



性能优化手册

金蝶Apusic分布式消息队列V2.0.6_for_rocketmq

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

- 1 前言
 - 1.1 适用对象
 - 1.2 相关文档
 - 1.3 技术支持
- 2 系统层面优化
 - 2.1 操作系统优化
 - 2.1.1 文件描述符
 - 2.1.2 内核参数
 - 2.1.3 磁盘优化
 - 2.2 网络优化
- 3 RocketMQ 配置优化
 - 3.1 Broker 配置
 - 3.2 存储配置
 - 3.3 JVM 配置
 - 3.3.1 NameServer JVM
 - 3.3.2 Broker JVM
- 4 客户端优化
 - 4.1 生产者优化
 - 4.1.1 发送方式选择
 - 4.1.2 生产者参数
 - 4.1.3 批量发送
 - 4.2 消费者优化
 - 4.2.1 消费线程数
 - 4.2.2 消费批量大小
 - 4.2.3 消费位点策略
- 5 集群优化
 - 5.1 节点部署策略
 - 5.2 主从复制优化
 - 5.2.1 同步复制 vs 异步复制
 - 5.3 刷盘策略优化
 - 5.3.1 同步刷盘 vs 异步刷盘
 - 5.4 Topic 队列数优化

- 6 监控与调优
 - 6.1 性能指标
 - 6.2 性能测试
 - 6.3 瓶颈分析
- 7 常见问题优化
 - 7.1 消息堆积优化
 - 7.2 发送性能优化
 - 7.3 消费延迟优化
- 8 性能优化检查清单
 - 8.1 系统层面
 - 8.2 RocketMQ 配置
 - 8.3 客户端优化
 - 8.4 集群优化

1 前言

本文档介绍金蝶Apusic分布式消息队列for RocketMQ（Apusic Distributed Message Queue for RocketMQ，简称：ADMQ for RocketMQ）的性能优化最佳实践，帮助用户充分利用系统性能。

1.1 适用对象

本文档适用于IT信息化业务负责人、研发经理、软件项目经理、软件架构师、运维工程师。

1.2 相关文档

了解更多ADMQ for RocketMQ产品相关的信息，请参阅以下ADMQ for RocketMQ产品手册文档集：

序号	手册文档	说明
1	金蝶Apusic分布式消息队列for RocketMQ 快速使用手册	简单介绍了如何快速上手使用ADMQ for RocketMQ 。
2	金蝶Apusic分布式消息队列for RocketMQ 安装手册	详细介绍如何在各操作系统上安装ADMQ for RocketMQ，以及ADMQ for RocketMQ服务启停等操作。
3	金蝶Apusic分布式消息队列for RocketMQ 消息引擎用户手册	详细介绍 ADMQ for RocketMQ 消息引擎相关功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分布式消息队列for RocketMQ 管控台用户手册	详细介绍ADMQ for RocketMQ管控台相关功能的使用和操作说明。
5	金蝶Apusic分布式消息队列for RocketMQ 开发手册	详细介绍基于各开发语言进行ADMQ for RocketMQ客户端应用开发的说明。
6	金蝶Apusic分布式消息队列for RocketMQ 迁移手册	详细介绍从RocketMQ迁移到ADMQ for RocketMQ的说明。
7	金蝶Apusic分布式消息队列for RocketMQ 运维手册	详细介绍ADMQ for RocketMQ的监控、运维、安全加固等运维说明。
8	金蝶Apusic分布式消息队列for RocketMQ 性能优化手册	详细介绍ADMQ for RocketMQ性能调优的说明。

1.3 技术支持

ADMQ for RocketMQ产品提供全面的技术支持服务，您可以通过以下方式获得技术支持：

- 网址: www.apusic.com
- 电话: 400-855-5800
- 邮箱: support@apusic.com
- 金蝶云社区: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技术支持时, 请提供如下信息:

1. 您的姓名
2. 公司与联系方式
3. 操作系统及其版本
4. 产品版本号
5. 出现异常及错误的日志、截图等详细信息

2 系统层面优化

2.1 操作系统优化

2.1.1 文件描述符

```
# 查看当前限制
ulimit -n

# 修改限制 (/etc/security/limits.conf)
* soft nofile 65535
* hard nofile 65535
```

2.1.2 内核参数

```
# /etc/sysctl.conf

# 增加端口范围
net.ipv4.ip_local_port_range = 1024 65535

# TCP 连接优化
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 30
net.core.somaxconn = 65535

# 内存优化
vm.swappiness = 10
vm.dirty_ratio = 40
vm.dirty_background_ratio = 10

# 应用配置
sysctl -p
```

2.1.3 磁盘优化

优化项	建议
文件系统	使用 XFS 文件系统
挂载选项	noatime,nodiratime
磁盘调度	使用 noop 或 deadline
RAID 配置	使用 RAID 10 或 RAID 0
SSD 优化	启用 TRIM, 预留 over-provisioning

```
# 查看磁盘调度算法
```

```
cat /sys/block/sda/queue/scheduler
```

```
# 设置为 deadline
```

```
echo deadline > /sys/block/sda/queue/scheduler
```

```
# 挂载选项 (/etc/fstab)
```

```
/dev/sda1 /data/rocketmq xfs noatime,nodiratime 0 0
```

2.2 网络优化

```
# 网络缓冲区
```

```
net.core.rmem_max = 134217728
```

```
net.core.wmem_max = 134217728
```

```
net.ipv4.tcp_rmem = 4096 87380 134217728
```

```
net.ipv4.tcp_wmem = 4096 65536 134217728
```

```
# TCP 优化
```

```
net.ipv4.tcp_window_scaling = 1
```

```
net.ipv4.tcp_timestamps = 1
```

```
net.ipv4.tcp_sack = 1
```

```
net.ipv4.tcp_no_metrics_save = 1
```


3 RocketMQ 配置优化

3.1 Broker 配置

参数	默认值	推荐值	说明
sendMessageThreadPoolNums	1	CPU 核数	发送消息线程数
pullMessageThreadPoolNums	16+	根据负载调整	拉取消息线程数
queryMessageThreadPoolNums	8	根据负载调整	查询消息线程数
adminBrokerThreadPoolNums	16	根据负载调整	管理操作线程数
brokerRole	ASYNC_MASTER	SYNC_MASTER	Broker 角色
flushDiskType	ASYNC_FLUSH	SYNC_FLUSH	刷盘方式
flushIntervalCommitLog	500ms	根据可靠性要求	CommitLog 刷盘间隔
flushIntervalConsumeQueue	1000ms	根据可靠性要求	ConsumeQueue 刷盘间隔

```
# broker.conf
sendMessageThreadPoolNums=16
pullMessageThreadPoolNums=16
brokerRole=SYNC_MASTER
flushDiskType=SYNC_FLUSH
flushIntervalCommitLog=500
flushIntervalConsumeQueue=1000
```

3.2 存储配置

参数	默认值	推荐值	说明
mappedFileSizeCommitLog	1GB	1GB	CommitLog 文件大小
mappedFileSizeConsumeQueue	300MB	300MB	ConsumeQueue 文件大小
flushCommitLogTimed	false	false	是否定时刷盘
deleteWhen	04	04	删除过期文件时间

fileReservedTime	72	根据磁盘容量	文件保留时间（小时）
diskMaxUsedSpaceRatio	75	75	磁盘最大使用比例

```
# broker.conf
mappedFileSizeCommitLog=1073741824
mappedFileSizeConsumeQueue=300000000
deleteWhen=04
fileReservedTime=72
diskMaxUsedSpaceRatio=75
```

3.3 JVM 配置

3.3.1 NameServer JVM

```
# bin/runserver.sh
JAVA_OPT="${JAVA_OPT} -server -Xms4g -Xmx4g -Xmn2g"
JAVA_OPT="${JAVA_OPT} -XX:+UseG1GC -XX:G1HeapRegionSize=16m"
JAVA_OPT="${JAVA_OPT} -XX:G1ReservePercent=25"
JAVA_OPT="${JAVA_OPT} -XX:InitiatingHeapOccupancyPercent=30"
```

3.3.2 Broker JVM

```
# bin/runbroker.sh
JAVA_OPT="${JAVA_OPT} -server -Xms8g -Xmx8g -Xmn4g"
JAVA_OPT="${JAVA_OPT} -XX:+UseG1GC -XX:G1HeapRegionSize=16m"
JAVA_OPT="${JAVA_OPT} -XX:G1ReservePercent=25"
JAVA_OPT="${JAVA_OPT} -XX:InitiatingHeapOccupancyPercent=30"
JAVA_OPT="${JAVA_OPT} -XX:MaxDirectMemorySize=4g"
```

4 客户端优化

4.1 生产者优化

4.1.1 发送方式选择

发送方式	吞吐量	可靠性	适用场景
同步发送	中	高	需要确认的场景
异步发送	高	中	高吞吐场景
单向发送	极高	低	日志采集等
批量发送	极高	中	高吞吐场景

4.1.2 生产者参数

```
DefaultMQProducer producer = new DefaultMQProducer("test-group");
producer.setNamesrvAddr("localhost:9876");

// 发送超时时间
producer.setSendMsgTimeout(3000);

// 压缩消息体阈值（超过该值压缩）
producer.setCompressMsgBodyOverHowmuch(1024 * 4);

// 最大消息大小
producer.setMaxMessageSize(1024 * 1024 * 4);

// 重试次数
producer.setRetryTimesWhenSendFailed(2);
producer.setRetryTimesWhenSendAsyncFailed(2);

// 是否在内部发送失败时重试另一台 Broker
producer.setRetryAnotherBrokerWhenNotStoreOK(false);
```

4.1.3 批量发送

```
List<Message> messages = new ArrayList<>();
for (int i = 0; i < 100; i++) {
    messages.add(new Message("batch-topic", "tag-a", ("msg-" +
i).getBytes()));
}
// 批量发送, 单批不超过 1MB
producer.send(messages);
```

4.2 消费者优化

4.2.1 消费线程数

```
DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("test-
group");
consumer.setNamesrvAddr("localhost:9876");

// 最小消费线程数
consumer.setConsumeThreadMin(20);

// 最大消费线程数
consumer.setConsumeThreadMax(64);

// 消费超时时间 (分钟)
consumer.setConsumeTimeout(15);
```

4.2.2 消费批量大小

```
// 每次消费的最大消息数
consumer.setConsumeMessageBatchMaxSize(10);

// 每次拉取的最大消息数
consumer.setPullBatchSize(32);
```

```
// 拉取间隔（毫秒）  
consumer.setPullInterval(0);
```

4.2.3 消费位点策略

```
// 从最后偏移量开始消费  
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_LAST_OFFSET)  
  
// 从最早偏移量开始消费  
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET)  
  
// 从指定时间开始消费  
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_TIMESTAMP);  
consumer.setConsumeTimestamp("20260101000000");
```

5 集群优化

5.1 节点部署策略

部署策略	说明	适用场景
同机房部署	所有节点在同一机房	低延迟、高吞吐
跨机房部署	节点分布在不同机房	高可用、容灾
异构部署	不同配置节点混合	成本优化

5.2 主从复制优化

5.2.1 同步复制 vs 异步复制

复制方式	可靠性	性能	适用场景
同步复制	高	低	金融交易等强一致场景
异步复制	中	高	一般业务场景

```
# broker.conf
# 同步复制
brokerRole=SYNC_MASTER

# 异步复制
brokerRole=ASYNC_MASTER
```

5.3 刷盘策略优化

5.3.1 同步刷盘 vs 异步刷盘

刷盘方式	可靠性	性能	适用场景
同步刷盘	高	低	强一致场景
异步刷盘	中	高	一般业务场景

```
# broker.conf
# 同步刷盘
flushDiskType=SYNC_FLUSH

# 异步刷盘
flushDiskType=ASYNC_FLUSH
```

5.4 Topic 队列数优化

场景	队列数建议
低吞吐	4-8
中等吞吐	8-16
高吞吐	16-32
超高吞吐	32-64

```
# 创建 Topic 时指定队列数
bin/mqadmin updateTopic -n localhost:9876 -c DefaultCluster -t test-
topic -r 16 -w 16
```

6 监控与调优

6.1 性能指标

指标	健康范围	优化建议
Broker TPS	根据硬件配置	扩容或优化生产者
消费 TPS	接近生产 TPS	增加消费者或优化消费逻辑
消息堆积	< 100万	增加消费者或优化消费逻辑
磁盘使用率	< 75%	扩容或清理过期消息
内存使用率	< 80%	增加内存或优化 JVM 参数
消费延迟	< 1秒	优化消费逻辑或增加消费者

6.2 性能测试

```
# 使用 benchmark 工具测试
bin/benchmark.sh producer -n localhost:9876 -t test-topic -s 128 -w
16

# 参数说明
# -n: NameServer 地址
# -t: Topic 名称
# -s: 消息大小 (字节)
# -w: 并发数
```

6.3 瓶颈分析

1. **CPU 瓶颈**: 发送/消费线程数不足、压缩解压、序列化
2. **内存瓶颈**: JVM GC 频繁、DirectBuffer 不足
3. **磁盘瓶颈**: 刷盘方式、磁盘 I/O 不足、文件系统性能
4. **网络瓶颈**: 跨机房部署、带宽不足、网络延迟

7 常见问题优化

7.1 消息堆积优化

1. 增加消费者数量
2. 优化消费者处理逻辑
3. 增加消费线程数
4. 使用批量消费
5. 扩容 Broker 节点

7.2 发送性能优化

1. 使用异步发送
2. 使用批量发送
3. 增加生产者数量
4. 优化消息大小
5. 关闭消息轨迹（如不需要）

7.3 消费延迟优化

1. 增加消费线程数
2. 优化消费逻辑
3. 使用并发消费
4. 扩容消费者
5. 检查是否有慢消费者

8 性能优化检查清单

8.1 系统层面

- ☐ 文件描述符限制已调整
- ☐ 内核参数已优化
- ☐ 磁盘使用 XFS 文件系统
- ☐ 网络参数已优化
- ☐ 系统资源充足（CPU、内存、磁盘）

8.2 RocketMQ 配置

- ☐ Broker 线程数合理设置
- ☐ 刷盘策略根据业务选择
- ☐ 复制策略根据业务选择
- ☐ 文件保留时间合理设置
- ☐ JVM 参数已优化

8.3 客户端优化

- ☐ 发送方式选择合适
- ☐ 消费线程数合理设置
- ☐ 批量大小合理设置
- ☐ 重试次数合理设置
- ☐ 消费位点策略合适

8.4 集群优化

- ☐ Topic 队列数合理设置
- ☐ 主从复制策略合适
- ☐ 刷盘策略合适
- ☐ 集群规模满足业务需求
- ☐ 节点部署策略合适

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年，前身为“金蝶中间件公司”，是金蝶集团旗下新一代软件基础云平台服务商，云计算国家标准制定企业，国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业，也是“两网一站四库十二金”国家重点工程的基础平台提供商，产品广泛应用于政府、军工、金融、能源等关键行业，累计服务客户总数超过10万家。



云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网：www.apusic.com

