



APUSIC
固若长城
睿比世界

效能調優

金蝶Apusic分散式訊息佇列V2.0.6_for_rocketmq

版權所有 © 深圳市金蝶天燕雲計算股份有限公司2026。保留所有權利。

版權聲明

本文件所涉及的軟件著作權、版權等知識產權已依法進行了註冊，由金蝶天燕雲計算股份有限公司合法擁有。受《中華人民共和國著作權法》《計算機軟件保護條例》《知識產權保護條例》和相關國際版權條約、法律、法規以及其它知識產權法律和條約的保護。未經授權許可，不得非法使用。

免責聲明

本文件包含的版權資訊由金蝶天燕雲計算股份有限公司合法擁有，受法律的保護，金蝶天燕雲計算股份有限公司對本文件可能涉及到的非金蝶天燕雲計算股份有限公司的資訊不承擔任何責任。在法律允許的範圍內，您可以查閱並僅能夠在《中華人民共和國著作權法》規定的合法範圍內複製和打印本文件。任何單位和個人未經金蝶天燕雲計算股份有限公司書面授權許可，不得使用、修改、再發佈本文件的任何部分和內容，否則將被視為侵權，金蝶天燕雲計算股份有限公司有依法追究其責任的權利。

本文件如有更新，不另行通知。對本文件中的問題您可向金蝶天燕雲計算股份有限公司告知或查詢。未經本公司明確授予的任何權利均予保留。

商標聲明

 是深圳市金蝶天燕雲計算股份有限公司向中華人民共和國國家商標局申請註冊的註冊商標，註冊商標專用權由金蝶天燕合法擁有，受法律保護。未經金蝶天燕的書面許可，任何單位及個人不得以任何方式或理由對該商標的任何部分進行使用、複製、修改、傳播、抄錄或與其它產品捆綁使用銷售。凡侵犯金蝶天燕商標權的，金蝶天燕將依法追究其法律責任。本文件提及的其他所有商標或註冊商標，由各自的所有人擁有。

目錄

- 1 前言
 - 1.1 適用對象
 - 1.2 相關文檔
 - 1.3 技術支援
- 2 系統層面優化
 - 2.1 作業系統優化
 - 2.1.1 檔案描述符
 - 2.1.2 核心參數
 - 2.1.3 磁碟最佳化
 - 2.2 網路最佳化
- 3 RocketMQ 配置最佳化
 - 3.1 Broker 配置
 - 3.2 儲存配置
 - 3.3 JVM 配置
 - 3.3.1 NameServer JVM
 - 3.3.2 Broker JVM
- 4 用戶端最佳化
 - 4.1 生產者最佳化
 - 4.1.1 傳送方式選擇
 - 4.1.2 生產者參數
 - 4.1.3 批次傳送
 - 4.2 消費者最佳化
 - 4.2.1 消費執行緒數
 - 4.2.2 消費批次大小
 - 4.2.3 消費位點策略
- 5 叢集最佳化
 - 5.1 節點部署策略
 - 5.2 主從複製最佳化
 - 5.2.1 同步複製 vs 非同步複製
 - 5.3 刷盤策略最佳化
 - 5.3.1 同步刷盤 vs 非同步刷盤
 - 5.4 Topic 佇列數最佳化

- 6 監控與調優
 - 6.1 效能指標
 - 6.2 效能測試
 - 6.3 瓶頸分析
- 7 常見問題最佳化
 - 7.1 訊息堆積最佳化
 - 7.2 傳送效能最佳化
 - 7.3 消費延遲最佳化
- 8 效能最佳化檢查清單
 - 8.1 系統層面
 - 8.2 RocketMQ 配置
 - 8.3 用戶端最佳化
 - 8.4 叢集最佳化

1 前言

本文檔介紹金蝶Apusic分散式訊息佇列for RocketMQ（Apusic Distributed Message Queue for RocketMQ，簡稱：ADMQ for RocketMQ）的效能優化最佳實踐，幫助使用者充分利用系統效能。

1.1 適用對象

本文檔適用於IT資訊化業務負責人、研發經理、軟體專案經理、軟體架構師、運維工程師。

1.2 相關文檔

了解更多ADMQ for RocketMQ產品相關的資訊，請參閱以下ADMQ for RocketMQ產品手冊文件集：

序號	手冊文檔	說明
1	金蝶Apusic分散式訊息佇列for RocketMQ 快速使用手冊	簡單介紹了如何快速上手使用ADMQ for RocketMQ。
2	金蝶Apusic分散式訊息佇列for RocketMQ 安裝手冊	詳細介紹如何在各作業系統上安裝ADMQ for RocketMQ，以及ADMQ for RocketMQ服務啟停等操作。
3	金蝶Apusic分散式訊息佇列for RocketMQ 訊息引擎使用者手冊	詳細介紹 ADMQ for RocketMQ 訊息引擎相關功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分散式訊息佇列for RocketMQ 管控台使用者手冊	詳細介紹ADMQ for RocketMQ管控台相關功能的使用和操作說明。
5	金蝶Apusic分散式訊息佇列for RocketMQ 開發手冊	詳細介紹基於各開發語言進行ADMQ for RocketMQ用戶端應用開發的說明。
6	金蝶Apusic分散式訊息佇列for RocketMQ 遷移手冊	詳細介紹從RocketMQ遷移到ADMQ for RocketMQ的說明。
7	金蝶Apusic分散式訊息佇列for RocketMQ 運維手冊	詳細介紹ADMQ for RocketMQ的監控、運維、安全加固等運維說明。
8	金蝶Apusic分散式訊息佇列for RocketMQ 效能優化手冊	詳細介紹ADMQ for RocketMQ效能調優的說明。

1.3 技術支援

ADMQ for RocketMQ產品提供全面的技術支援服務，您可以透過以下方式獲得技術支援：

- 網址: www.apusic.com
- 電話: 400-855-5800
- 郵箱: support@apusic.com
- 金蝶雲社區: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技術支援時，請提供如下資訊：

1. 您的姓名
2. 公司資訊與聯絡方式
3. 作業系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細資訊

2 系統層面優化

2.1 作業系統優化

2.1.1 檔案描述符

```
# 查看目前限制
ulimit -n

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

2.1.2 核心參數

```
# /etc/sysctl.conf

# 增加埠範圍
net.ipv4.ip_local_port_range = 1024 65535

# TCP 連線最佳化
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 30
net.core.somaxconn = 65535

# 記憶體最佳化
vm.swappiness = 10
vm.dirty_ratio = 40
vm.dirty_background_ratio = 10

# 應用配置
sysctl -p
```

2.1.3 磁碟最佳化

最佳化項	建議
檔案系統	使用 XFS 檔案系統
掛載選項	noatime,nodiratime
磁碟排程	使用 noop 或 deadline
RAID 配置	使用 RAID 10 或 RAID 0
SSD 最佳化	啟用 TRIM, 預留 over-provisioning

查看磁碟排程演算法

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

設定為 deadline

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

掛載選項 (/etc/fstab)

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

2.2 網路最佳化

網路緩衝區

```
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 最佳化

```
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 配置最佳化

3.1 Broker 配置

參數	預設值	推薦值	說明
sendMessageThreadPoolNums	1	CPU 核心數	傳送訊息執行緒數
pullMessageThreadPoolNums	16+	根據負載調整	拉取訊息執行緒數
queryMessageThreadPoolNums	8	根據負載調整	查詢訊息執行緒數
adminBrokerThreadPoolNums	16	根據負載調整	管理操作執行緒數
brokerRole	ASYNC_MASTER	SYNC_MASTER	Broker 角色
flushDiskType	ASYNC_FLUSH	SYNC_FLUSH	刷盤方式
flushIntervalCommitLog	500ms	根據可靠性要求	CommitLog 刷盤間隔
flushIntervalConsumeQueue	1000ms	根據可靠性要求	ConsumeQueue 刷盤間隔

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

3.2 儲存配置

參數	預設值	推薦值	說明
mappedFileSizeCommitLog	1GB	1GB	CommitLog 檔案大小
mappedFileSizeConsumeQueue	300MB	300MB	ConsumeQueue 檔案大小
flushCommitLogTimed	false	false	是否定時刷盤
deleteWhen	04	04	刪除過期檔案時間

fileReservedTime	72	根據磁碟容量	檔案保留時間（小時）
diskMaxUsedSpaceRatio	75	75	磁碟最大使用比例

```
# 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 用戶端最佳化

4.1 生產者最佳化

4.1.1 傳送方式選擇

傳送方式	吞吐量	可靠性	適用場景
同步傳送	中	高	需要確認的場景
非同步傳送	高	中	高吞吐場景
單向傳送	極高	低	日誌採集等
批次傳送	極高	中	高吞吐場景

4.1.2 生產者參數

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

// 傳送超時時間
producer.setSendMsgTimeout(3000);

// 壓縮訊息體閾值（超過該值壓縮）
producer.setCompressMsgBodyOverHowmuch(1024 * 4);

// 最大訊息大小
producer.setMaxMessageSize(1024 * 1024 * 4);

// 重試次數
producer.setRetryTimesWhenSendFailed(2);
producer.setRetryTimesWhenSendAsyncFailed(2);

// 是否在內部傳送失敗時重試另一台 Broker
producer.setRetryAnotherBrokerWhenNotStoreOK(false);
```

4.1.3 批次傳送

```
List<Message> messages = new ArrayList<>();
for (int i = 0; i < 100; i++) {
    messages.add(new Message("batch-topic", "tag-a", ("msg-" +
i).getBytes()));
}
// 批次傳送, 單批不超過 1MB
producer.send(messages);
```

4.2 消費者最佳化

4.2.1 消費執行緒數

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

// 最小消費執行緒數
consumer.setConsumeThreadMin(20);

// 最大消費執行緒數
consumer.setConsumeThreadMax(64);

// 消費超時時間 (分鐘)
consumer.setConsumeTimeout(15);
```

4.2.2 消費批次大小

```
// 每次消費的最大訊息數
consumer.setConsumeMessageBatchMaxSize(10);

// 每次拉取的最大訊息數
consumer.setPullBatchSize(32);
```

```
// 拉取間隔（毫秒）  
consumer.setPullInterval(0);
```

4.2.3 消費位點策略

```
// 從最後偏移量開始消費  
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_LAST_OFFSET)  
  
// 從最早偏移量開始消費  
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET)  
  
// 從指定時間開始消費  
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_TIMESTAMP);  
consumer.setConsumeTimestamp("20260101000000");
```

5 叢集最佳化

5.1 節點部署策略

部署策略	說明	適用場景
同機房部署	所有節點在同一機房	低延遲、高吞吐
跨機房部署	節點分佈在不同機房	高可用、容災
異構部署	不同配置節點混合	成本最佳化

5.2 主從複製最佳化

5.2.1 同步複製 vs 非同步複製

複製方式	可靠性	效能	適用場景
同步複製	高	低	金融交易等強一致場景
非同步複製	中	高	一般業務場景

```
# broker.conf
# 同步複製
brokerRole=SYNC_MASTER

# 非同步複製
brokerRole=ASYNC_MASTER
```

5.3 刷盤策略最佳化

5.3.1 同步刷盤 vs 非同步刷盤

刷盤方式	可靠性	效能	適用場景
同步刷盤	高	低	強一致場景
非同步刷盤	中	高	一般業務場景

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

# 非同步刷盤
flushDiskType=ASYNC_FLUSH
```

5.4 Topic 佇列數最佳化

場景	佇列數建議
低吞吐	4-8
中等吞吐	8-16
高吞吐	16-32
超高吞吐	32-64

```
# 建立 Topic 時指定佇列數
bin/mqadmin updateTopic -n localhost:9876 -c DefaultCluster -t test-
topic -r 16 -w 16
```

6 監控與調優

6.1 效能指標

指標	健康範圍	最佳化建議
Broker TPS	根據硬體配置	擴容或最佳化生產者
消費 TPS	接近生產 TPS	增加消費者或最佳化消費邏輯
訊息堆積	< 100萬	增加消費者或最佳化消費邏輯
磁碟使用率	< 75%	擴容或清理過期訊息
記憶體使用率	< 80%	增加記憶體或最佳化 JVM 參數
消費延遲	< 1秒	最佳化消費邏輯或增加消費者

6.2 效能測試

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

# 參數說明
# -n: NameServer 位址
# -t: Topic 名稱
# -s: 訊息大小 (位元組)
# -w: 並發數
```

6.3 瓶頸分析

1. CPU 瓶頸：傳送/消費執行緒數不足、壓縮解壓、序列化
2. 記憶體瓶頸：JVM GC 頻繁、DirectBuffer 不足
3. 磁碟瓶頸：刷盤方式、磁碟 I/O 不足、檔案系統效能
4. 網路瓶頸：跨機房部署、頻寬不足、網路延遲

7 常見問題最佳化

7.1 訊息堆積最佳化

1. 增加消費者數量
2. 最佳化消費者處理邏輯
3. 增加消費執行緒數
4. 使用批次消費
5. 擴容 Broker 節點

7.2 傳送效能最佳化

1. 使用非同步傳送
2. 使用批次傳送
3. 增加生產者數量
4. 最佳化訊息大小
5. 關閉訊息軌跡（如不需要）

7.3 消費延遲最佳化

1. 增加消費執行緒數
2. 最佳化消費邏輯
3. 使用並發消費
4. 擴容消費者
5. 檢查是否有慢消費者

8 效能最佳化檢查清單

8.1 系統層面

- ☐ 檔案描述符限制已調整
- ☐ 核心參數已最佳化
- ☐ 磁碟使用 XFS 檔案系統
- ☐ 網路參數已最佳化
- ☐ 系統資源充足（CPU、記憶體、磁碟）

8.2 RocketMQ 配置

- ☐ Broker 執行緒數合理設定
- ☐ 刷盤策略根據業務選擇
- ☐ 複製策略根據業務選擇
- ☐ 檔案保留時間合理設定
- ☐ JVM 參數已最佳化

8.3 用戶端最佳化

- ☐ 傳送方式選擇合適
- ☐ 消費執行緒數合理設定
- ☐ 批次大小合理設定
- ☐ 重試次數合理設定
- ☐ 消費位點策略合適

8.4 叢集最佳化

- ☐ Topic 佇列數合理設定
- ☐ 主從複製策略合適
- ☐ 刷盤策略合適
- ☐ 叢集規模滿足業務需求
- ☐ 節點部署策略合適

全國統一服務熱線

4008-555-800

金蝶天燕雲計算股份有限公司（簡稱“金蝶天燕”）成立於2000年，前身為“金蝶中間件公司”，是金蝶集團旗下基礎軟件平台提供商，雲計算國家標準制定企業，國家信創產業核心軟件企業。金蝶天燕是國家863重點研發計劃與核高基重大專項承接企業，也是“兩網一站四庫十二金”國家重點工程的基礎平台提供商，產品廣泛應用於政府、軍工、金融、能源等關鍵行業，累計服務客戶總數超過10萬家。



雲計算國家標準制定企業

金蝶集團旗下基礎軟件企業

資訊技術應用創新核心企業

官網：www.apusic.com

