



# RocketMQ运维分享

数列科技 – 平威 2018



## 陆学慧 (平威)

杭州数列联合创始人&CTO

分布式技术爱好者，曾主导多个核心中间件产品的诞生和研发。

\* 3年分布式服务框架经验HSF/Dubbo

\* 2年分布式消息经验MetaQ/RocketMQ

\* 2年软负载和配置中心经验ConfigServer/Diamond/VipServer

业余爱好：乒乓球、路亚、王者荣耀

## 概览:

01



RocketMQ 简介

02



RocketMQ 模块介绍

03



RocketMQ 日常运维

04



RocketMQ 运维建议

01

RocketMQ 简介

## RocketMQ介绍

RocketMQ是企业级互联网架构的核心产品，服务于整个阿里巴巴集团6年时间，经过阿里巴巴交易核心链路反复打磨与理念双十一的严苛考验，是一个真正具备**低延时**、**高并发**、**高可用**、**高可靠**，可制成万亿级数据洪峰的分布式消息中间件。



- 服务于阿里巴巴集团6年，经历5次双十一一年检
- 大量使用与阿里巴巴交易核心链路
- 支持海量高并发
- 支持万亿级消息流转
- 支持海量消息堆积
- 支持高可靠、高可用方案

## 02

### ■ RocketMQ 模块介绍

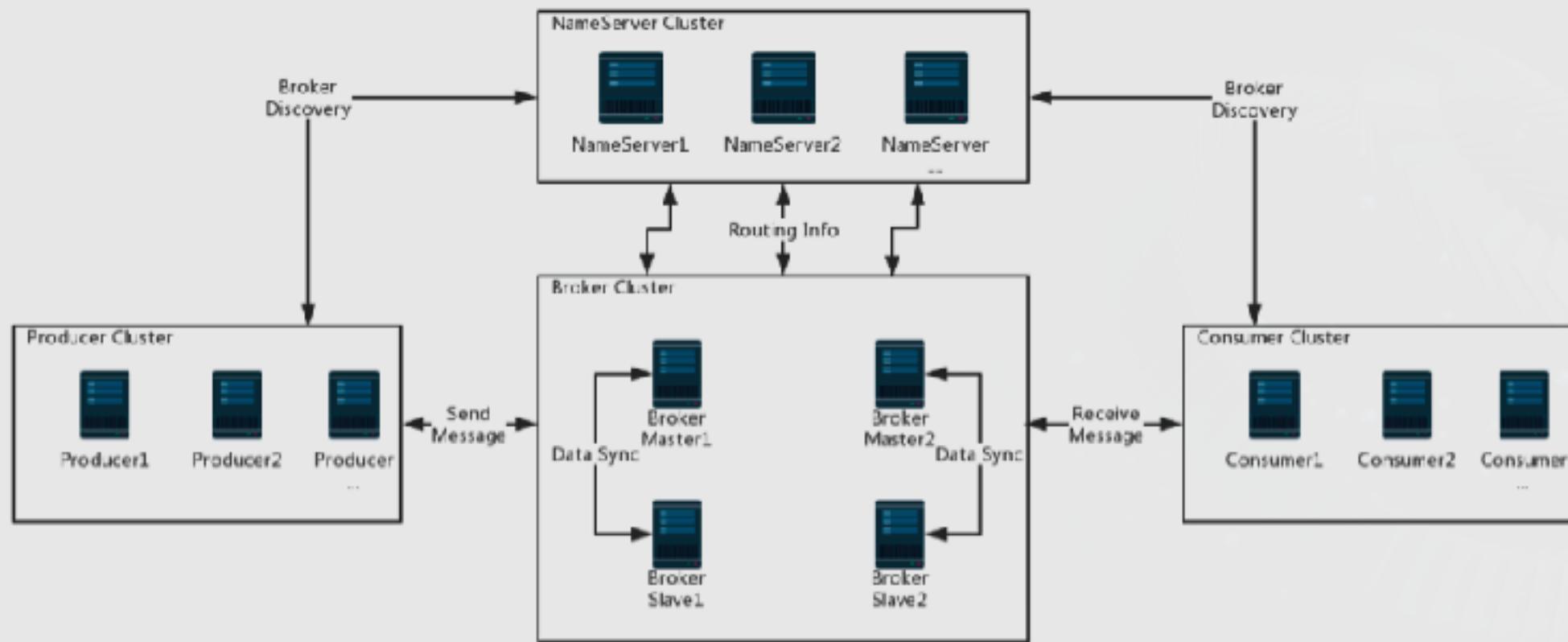
✘ RocketMQ模块

✘ NameServer

✘ Broker

✘ Producer/Consumer

## RocketMQ模块



## RocketMQ模块介绍 – NameServer

### ||| 功能 |||

#### 集群管理：

➤NameServer接受来自Broker集群的注册请求，并提供心跳机制来检查代理是否存活。

#### 路由管理：

➤每个NameServer将保存有关代理群集整个路由信息和客户端查询的队列信息。

### ||| 方法 |||

#### 代码设置：

➤`producer.setNamesrvAddr("10.24.10.24:1024")`

#### JVM参数：

➤`Drocketmq.namesrv.addr=10.24.10.24:1024`

#### 环境变量：

➤`NAMESRV_ADDR=10.24.10.24:1024`

#### HTTP请求：

➤`http://jmenv.tbsite.net:8080/rocketmq/nsaddr`

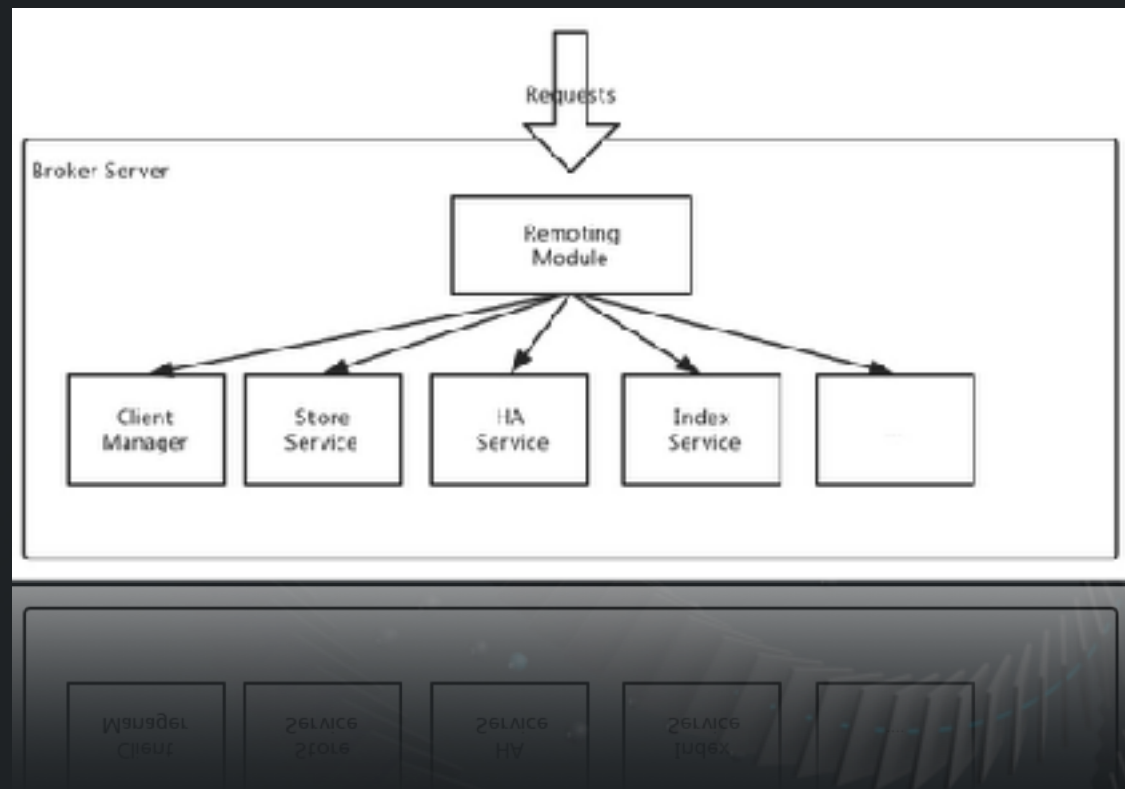


## RocketMQ模块介绍 – Broker

Broker是RocketMQ服务器的大名，主要负责消息的存储、投递、查询、Ha保障等。

有几个重要的子模块：

- **网络连接模块**：提供高性能的网络请求处理，处理来自客户端的网络请求
- **客户端管理**：管理客户端（生产者和消费者），并维护消费这的订阅关系
- **存储服务**：提供简单的API来存储、查询物理磁盘中的消息
- **HA服务**：提供Master和Slave之间的数据同步功能
- **索引服务**：按指定的数据构建索引，并提供快速的消息查询



## RocketMQ模块介绍 – Producer/Consumer

Producer：消息发送者

- 生产者支持分布式部署
- 支持多种负载均衡策略
- 支持快速失败机制
- 支持同步发送和异步发送

Consumer：消息的消费者

- 消费者支持分布式部署
- 支持Push和Pull模型
- 支持集群消费和消息广播。
- 它提供实时消息订阅机制，可以满足大多数消费者的需求

# 03

## RocketMQ 日常运维

✘ 准备工作

✘ NameServer集群部署

✘ Broker集群部署

✘ NameServer集群扩/缩容

✘ Broker集群扩/缩容

✘ 控制台功能介绍

✘ 顺序消息Broker集群扩缩容

## RocketMQ日常运维 – 准备工作

- 安装环境

jdk8+  
centos6+

- RocketMQ-4.3下载

[http://rocketmq.apache.org/release\\_notes/release-notes-4.3.1/](http://rocketmq.apache.org/release_notes/release-notes-4.3.1/)

- 目录结构

**benchmark** : 性能测试脚本

**bin**: 启动、停止、管理、参数优化等

**conf**: 配置文件

**lib**: 依赖包

# RocketMQ日常运维——准备工作

## Benchmark 常用压测脚本

### ● consumer.sh

- t : 指定topic
- g: 指定分组
- p: 是否指定前缀, 如果设置为true, 会在指定分组名称后增加当前的时间戳作为实际分组
- f: 指定过滤类型, 支持Tag、SQL过滤
- e: 过滤表达式

### ● producer.sh

- t: 指定topic
- w: 指定发送线程数
- s: 指定消息大小
- k: 设置是否支持message key, 如果设置为true, 默认设置key=System.currentTimeMillis/1000
- p: 设置消息头的大小

## RocketMQ日常运维——准备工作

- Bin 常用脚本

- **mqadmin**: rocketmq管理控制脚本（后续高级功能详细介绍）
- **mqbroker**: 启动rocketmq服务端
- **mqnamesrv**: 启动nameserver
- **mqshutdown**: 关闭rocketmq服务端
- **os.sh**: 优化系统参数，如果没有对内核参数调优熟悉的专业人士，在启动前一定要执行一遍这个脚本

## RocketMQ日常运维——NameServer集群部署

- 启动

`nohup {rocketmq.home}/bin/mqnamesrv &`

- 配置

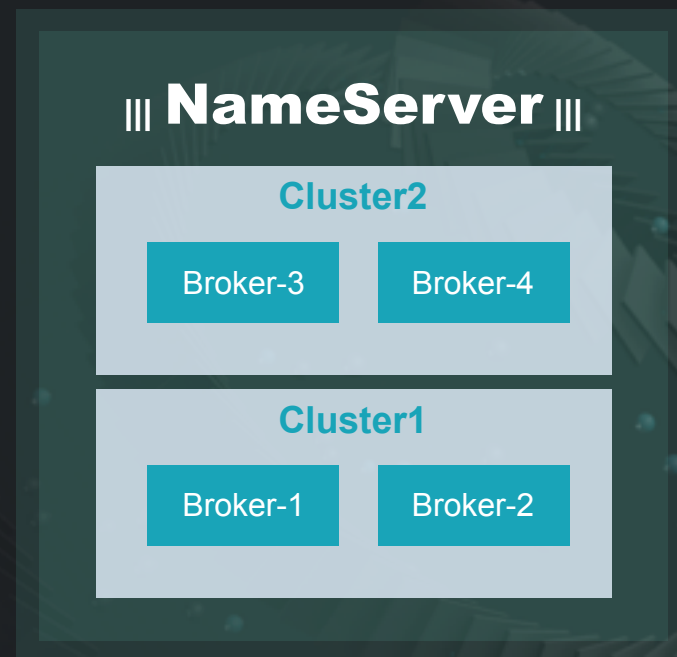
无

- 注意事项

部署多台NameServer保证高可用

一个NameServer可以管理多个集群

客户端尽量使用域名或者VIP访问NameServer，不要直接使用IP



## RocketMQ日常运维——NameServer集群扩缩容

### 机器维护：

- 结束服务
- 机器维护
- 启动服务

### 换机器：

- 新机器服务启动
- 结束服务
- 替换新机器
- 新机器更新到原VIP

### 缩容：

- 结束服务
- 机器下线
- 更新Broker配置

### 扩容：

- 新机器服务启动
- 新增VIP
- 更新Broker配置



## RocketMQ日常运维——Broker集群部署

- 启动

`nohup {rocketmq.home}/bin/mqbroker -c broker.conf &`

- 配置文件

`conf/broker.conf`

`conf/2m-2s/async/broker.properties`

- 注意事项

为了达到更高的性能，一台物理机上不要启动多个实例  
更大的内存，配合合理的参数可以获得更高的性能

## RocketMQ日常运维——Broker集群部署

配置项	含义
brokerClusterName	broker的集群名称，同一个集群的名称相同
brokerName	broker的名称，默认为机器名
brokerId	broker的id，0表示master，正整数表示slave
brokerIP1	如果有多块网卡，需要手工配置该项，如：docker容器ip端口映射场景
brokerPermission	指定broker是只读还是可读可写
defaultTopicQueueNums	默认的topic下分区数量
autoCreateTopicEnable	是否自动创建Topic，一般测试环境设置true，生产环境设置false
autoCreateSubscriptionGroup	是否自动创建分组，一般测试环境设置true，生产环境设置false
flushDiskType	刷盘策略：异步刷盘、同步刷盘
listenPort	broker监听的端口

## RocketMQ运维——Broker集群部署

配置项	含义
storePathCommitLog	broker上commitLog存储的路径
storePathConsumeQueue	broker上消费位点文件的存储路径
storePathIndex	broker上索引文件的存储路径
messageDelayLevel	消息重试队列投递的间隔时间，默认1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m 10m 20m 30m 1h 2h
maxMessageSize	broker接受的单条消息的最大值，超过抛出异常，默认4M
pullMessageThreadPoolNums	服务端处理客户端拉取消息的线程数，一般设置为CPU个数*2
sendMessageThreadPoolNums	服务端处理客户端发送消息的线程数，一般设置为CPU个数
fileReservedTime	保留多长时间的消息内容，单位是小时
deleteWhen	清理时机，每天的几点
diskMaxUsedSpaceRatio	最大的磁盘上使用率，达到阈值后不再接受消息，默认85%

## RocketMQ日常运维——Broker集群扩缩容

### 机器维护:

- 结束服务
- 机器维护
- 启动服务

### 换机器:

- 新机器服务启动, 加入集群
- 老机器停止消息写入
- 等待老机器消息被完全消费, 结束服务
- 老机器下线

### 缩容:

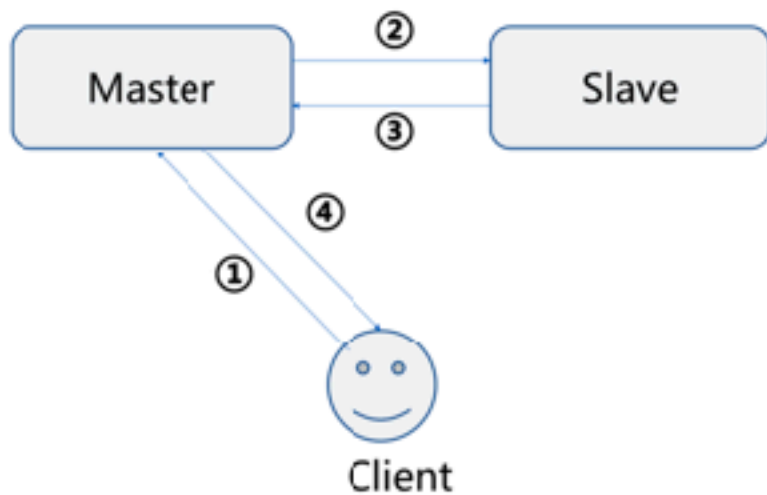
- 下线机器停止写入
- 等待下线机器的消息被完全消费
- 停止服务

### 扩容:

- 新机器服务启动, 加入集群

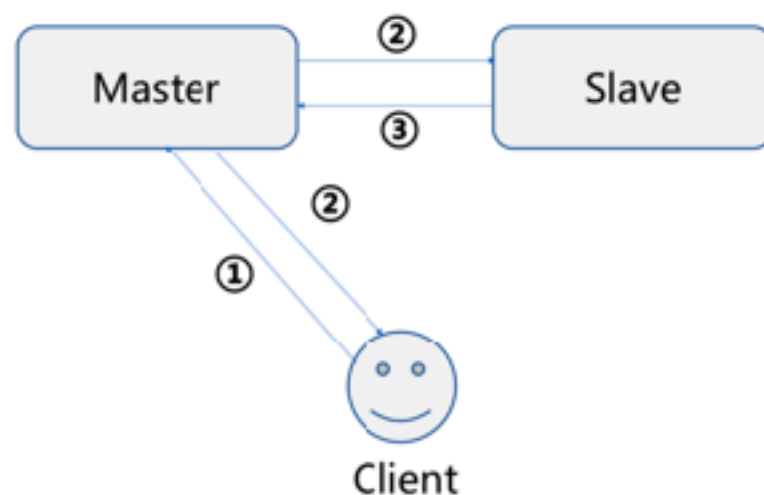
## RocketMQ高可用方案

Sync Mode



- High latency, Low throughput
- No data loss, Automatic failover
- Strong consistency

Async Mode



- Low latency, High throughput
- Some data loss, long MTTR
- Eventual consistency

## RocketMQ日常运维——顺序消息扩缩容

### 缩容：

- 停止集群所有Broker的写入
- 等待集群所有消费完成
- 下线机器
- 开启集群所有Broker的写入

### 扩容：

- 停止集群所有Broker的写入
- 等待集群所有消费完成
- 新增机器，启动服务
- 开启集群所有Broker的写入

### ！ 注意事项

- 顺序消息使用独立集群
- 能不使用顺序消息的就不使用，特别是全局顺序消息
- 顺序消息维护需要特别小心，很容易因为运维的原因导致消息错乱，引起业务问题
- 顺序消息的写入，一般使用【业务ID%分区数量】的方式，这里的分区数量不要使用动态能获取的方式，建议直接在配置中写死

## RocketMQ日常运维 – 控制台介绍

Cluster : DefaultCluster

Broker	NO.	Address	Version	Produce Massage TPS	Consumer Massage TPS	Yesterday Produce Count	Yesterday Consume Count	Today Produce Count	Today Consume Count	Operation
broker-a	0(master)	192.168.1.251:12961	V4_2_0_SNAPSHOT	0.00	0.00	0	0	0	0	<a href="#">STATUS</a> <a href="#">CONFIG</a>

### 配置项

### 含义

Broker	broker的名称，对应配置中brokerName
NO.	broker的Id（角色）
Address	broker的IP+PORT
Version	broker的版本号
Produce Message TPS	发送消息的TPS
Consumer Message TPS	消息推送的TPS
Yesterday Produce Count	昨天接受的消息数量
Yesterday Consume Count	昨天推送的消息数量
Today Produce Count	今天接受的消息数量

## RocketMQ日常运维 – 控制台介绍

Topic: ☒ NORMAL ☐ RETRY ☐ DLO

普通Topic

重试Topic

死信Topic

重置消费位点：

resetOffset

SubscriptionGroup:

Time: 2018-10-29 11:18



## RocketMQ日常运维 – 控制台介绍

SubscriptionGroup	Quantity	Version	Type	Mode	TPS	Delay	Operation
C_FAST_TRACE_GROUP	1	HigherVersion	CONSUME_PASSIVELY	CLUSTERING	0	0	<a href="#">CLIENT</a> <a href="#">CONSUME DETAIL</a> <a href="#">CONFIG</a> <a href="#">DELETE</a>
C_10064_sink	1	HigherVersion	CONSUME_PASSIVELY	CLUSTERING	0	0	<a href="#">CLIENT</a> <a href="#">CONSUME DETAIL</a> <a href="#">CONFIG</a> <a href="#">DELETE</a>

[C\_FAST\_TRACE\_GROUP]CLIENT

ClientId	ClientAddr	Language	Version
192.168.1.242@17748	192.168.1.242:17338	JAVA	HigherVersion

Below is subscription:

Topic	SubExpression
T_trace_client	*
%RETRY%C_FAST_TRACE_GROUP	*

ConsumeType: CONSUME\_PASSIVELY

MessageModel: CLUSTERING

ConsumeFromWhere: CONSUME\_FROM\_FIRST\_OFFSET

[CLOSE](#)

[C\_FAST\_TRACE\_GROUP]Detail

Topic	%RETRY%C_FAST_TRACE_GROUP	Delay	0	LastConsumeTime	1970-01-01 08:00:00	
broker	queue	consumerClient	brokerOffset	consumerOffset	diffTotal	lastTimestamp
broker-a	0	192.168.1.242@17748	224214	224214	0	1970-01-01 08:00:00

Topic	T_trace_client	Delay	0	LastConsumeTime	2018-10-24 11:25:08	
broker	queue	consumerClient	brokerOffset	consumerOffset	diffTotal	lastTimestamp
broker-a	0		134044	134044	0	2018-10-24 11:25:08
broker-a	1		134042	134042	0	2018-10-24 11:25:08
broker-a	2		134045	134045	0	2018-10-24 11:25:08
broker-a	3		134047	134047	0	2018-10-24 11:25:08

## RocketMQ日常运维 – 控制台介绍

TOPIC

MESSAGE KEY

MESSAGE ID

Only Return 2000 Messages

Topic:  Begin: 2018-10-28 10:24 End: 2018-10-29 12:24

TOPIC

MESSAGE KEY

MESSAGE ID

Only Return 64 Messages

Topic:  Key:

TOPIC

MESSAGE KEY

MESSAGE ID

Topic will be empty if your producer client version &lt; 3.5.6

Topic:  MessageId:  

Message ID

AC150C01412E1216025C901466BFC00

Topic

test

Tag

test

Key

155

Online

2018-10-29

Message

消息消息体

messageTrackList

consumerGroup	trackType	Operation
C_1077C_sink	NOT_ONLINE	<input type="button" value="IGNORE MESSAGE"/> <input type="button" value="VIEW EXCEPTION"/>

## RocketMQ日常运维 – NameServer集群扩缩容

机器维护	换机器	缩容	扩容
<ul style="list-style-type: none"><li>➤ 结束服务</li><li>➤ 机器维护</li><li>➤ 启动服务</li></ul>	<ul style="list-style-type: none"><li>➤ 新机器服务启动</li><li>➤ 结束服务</li><li>➤ 替换新机器</li><li>➤ 新机器更新到原VIP</li></ul>	<ul style="list-style-type: none"><li>➤ 结束服务</li><li>➤ 机器下线</li><li>➤ 更新Broker配置</li></ul>	<ul style="list-style-type: none"><li>➤ 新机器服务启动</li><li>➤ 新增VIP</li><li>➤ 更新Broker配置</li></ul>

# 04



## RocketMQ 运维建议

mqadmin工具

JVM建议

## RocketMQ运维建议 – mqadmin工具

- 获取方式

位于RocketMQ\_HOME/bin/mqadmin

- 如何使用

```
[root@pomies101 bin]# sh mqadmin help updateTopic
java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
usage: mqadmin updateTopic [-b <arg>] [-c <arg>] [-h] [-n <arg>] [-o <arg>] [-p <arg>] [-r <arg>] [-s <arg>]
      -t <arg> [-u <arg>] [-w <arg>]
-b,--brokerAddr <arg>    create topic to which broker
-c,--clusterName <arg>    create topic to which cluster
-h,--help                  Print help
-n,--namesrvAddr <arg>    Name server address list, eg: 192.168.0.1:9876;192.168.0.2:9876
-o,--order <arg>          set topic's order(true|false)
-p,--perm <arg>           set topic's permission(71416), Intro[7:RW 4:R; 6:RW]
-r,--readQueueNums <arg>  set read queue nums
-s,--hasUnitSub <arg>     has unit sub (true|false)
-t,--topic <arg>          topic name
-u,--unit <arg>           is unit topic (true|false)
-w,--writeQueueNums <arg> set write queue nums
```

The most commonly used mqadmin commands are:

updateTopic	Update or create topic
deleteTopic	Delete topic from broker and NameServer.
updateSubGroup	Update or create subscription group
deleteSubGroup	Delete subscription group from broker.
updateBrokerConfig	Update broker's config
updateTopicPerm	Update topic perm
topicRoute	Examine topic route info
topicStatus	Examine topic status info
topicClusterList	get cluster info for topic
brokerStatus	Fetch broker runtime status data
queryMsgById	Query Message by Id
queryMsgByKey	Query Message by Key
queryMsgByUniqueKey	Query Message by Unique key
queryMsgByOffset	Query Message by offset
queryMsgByUniqueKey	Query Message by Unique key
printMsg	Print Message Detail
printMsgByQueue	Print Message Detail
sendMsgStatus	send msg to broker.
brokerConsumeStats	Fetch broker consume stats data
producerConnection	Query producer's socket connection and client version
consumerConnection	Query consumer's socket connection, client version and subscription
consumerProgress	Query consumer's progress, speed
consumerStatus	Query consumer's internal data structure
cloneGroupOffset	clone offset from other group.
clusterList	List all of clusters
topicList	Fetch all topic list from name server
updateKVConfig	Create or update KV config.
deleteKVConfig	Delete KV config.
wipeWritePerm	Wipe write perm of broker in all name server
resetOffsetByTime	Reset consumer offset by timestamp(without client restart).
updateOrderConf	Create or update or delete order conf
cleanExpiredCQ	Clean expired ConsumeQueue on broker.
cleanUnusedTopic	Clean unused topic on broker.
startMonitoring	Start Monitoring
statsAll	Topic and Consumer top stats
allocateMQ	Allocate MQ
checkMsgSendRT	check message send response time
clusterRT	List All clusters Message Send RT
getNamesrvConfig	Get configs of name server.
updateNamesrvConfig	Update configs of name server.
getBrokerConfig	Get broker config by cluster or special broker
queryCq	Query cq command.

## RocketMQ运维建议 – JVM使用建议

### ● 堆内存大小

`-server -Xms8g -Xmx8g -Xmn4g`  
`-XX:+AlwaysPreTouch` （启动时预分配全部内存）

### ● 垃圾回收

`-XX:+UseG1GC -XX:G1HeapRegionSize=16m -XX:G1ReservePercent=25 -XX:InitiatingHeapOccupancyPercent=30`

开启垃圾回收日志

`-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=30m`

对外内存使用

`MaxDirectMemorySize`

### ● Linux内核参数

`vm.extra_free_kbytes`：对于低延迟内存分配场景，需要设置部分额外内存，以避免延时

`vm.min_free_kbytes`：如果将其设置为低于1024KB，高负载下容易出现死锁

`vm.max_map_count`：限制进程最大内存映射数量。RocketMQ将使用mmap加载CommitLog和ConsumeQueue，因此建议为此参数设置较大的值，如655350

`vm.swappiness`：定义内核交换内存页面的积极程度，较低的值会减少交换量，建议设置为1

`ulimit -n`：打开的文件句柄数，建议655350

队列调度类型：`deadline`

## RocketMQ运维建议 – 其他建议

- 部署建议

- 一台物理机部署一个实例

- 高可用部署

- 根据场景选择刷盘和同步策略

- SSD盘对综合性能有提升，尤其在消息堆积时的表现

# END