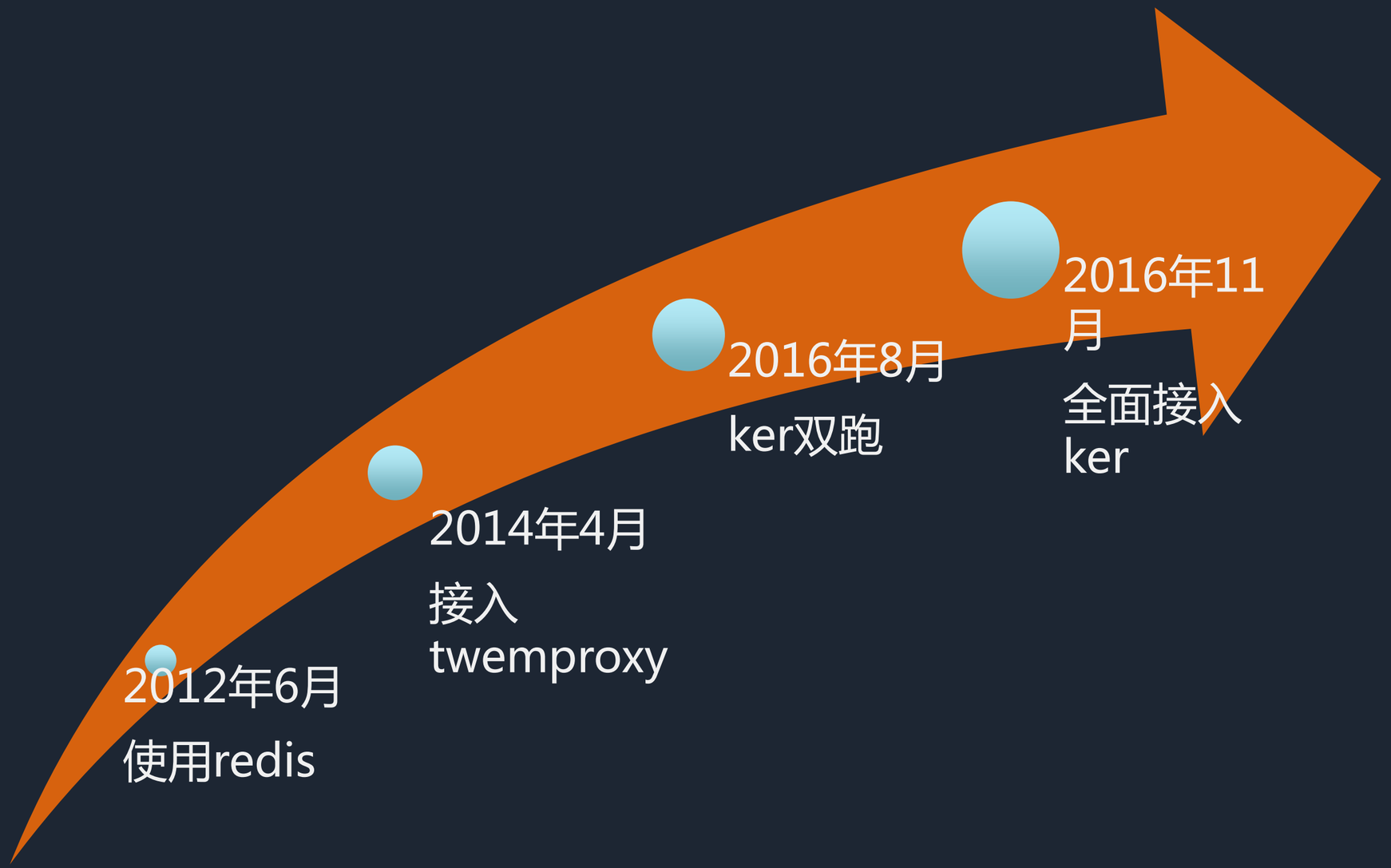
小米网redis服务平台建设历程

徐成选@小米

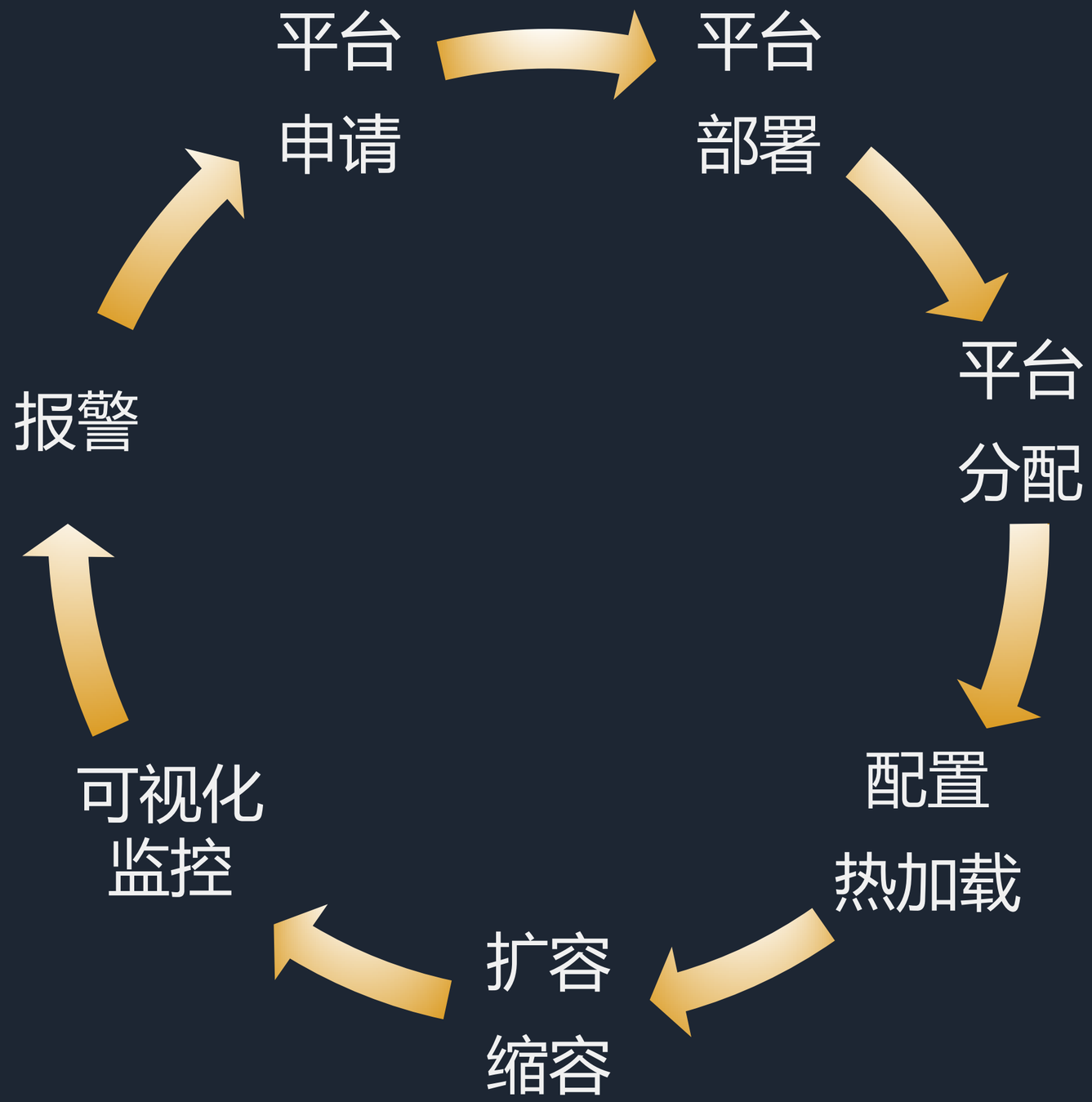
2017年8月

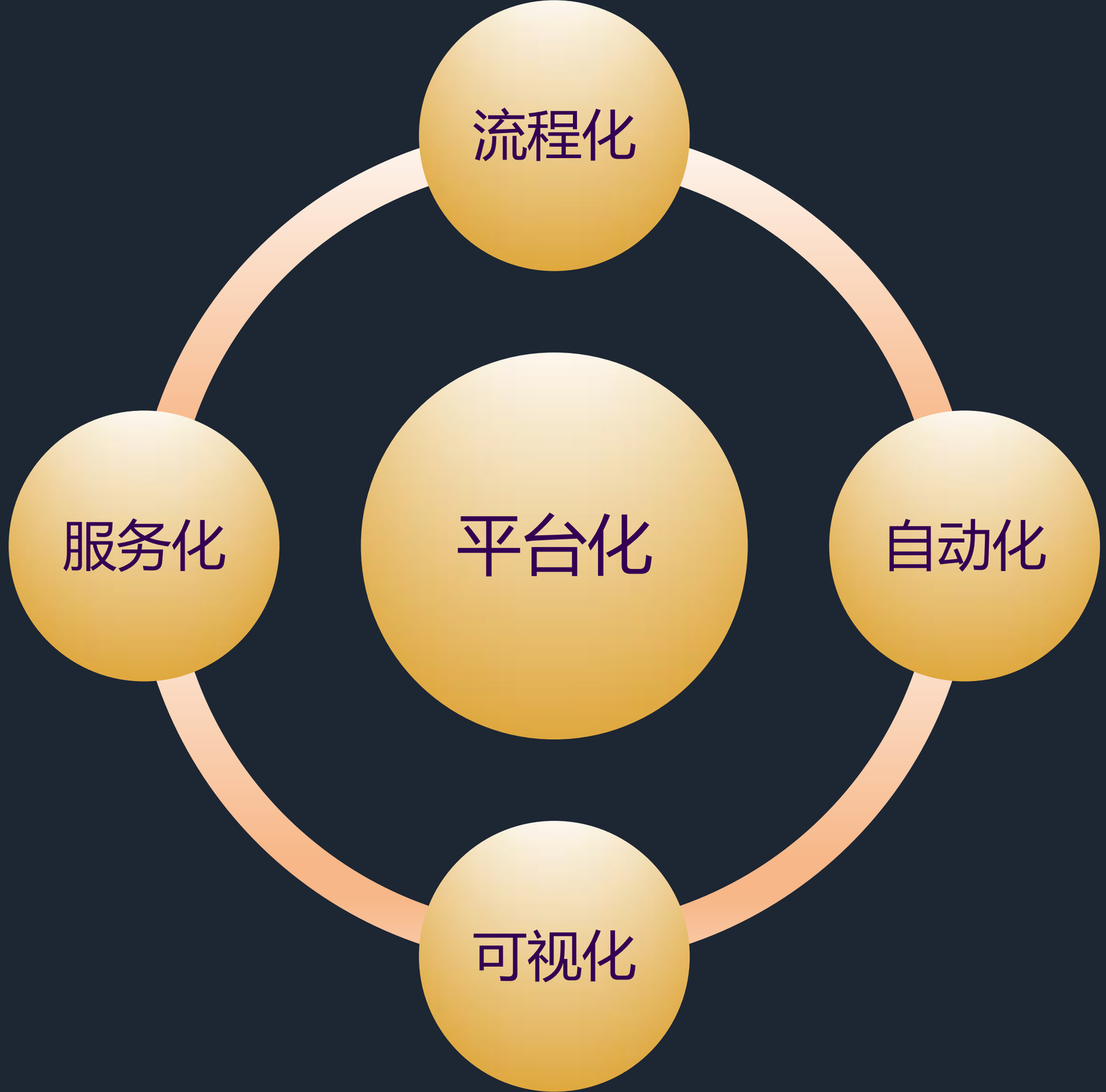
- twemproxy面临的问题
- KER平台设计考量
- KER平台架构
- 管理平台
- 监控与报警


2012年6月
使用redis
2014年4月
接入
twemproxy
2016年8月
ker双跑
2016年11
月
全面接入
ker


手工
运维
在线
排查
问题
修改配置
重启
问题
扩容
缩容
缺乏指标
数据
接入
成本高

- twemproxy面临的问题
- KER平台设计考量
- KER平台架构
- 管理平台
- 监控与报警





- twemproxy面临的问题
- KER平台设计考量
- KER平台架构
- 管理平台
- 监控与报警

KER迭代过程

KER1.0

- 兼容twemproxy
- 配置热加载
- 自助接入、平台运维
- 可视化监控

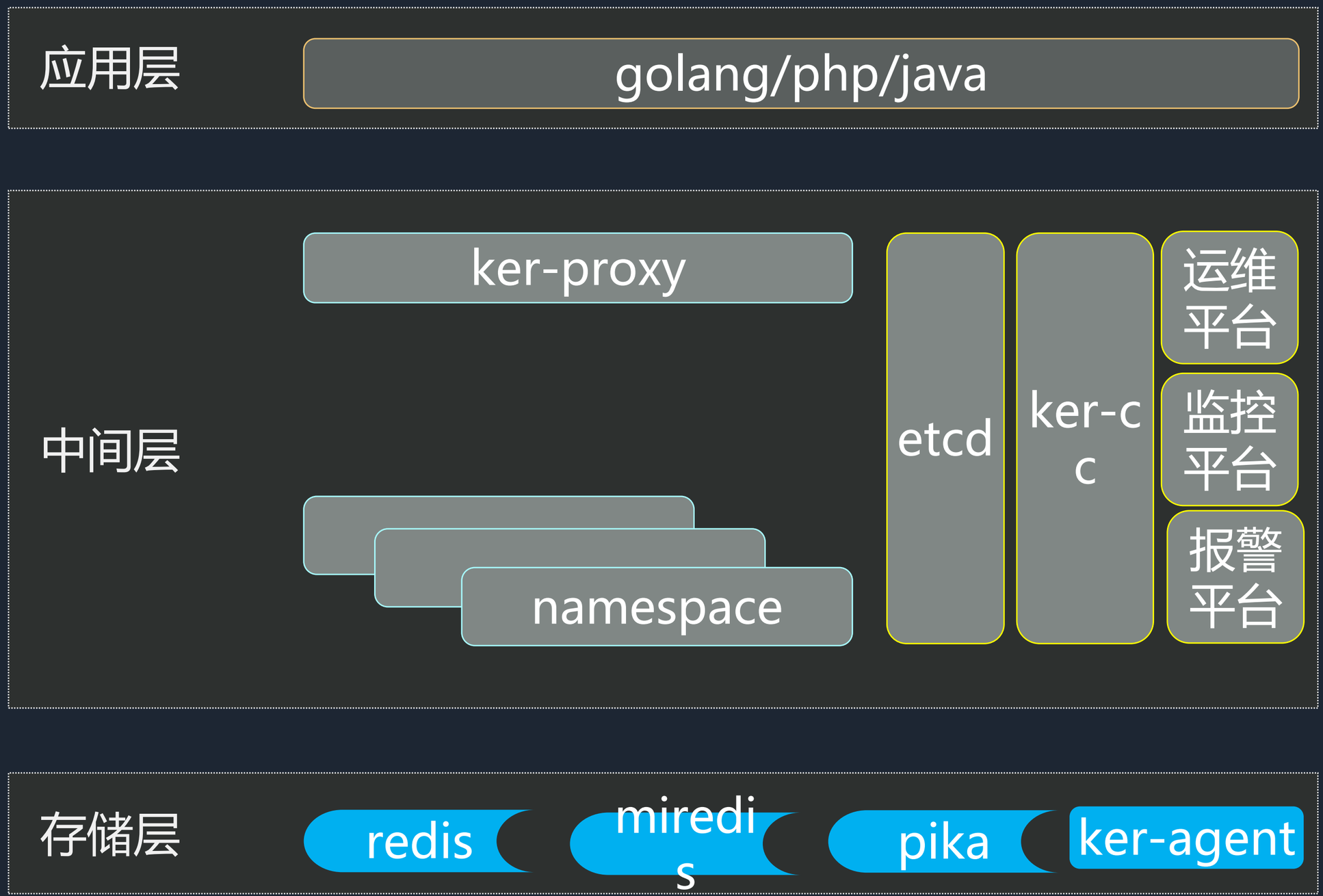
KER2.0

- 基于规则报警
- 动态扩缩容
- SSD
- 一键主从切换

KER3.0

- 自动化部署、配置
- 服务化

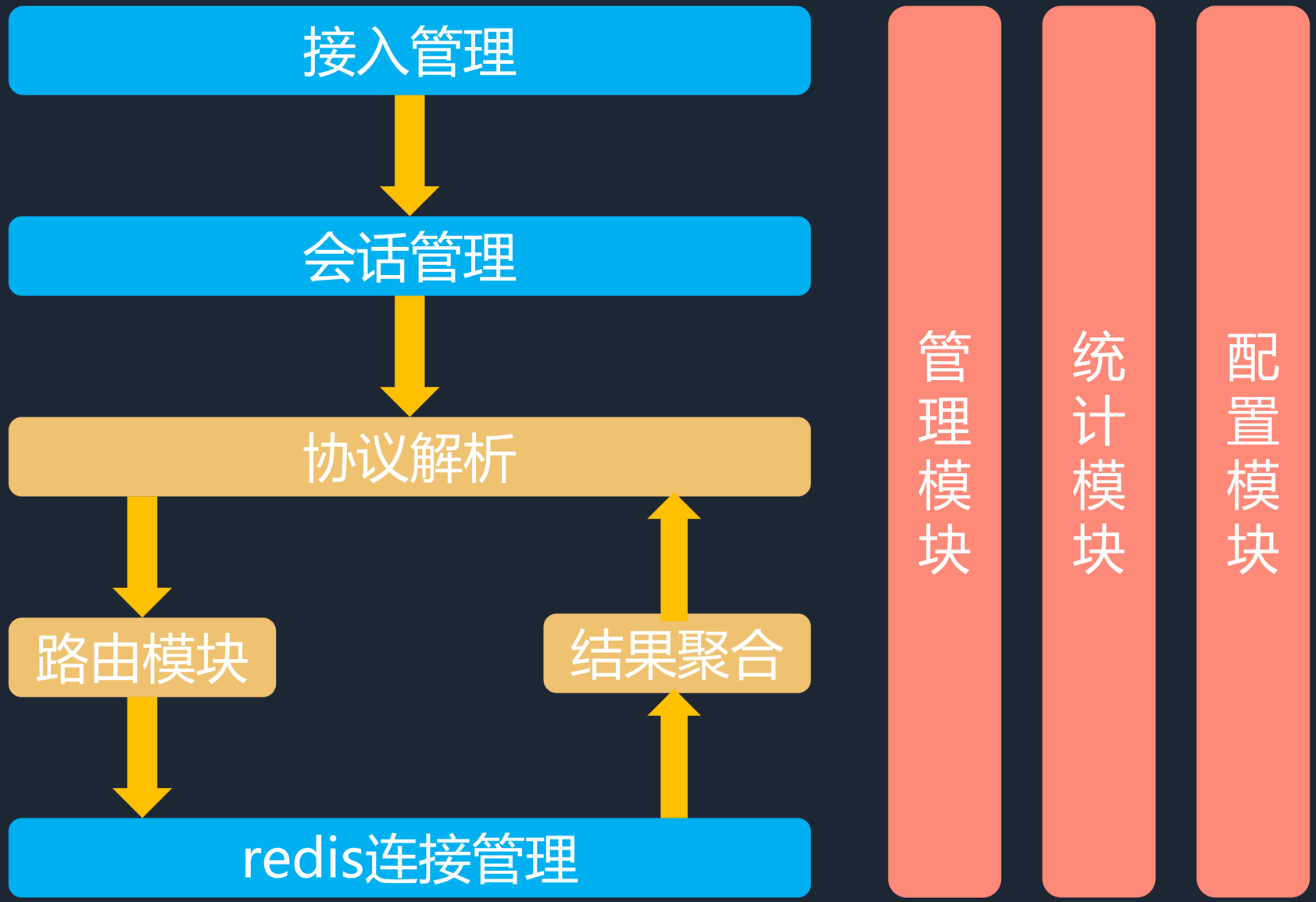
整体架构


应用层
golang/php/java
中间层
ker-proxy
namespace
etcd
ker-cc
运维平台
监控平台
报警平台
存储层
redis
miredis
pika
ker-agent

ker-proxy

只做最核心的功能

ker-proxy



ker-proxy

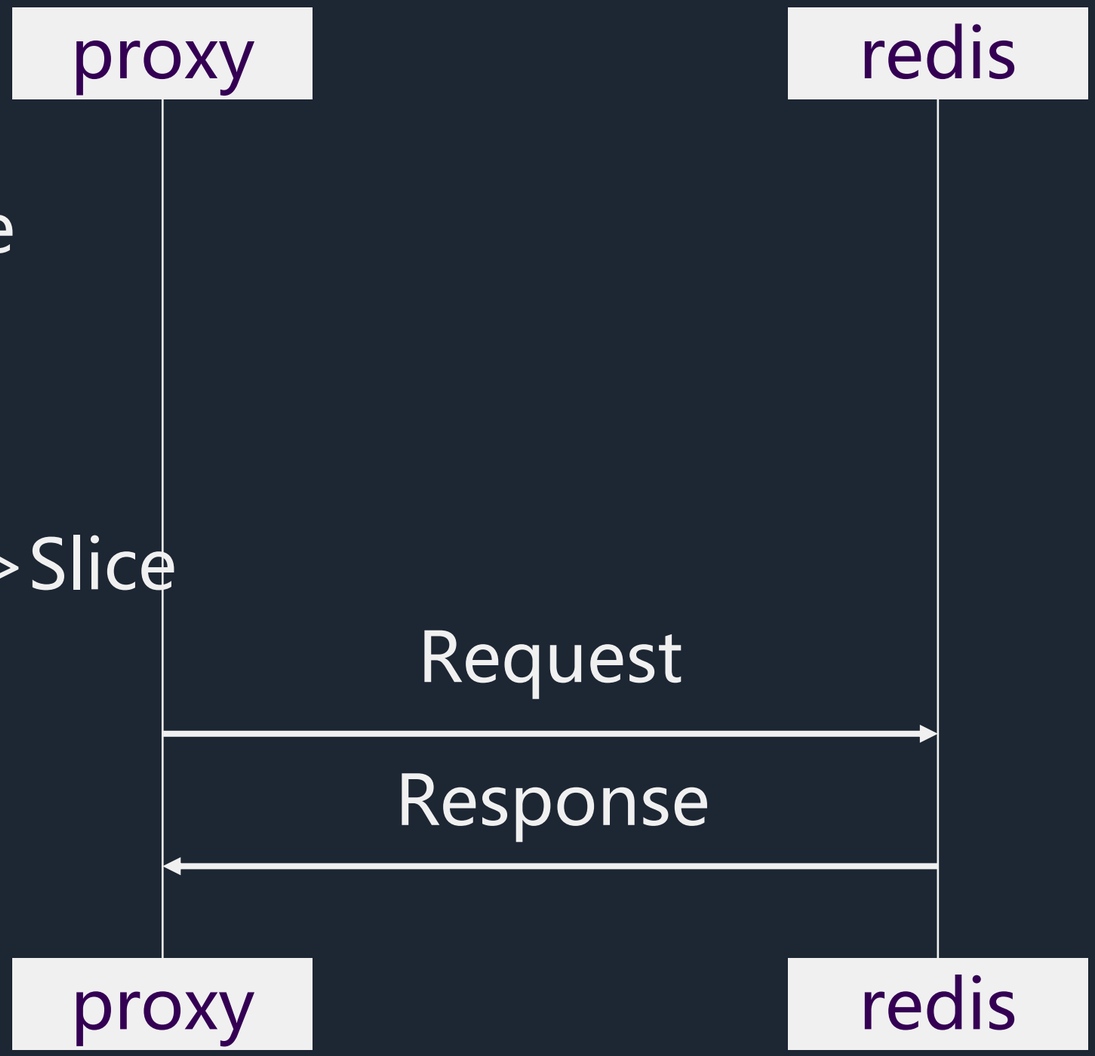

多端口
接入
会话
管理
读写
分离
访问
控制
命令
统计
多路由
转发
结果
合并
服务
化模
块
自定义
hyperlogl
og
后端连
接管理

ker-proxy处理流程

根据auth匹配namespace

Hash(Key)->Slice
Or
Hash(Key)->Slot、PreHash(Slot)->Slice

Slice将请求推送到主从



ker-cc

全局掌控，干所有脏活、累活

ker-cc

etcd

实例管理

配置管理

后台任务

ker-cc

ker-agent

物理机实例部署、配置、启停

配置热加载

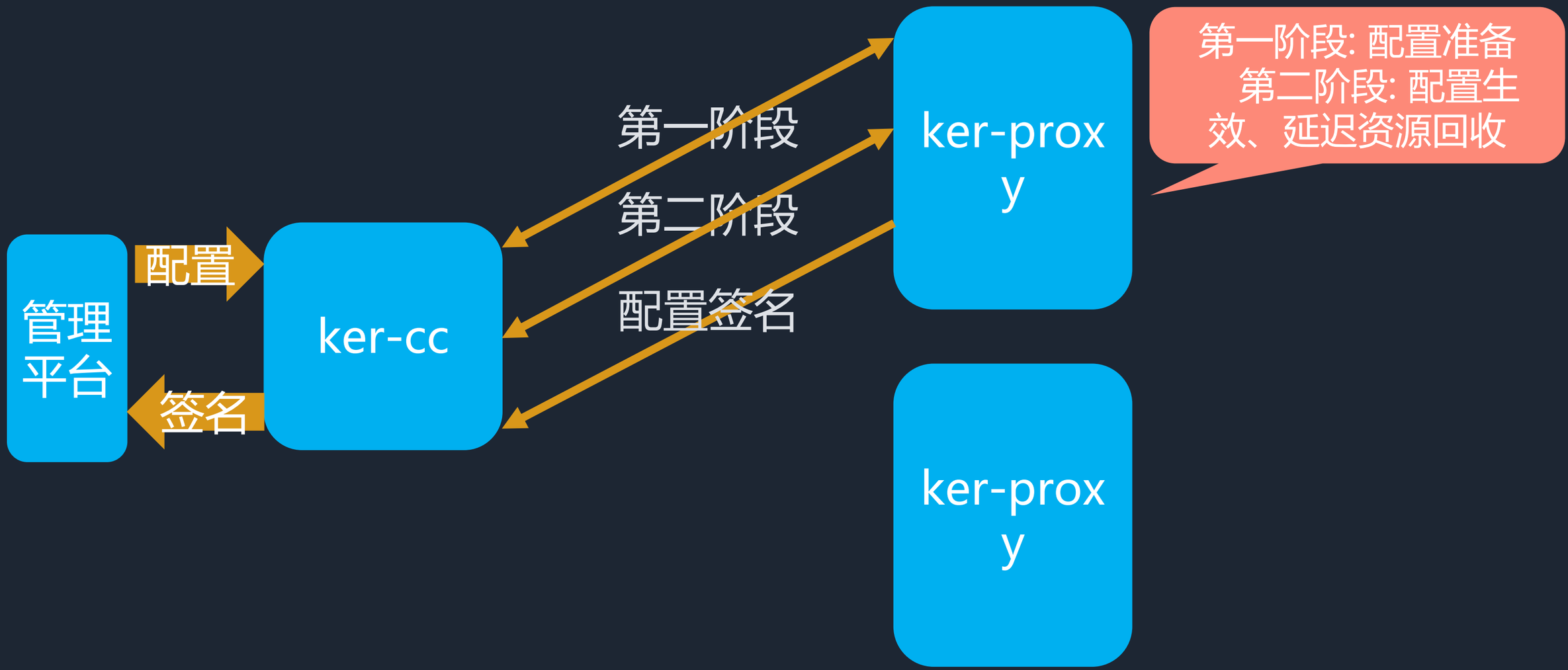
twemproxy

- 3000行的yaml文件
- 批量替换、批量重启
- 易出错

KER

- 配置按照namespace分别存储到mysql、etcdl、本地缓存文件
- ker-proxy上报当前配置签名，校验配置变更是否生效、一致
- ker-cc校验配置准确性，通过两阶段提交的方式变更ker-proxy配置
- COW策略
- 第一阶段全量复制配置
- 第二阶段以namespace为单位，生效配置、延迟回收动态资源

配置热加载


管理平台
配置
签名
ker-cc
第一阶段
第二阶段
配置签名
ker-proxy
第一阶段: 配置准备
第二阶段: 配置生效、延迟资源回收
ker-proxy

动态扩缩容

Slice路由

- 翻倍扩容
- 提供清理工具，线下清理脏数据

Slots路由

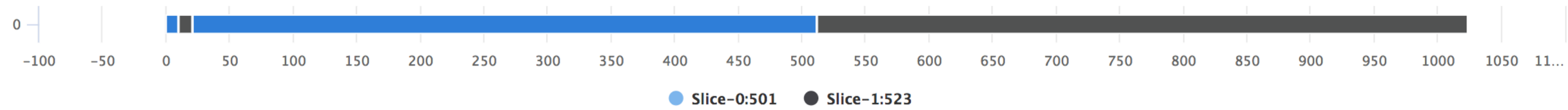
- 每个Namespace预分配1024个槽
  - DBA以槽为单位发起数据迁移
  - ker-cc以后台任务的方式不断迁移槽内数据
  - ker-proxy主动迁移槽内数据

动态扩缩容-平台

迁移Slots

设置参数 30 ~ 40 to 1 迁移Slots

Slots分布



Slots信息

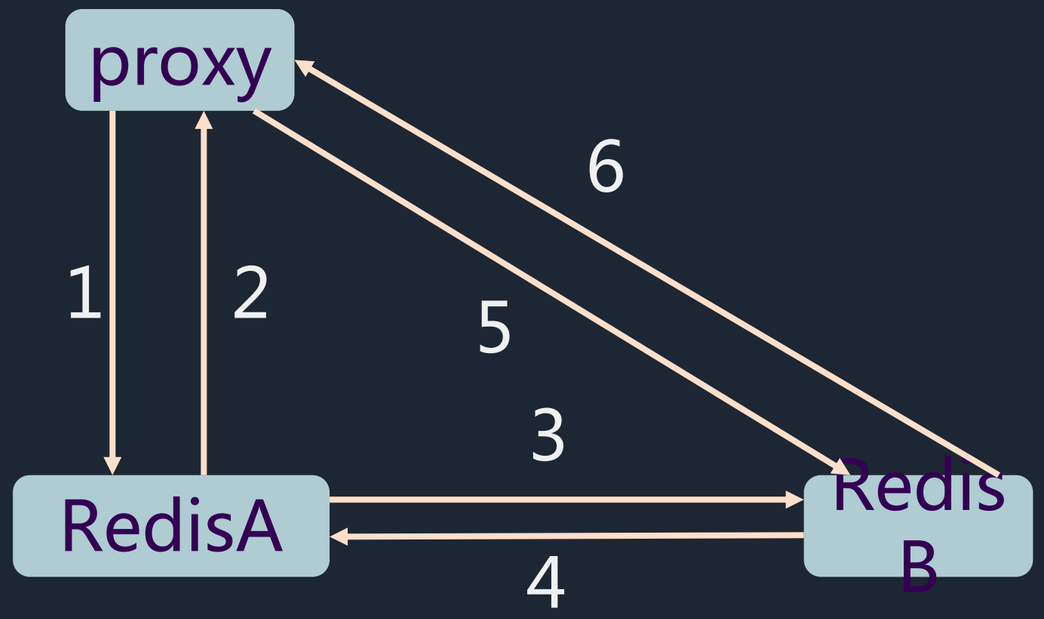
Show 20 entries Search:

SlotId	SliceId	State	TargetSliceId
0	0	-1	-1
1	0	-1	-1

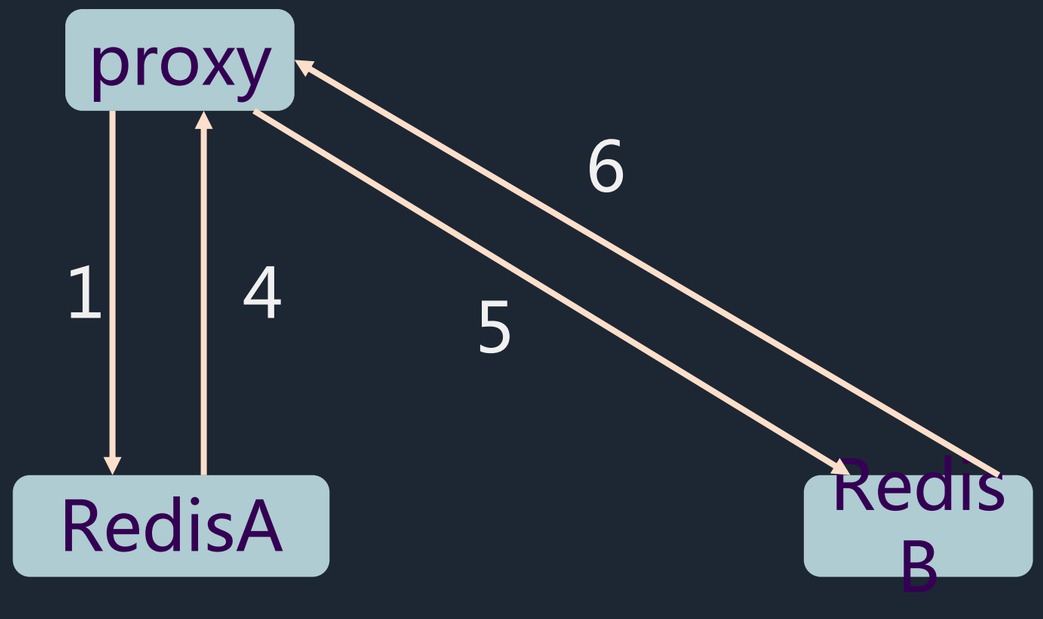
迁移信息

Slot	Slice	Target	Status
32	0	1	正在迁移
33	0	1	等待迁移
34	0	1	等待迁移

动态扩缩容-迁移



1 - SLOTSMGRTONE
2 - OK
3 - SLOTSRESTORE
4 - OK
5 - redis cmd
6 - OK



1 – SLOTSMGRTONE
4 - OK
5 - redis cmd
6 - OK

服务化

需求

- 基础服务接入微服务体系
- 应用系统跨过LVS直连ker-proxy

方案

- ker-proxy启动后注册信息到etcd、定时更新
- soa服务只需要配置auth即可通过xredis库访问redis
- 多个IP负载均衡、IP对应连接池延迟创建

服务化


```
//lvs
if p.kerDirect == 0 {
    reply, err = p.do(p.pool, cmd, args...)
    return
} else { //ker
    s, err := xconfig.GetService(KER_SERVICE)
    if err != nil {
        return reply, err
    }

    ip, port, err := xbalance.DoBalance(s)
    if err != nil {
        return reply, err
    }
```

```
p.kerLock.RLock()
pool, ok := p.kerPool[k]
if ok {
    p.kerLock.RUnlock()
    reply, err = p.do(pool, cmd, args...)
    return reply, err
}
p.kerLock.RUnlock()
p.kerLock.Lock()

pool, err = NewXRedisPool(p.name, ip, port, p.auth, uint(p.connTimeout),
    uint(p.readTimeout), uint(p.writeTimeout), p.maxOpen)
p.kerPool[k] = pool
p.kerLock.Unlock()
```

- twemproxy面临的问题
- KER平台设计考量
- KER平台架构
- 管理平台
- 监控与报警

管理平台-申请

我的业务

认领redis　申请

业务方接入Ker-Redis代理

服务（APP_ID)	请选择一个服务（app_id)
机房	请选择一个机房(cluster)
类型	缓存
是否读写分离	使用读写分离:性能高，但是牺牲一致性
命名空间*	命名空间　前缀:
所需存储空间	10 MB
一年后存储	20 MB
读 Qps	1000 次数
写 Qps	1000 次数
key-value对平均大小	100 Byte
申请描述信息	

管理平台-DBA分配

236	系统组	auth_test_one	金山云测试	js_systech_test_slots12	申请中	上线	存储	slots	miredis	是	yangsi	2017-06-30 16:56:37	dba管理 删除 下线 修改 查看
242	数据组	redis_cost_sharing	大陆测试	cn_midata_cost_sharing	申请中	上线	缓存	slots	miredis	是	liuyifan3	2017-07-14 16:33:28	dba管理 删除 下线 修改 查看
248	门店组	xmstore_it	金山云测试	js_xmstore_test	申请中		缓存	slots	miredis	是	hanliguo	2017-07-27 11:39:47	dba管理 删除 下线 修改 查看
3	运维组	redis_test.damn	大陆测试	redis_test.damn	申请完成	上线	缓存	slice		是	zhanghua	2016-11-10 09:57:13	dba管理 删除 下线 修改 查看
4	运维组	redis_test.damn	大陆测试	cn_redis_test.damn	申请完成	上线	缓存	slice		是	zhangxionghu	2016-10-12 23:14:54	dba管理 删除 下线 修改 查看
6	系统组	ker_redis_all_test	大陆测试	xm_stock_aws	申请完成	上线	缓存	slice		是	zhangxionghu	2016-10-17 09:35:23	dba管理 删除 下线 修改 查看

服务 （APP_ID）	xuchengxuan_test_hello---系统组
机房	大陆测试
管理用户	xuchengxuan---徐成选
类型	缓存
所需存储空间	100 MB

保存到db　添加slice文本　添加slice列表　重置表单　发送到ker中控

命名空间	cn_systech_cn_systech_bigpyer_test_1
Proxy Auth	cn_systech_cn_systech_bigpyer_test_1_1vwwVfn7
Redis Auth	6621

redis代理-管理信息

字段	命名空间	ProxyAuth	RedisAuth	HashTag	Readonly	读写分离
DB	cn_systech_cn_systech_bigpyer_test_1	cn_systech_cn_systech_bigpyer_test_1_1vwwVfn7nbp1N	6621		否	是
REMOTE	cn_systech_cn_systech_bigpyer_test_1	cn_systech_cn_systech_bigpyer_test_1_1vwwVfn7nbp1N	6621		否	是

redis代理-Slice列表

字段	slice_id	master	slave	slots	操作
DB	0	10.236.120.9:7480		[0,9] [21,29] [41,511]	修改 切换 配置 查看
REMOTE		10.236.120.9:7480			
DB	1	10.236.120.9:8480		[10,20] [30,40] [512,1023]	修改 切换 配置 查看
REMOTE		10.236.120.9:8480			

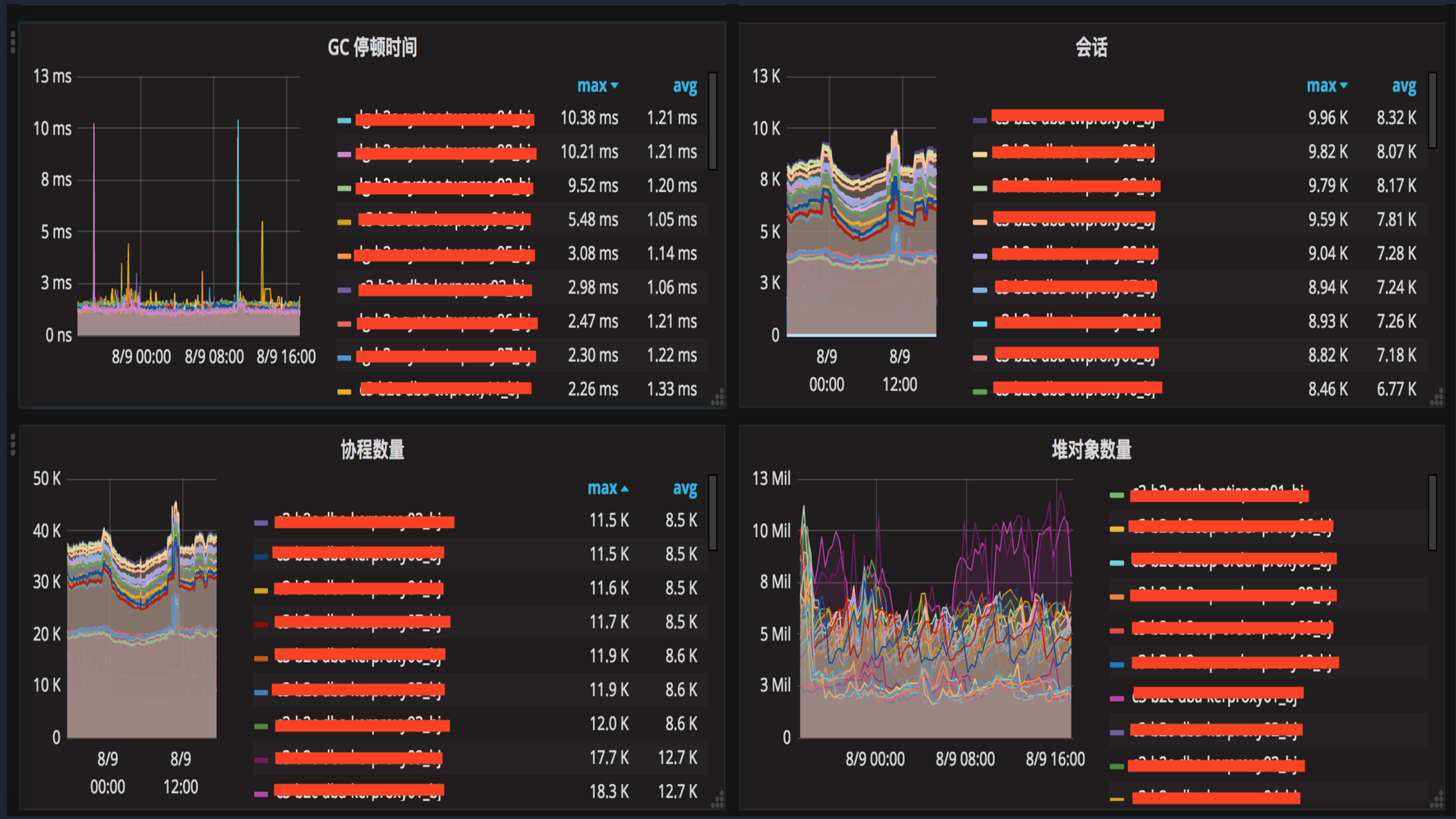
管理平台

- twemproxy面临的问题
- KER平台设计考量
- KER平台架构
- 管理平台
- 监控与报警

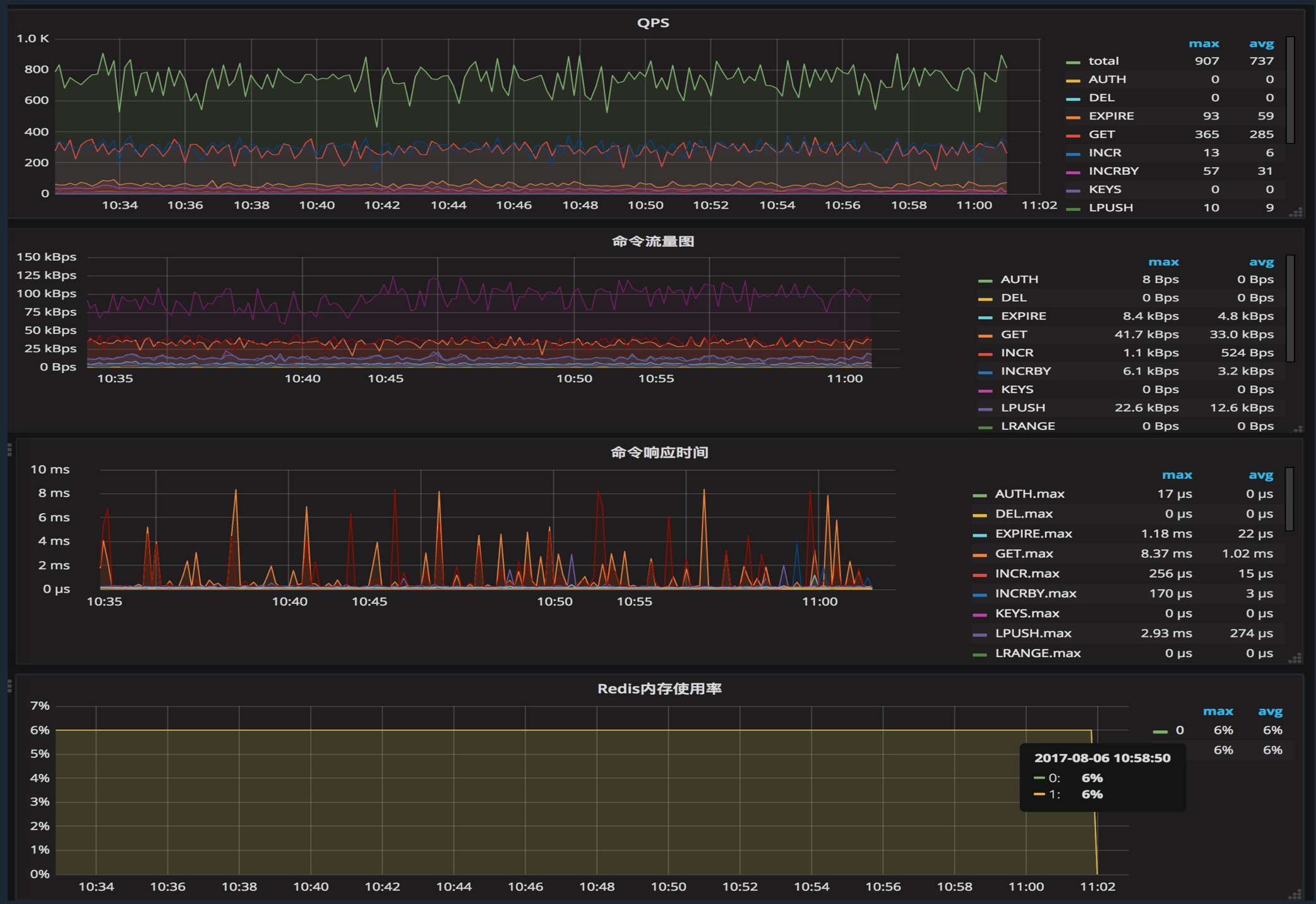
监控大盘


ker-proxy
Zoom Out
Last 30 minutes
集群 MAX QPS
733.74 K
集群网络负载MAX
276.4 MBps
集群平均响应时间
245 µs
业务QPS
800 K
600 K
400 K
200 K
0
10:30:35:40:45:50:55
max
avg
总 QPS 734 K 639 K
[redacted] 232 K 165 K
[redacted] 178 K 132 K
[redacted] 81 K 79 K
[redacted] 58 K 54 K
[redacted] 56 K 17 K
[redacted] 32 K 26 K
[redacted] 32 K 23 K
[redacted] 25 K 7 K
业务内存使用情况
200 GB
150 GB
100 GB
50 GB
0 B
[redacted] 174.7 GB 174.7 GB
[redacted] 124.3 GB 124.0 GB
[redacted] 66.3 GB 66.3 GB
[redacted] 32.0 GB 32.0 GB
[redacted] 31.6 GB 31.6 GB
[redacted] 31.1 GB 31.1 GB
[redacted] 30.1 GB 30.1 GB
[redacted] 25.0 GB 24.9 GB
[redacted] 24.0 GB 24.0 GB
后端 QPS
400 K
300 K
200 K
100 K
max
avg
[redacted]
业务 流量
300 MBps
250 MBps
200 MBps
150 MBps
100 MBps
50 MBps
max
avg
总流量 294.1 MBps 231.9 MBps
[redacted] 59.2 MBps 35.6 MBps
[redacted] 50.4 MBps 22.4 MBps
[redacted] 40.1 MBps 23.2 MBps
[redacted] 33.5 MBps 10.9 MBps
[redacted] 29.0 MBps 16.5 MBps
[redacted] 21.1 MBps 19.7 MBps

监控大盘


GC 停顿时间
13 ms
10 ms
8 ms
5 ms
3 ms
0 ns
8/9 00:00
8/9 08:00
8/9 16:00
max
avg
10.38 ms 1.21 ms
10.21 ms 1.21 ms
9.52 ms 1.20 ms
5.48 ms 1.05 ms
3.08 ms 1.14 ms
2.98 ms 1.06 ms
2.47 ms 1.21 ms
2.30 ms 1.22 ms
2.26 ms 1.33 ms
会话
13 K
10 K
8 K
5 K
3 K
0
8/9 00:00
8/9 12:00
max
avg
9.96 K 8.32 K
9.82 K 8.07 K
9.79 K 8.17 K
9.59 K 7.81 K
9.04 K 7.28 K
8.94 K 7.24 K
8.93 K 7.26 K
8.82 K 7.18 K
8.46 K 6.77 K
协程数量
50 K
40 K
30 K
20 K
10 K
0
8/9 00:00
8/9 12:00
max
avg
11.5 K 8.5 K
11.5 K 8.5 K
11.6 K 8.5 K
11.7 K 8.5 K
11.9 K 8.6 K
11.9 K 8.6 K
12.0 K 8.6 K
17.7 K 12.7 K
18.3 K 12.7 K
堆对象数量
13 Mil
10 Mil
8 Mil
5 Mil
3 Mil
0
8/9 00:00
8/9 08:00
8/9 16:00

监控-某业务


QPS
1.0 K
800
600
400
200
0
10:34
10:36
10:38
10:40
10:42
10:44
10:46
10:48
10:50
10:52
10:54
10:56
10:58
11:00
11:02
max
avg
total 907 737
AUTH 0 0
DEL 0 0
EXPIRE 93 59
GET 365 285
INCR 13 6
INCRBY 57 31
KEYS 0 0
LPUSH 10 9
命令流量图
150 kBps
125 kBps
100 kBps
75 kBps
50 kBps
25 kBps
0 Bps
10:35
10:40
10:45
10:50
10:55
11:00
max
avg
AUTH 8 Bps 0 Bps
DEL 0 Bps 0 Bps
EXPIRE 8.4 kBps 4.8 kBps
GET 41.7 kBps 33.0 kBps
INCR 1.1 kBps 524 Bps
INCRBY 6.1 kBps 3.2 kBps
KEYS 0 Bps 0 Bps
LPUSH 22.6 kBps 12.6 kBps
LRANGE 0 Bps 0 Bps
命令响应时间
10 ms
8 ms
6 ms
4 ms
2 ms
0 µs
10:35
10:40
10:45
10:50
10:55
11:00
max
avg
AUTH.max 17 µs 0 µs
DEL.max 0 µs 0 µs
EXPIRE.max 1.18 ms 22 µs
GET.max 8.37 ms 1.02 ms
INCR.max 256 µs 15 µs
INCRBY.max 170 µs 3 µs
KEYS.max 0 µs 0 µs
LPUSH.max 2.93 ms 274 µs
LRANGE.max 0 µs 0 µs
Redis内存使用率
7%
6%
5%
4%
3%
2%
1%
0%
10:34
10:36
10:38
10:40
10:42
10:44
10:46
10:48
10:50
10:52
10:54
10:56
10:58
11:00
11:02
max
avg
0 6% 6%
6% 6%
2017-08-06 10:58:50
0: 6%
1: 6%

报警

规则ID	规则名称	数据源	Topic	Grafana名称	Grafana图表	合并策略	发送策略	优先级	逐级上报	状态	操作
1	内存阈值-xm_stock-[redacted]	KAFKA	b2c_ker_cc_applogs_error			endpoint_metric	notify1	P2	否	启用	禁用 修改 删除 异常信息
3	实例存活检测-proxy-[redacted]	KAFKA	b2c_ker_cc_applogs_error			endpoint_metric_priority	notify7	P0	否	启用	禁用 修改 删除 异常信息

数据源: KAFKA

Topic: b2c_ker_cc_applogs_error

合并策略: endpoint_metric

发送策略: notify1

优先级: P2

报警接收人: × xuchengxuan---徐成选

逐级上报: 否

+ 添加规则

Namespace	=(字符串)	xm_stock	
and	Host	=(字符串)	[redacted]
and	MemRate	>=(数字)	98

元字段: × Namespace=xm_stock × Host=[redacted]

备注: 内存阈值-xm_stock-[redacted]

提交 重置

商城系统组

- 分布式缓存
- 消息中间件
- 数据库中间件
- 微服务平台、框架
- 日志处理、挖掘
- 监控、报警等

Thanks