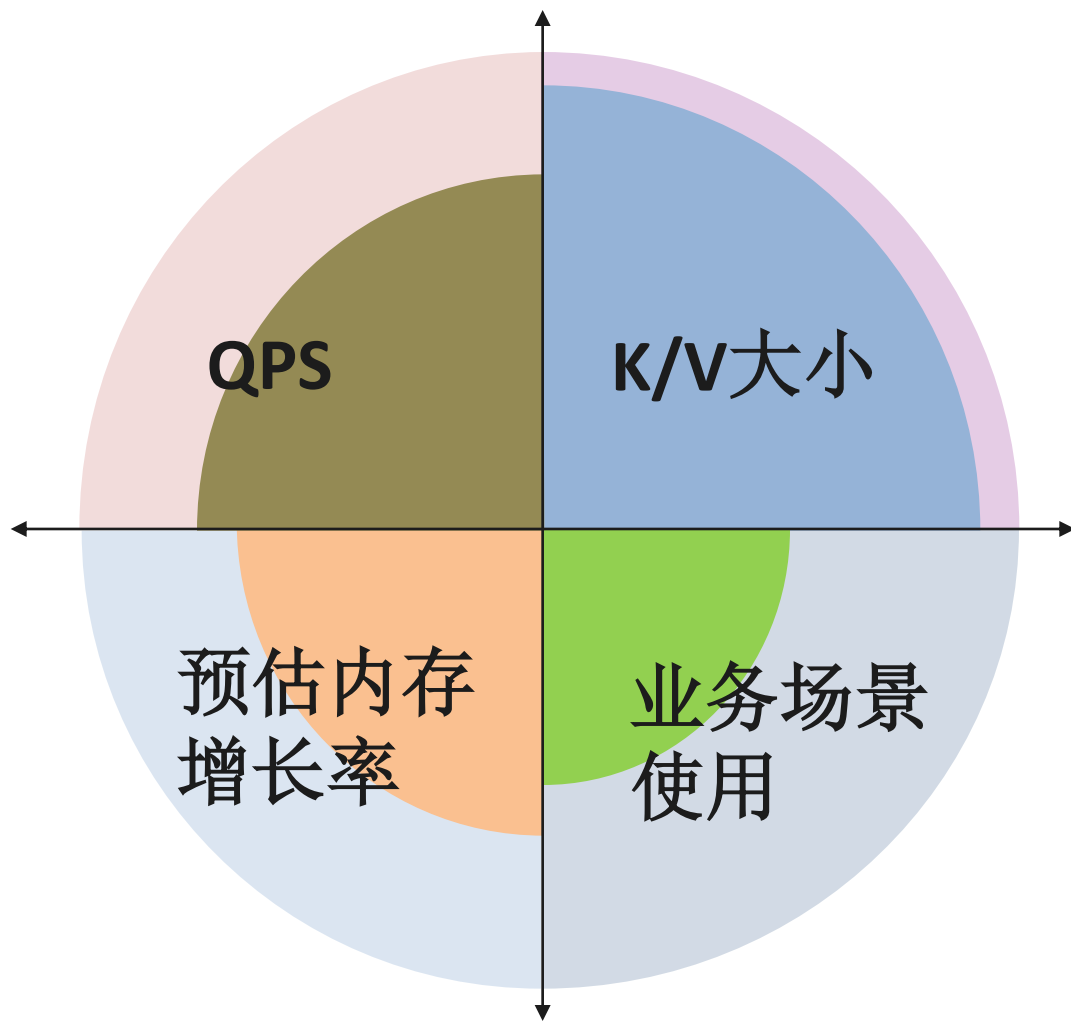
监控项

进程、端口

QPS、连接数

是否出现错误日志

Redis运维准入简要说明



简要的参数调优

- 1. 多队列网卡、Taskset?
- **ex: taskset -pc 1-5 process_pid**
- 2. /proc/sys/vm/overcommit_memory ?

关于问题定位

- 1.check监控，监控齐全，则定位更准
- 2. 慢查询？ Redis slowlog也许可以辅助定位，
- 也需要业务方check 请求K/V大小（K/V越大，长尾请求出现的概率越大）

存储选型?

www.eegee.com

草根交流，面对面

- Question1:Mysql, mongo, redis, ssdb?该如何选择?各有哪些优势?
- Question2:多家公司的redis集群对比

活动公众帐号eeqee_com

向上电话:

18010082538

13261951188

向上微信:

goodzjh

18010082538



Thank You !