Apusic 金蝶天燕

生成時間：2026-10-19

性能優化手冊

金蝶Apusic分布式缓存 V2.0.6

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

Apusic 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 技術支持
- 2 性能瓶頸來源
  - 2.1 內存管理配置
    - 2.1.1 內存管理優化點
    - 2.1.2 優化方式
      - 2.1.2.1 預留空間
      - 2.1.2.2 調整淘汰策略
    - 2.1.3 驗證方式
  - 2.2 持久化配置
    - 2.2.1 持久化優化點
      - 2.2.1.1 優化方式
      - 2.2.1.2 場景化策略
  - 2.3 網絡配置
    - 2.3.1 優化點
  - 2.4 系統配置

1 前言

本文檔為金蝶Apusic分佈式緩存（AMDC）V2.0.6的產品性能優化手冊，旨在為AMDC V2.0.6用戶提供全面、系統的性能優化指導。通過深入分析AMDC自身配置的性能瓶頸，結合典型業務場景的優化案例幫助用戶解決在高並發環境下可能遇到的性能問題。

文檔內容涵蓋了從基礎的系統參數調優到複雜的百萬級並發連接優化，從常見的超時配置到關鍵的長連接復用策略，以及在性能測試過程中可能遇到的各種問題排查方法。

通過本文的指導，用戶可以：

- 識別和解決AMDC在高並發場景下的性能瓶頸
- 合理配置操作系統和AMDC參數以提升系統整體性能
- 快速定位和解決性能測試中遇到的常見問題
- 根據具體業務場景實施相應的優化方案

1.1 適用對象

本文檔適用於AMDC產品運維工程師、IT系統運維工程師、開發工程師、軟件架構師及研發經理等人員。

1.2 相關文檔

了解更多AMDC V2.0.6產品相關的信息，請參閱以下AMDC V2.0.6產品手冊文檔集：

序號	手冊文檔	說明
1	金蝶Apusic分佈式緩存 V2.0.6 快速使用手冊	簡單介紹了如何快速上手使用AMDC。
2	金蝶Apusic分佈式緩存 V2.0.6 安裝手冊	詳細介紹如何在各操作系統上安裝AMDC，以及AMDC服務啟停操作，產品的註冊過程。
3	金蝶Apusic分佈式緩存 V2.0.6 緩存核心用戶手冊	詳細介紹 AMDC 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分佈式緩存 V2.0.6 管控台用戶手冊	詳細介紹AMDC管控台相關功能的使用和操作說明。
5	金蝶Apusic分佈式緩存 V2.0.6 開發手冊	詳細介紹基於各開發語言進行AMDC客戶端應用開發的說明。

6	金蝶Apusic分佈式緩存 V2.0.6 遷移手冊	詳細介紹AMDC歷史版本遷移升級到V2.0.6版本的說明，以及Redis遷移到AMDC的說明。
7	金蝶Apusic分佈式緩存 V2.0.6 運維手冊	詳細介紹AMDC的監控、運維、安全加固等運維說明。
8	金蝶Apusic分佈式緩存 V2.0.6 性能優化手冊	詳細介紹AMDC性能調優的說明。

1.3 技術支持

AMDC產品提供全面的技術支持服務，您可以通過以下方式獲得技術支持：

- 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

您在取得技術支持時，請提供如下信息：

1. 您的姓名
2. 公司信息與聯繫方式
3. 操作系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細信息

2 性能瓶頸來源

AMDC作為高性能內存數據庫，其性能表現受多維因素制約，需從系統性視角審視瓶頸來源與優化路徑，性能瓶頸主要呈現為四大維度：

- 內存維度是核心制約點，表現為物理內存不足觸發淘汰機制、大Key操作引發阻塞、內存碎片導致利用率下降以及fork操作時的寫時複製開銷，這些因素共同構成內存密集型應用的性能天花板。
- 網絡維度在高並發場景尤為關鍵，連接數受限於內核參數與文件描述符，緩衝區設置不當影響吞吐量，網絡延遲與帶寬瓶頸則直接制約跨節點通信效率。
- CPU維度受單線程架構約束，複雜命令執行、持久化fork操作、主從同步及鍵過期清理等計算密集型任務均會引發主線程阻塞，導致延遲波動。

接下來我們會從內存管理，持久化配置，網絡配置，系統配置幾個方面介紹AMDC調優思路。

2.1 內存管理配置

2.1.1 內存管理優化點

- AMDC 所有數據駐留內存，若用光可用 RAM，將觸發 OOM（Linux 直接 kill 進程）或引發劇烈 swap，QPS 驟降。
- 熱點數據往往只佔 20% 卻承擔 80 % 請求，把有限內存讓給熱鍵，可最大化緩存命中率、降低平均延遲。
- 操作系統自身、後台備份、AOF rewrite、網絡緩衝區同樣需要 30%-40% 空閒內存，否則 fork 時瞬間雙倍內存開銷會導致失敗。

2.1.2 優化方式

2.1.2.1 預留空間

根據服務器內存總量合理設置，建議預留 30%-40% 內存給操作系統及其他進程，避免內存溢出。例如，若服務器內存為 `16GB`，可將 `maxmemory` 設為 `10GB`，配置語句為 `maxmemory 10gb`。

2.1.2.2 調整淘汰策略

根據業務數據特性選擇合適的內存淘汰策略。當業務中熱點數據集中，且需優先保留熱點數據時，推薦使用 `volatile-lru`（淘汰設置了過期時間的鍵中最近最少使用的）或 `allkeys-lru`（淘汰所有鍵中最近最少使用的）。

若業務對數據過期時間敏感，可選擇volatile-ttl（淘汰設置了過期時間的鍵中剩餘時間最短的）。配置示例：`maxmemory-policy volatile-lru`。

2.1.3 驗證方式

```
#内存用量
amdc-cli INFO memory | grep used_memory_human
#觀察 ≤ 10 GB；同時 used_memory_rss_human 與 used_memory_human 差值 <
200 MB（碎片低）。

#淘汰計數
amdc-cli INFO stats | grep evicted_keys
#若持續增長且命中下降，說明 10 GB 偏小，可調大或再評估熱鍵比例。

#命中率
amdc-cli INFO stats | grep keyspace_hits
#keyspace_hits / (hits + misses) ≥ 95 % 為健康。

#延遲毛刺
amdc-cli --latency -p 6359
#p99 ≤ 1 ms（本機萬兆網卡、SSD 下）。
```

2.2 持久化配置

2.2.1 持久化優化點

RDB持久化主要問題：

- 數據丟失窗口：兩次快照間宕機會丟失期間所有寫入
- 磁盤I/O峰值：保存期間可能產生大量磁盤寫入

AOF持久化主要問題：

- AOF重寫膨脹：BGREWRITEAOF期間雙寫壓力（舊AOF追加 + 新AOF重寫）
- fsync性能損耗：appendfsync always 性能下降約90%

2.2.1.1 優化方式

關於RDB智能保存觸發（避免頻繁bgsave），需要我們選擇合適的觸發保存的條件

```
save 900 1      # 15分鐘至少1個key變化
save 300 10     # 5分鐘至少10個key變化
```

```
save 60 10000  # 1分鐘至少10000個key變化
```

反之，AOF的保存，我們則更需要關注重寫時候的壓力和刷盤的頻率。

```
# 1. 合理的fsync策略（95%場景選everysec）
appendfsync everysec  # 平衡性能與安全
# appendfsync no      # 僅當可承受分鐘級數據丟失時使用

# 2. 自動重寫控制
auto-aof-rewrite-percentage 100   # 增長100%時觸發重寫
auto-aof-rewrite-min-size 64mb    # AOF文件最小重寫閾值

# 3. 優化重寫過程
no-appendfsync-on-rewrite yes     # 重寫期間不fsync，避免雙重阻塞
aof-rewrite-incremental-fsync yes # 增量式同步，減少單次fsync數據量
```

或採取更中和的方式， 混合持久化（RDB + AOF），此模式能夠在盡量接近AOF級別的數據安全的情況下，有著較於aof模式更小的磁盤佔用，屬於一種平衡aof和rdb兩者優缺的方式。

```
aof-use-rdb-preamble yes  # 開啟混合模式
# 配合以下參數:
aof-rewrite-incremental-fsync yes
rdbcompression yes        # RDB部分仍壓縮
```

2.2.1.2 場景化策略

緩存型應用（數據可重建） 推薦策略：僅RDB 配置示例：

```
save 3600 1       # 每小時備份一次
stop-writes-on-bgsave-error no  # 允許寫入繼續
```

但是這是建立在恢復速度快（毫秒級，磁盤佔用最小化，數據丟失1小時可接受的情況下）。 與此同時我們還需要配合多級緩存（本地緩存+amdc），並設置過期時間自動淘汰。

金融交易類（零數據丟失要求） AOF(always) + RDB冷備 配置示例：

```
appendfsync always
auto-aof-rewrite-percentage 50   # 更頻繁重寫減少恢復時間
aof-rewrite-min-size 32mb
```

此時我們可以考慮使用一些性能補償的手段來彌補數據安全帶來的性能下降，比如使用性能更好的磁盤，使用主從架構，在從節點完成持久化操作。

2.3 網絡配置

2.3.1 優化點

為了達到長期穩定運行的目標，我們的網絡配置優化不僅會從性能瓶頸處理，以及高可用、故障恢復等方面的配置優化上加以說明。

網絡穩定離不開內核參數優化，以及節點之間的通信正常。

下列是網絡配置的幾個基礎且關鍵的配置，需要結合內核參數調整，並結合業務需要調整主從或者集群節點間的通信配置。

```
maxclients 10000                    # 最大客戶端連接數（需結合系統限制）
tcp-backlog 511                     # 半連接隊列長度，必須 ≤ somaxconn
timeout 300                         # 空閒連接超時（秒），0表示禁用
repl-timeout 60                     # 主從複製超時
cluster-node-timeout 15000          # 集群節點超時（毫秒）
```

在高可用場景下，我們需要調整主從節點之間交互的一些參數配置來保障整個服務系統的穩定。其中我們大多數時候需要根據不同的業務和內存需要調整 repl-backlog-size，過大會佔用內存，過小可能會使得增量同步不完全從而導致全量同步，可能導致cpu佔用和內存過高。

```
# 主節點配置
repl-backlog-size 512mb             # 複製積壓緩衝區（建議：內存的10%）
repl-backlog-ttl 3600               # 緩衝區保留時間 (1h)
repl-disable-tcp-nodelay no         # 啟用Nagle算法，減少包數

# 從節點配置
repl-ping-replica-period 10         # 從節點ping主節點頻率
repl-timeout 60                     # 複製超時時間（需>ping週期×3）
```

同時需要注意哨兵的配置上 哨兵數量公式（保證仲裁）： 哨兵數量 = 2N + 1 (N為容忍的故障哨兵數) 最小推薦：3個哨兵，容忍1個故障

同時哨兵的部署不同於服務，我們需要獨立部署它，不能和amdc 實例同機，導致單點故障。 以下是哨兵配置的參數建議：

```
# 1. 監控配置（每個哨兵都需要）
sentinel monitor mymaster 10.0.1.100 6379 2
# 格式：sentinel monitor <主節點名稱> <IP> <端口> <法定票數>

# 2. 主觀下線判斷
sentinel down-after-milliseconds mymaster 30000
# 值說明：
# - <10000ms（內網低延遲環境）
# - 30000ms（標準生產環境）
# - >60000ms（跨機房或高延遲網絡）

# 3. 故障轉移超時控制
sentinel failover-timeout mymaster 180000
# 作用：控制故障轉移各階段的超時時間

# 4. 並行同步限制
sentinel parallel-syncs mymaster 1
# 值說明：
# - 1：最安全，新主節點壓力小
# - 2-3：平衡同步速度與主節點負載
# - >3：可能導致新主節點過載

# 5. 節點通信優化
sentinel resolve-hostnames no       # 禁用DNS解析，使用IP
sentinel announce-ip 10.0.1.200     # 哨兵通告IP
sentinel announce-port 26379        # 哨兵通告端口
```

使用集群時，需注意節點之間的超時時間設置，一般採取默認值即可，不宜過小，導致頻繁進行集群檢測，也不宜過大，導致節點宕機卻未能監測。 cluster-announce-ip 是在集群部署中明確指定節點對外服務的IP

```
# 集群節點配置
cluster-enabled yes
cluster-config-file nodes.conf
cluster-require-full-coverage no    # 避免少數槽不可用導致全集群不可用

# 節點超時與心跳
cluster-node-timeout 15000          # 節點超時時間（影響故障轉移）
cluster-replica-validity-factor 10  # 從節點數據有效性因子
cluster-announce-port 6379
cluster-announce-bus-port 16379     # 集群總線端口（端口+10000）

# 跨機房部署優化
cluster-announce-ip 192.168.1.100   # 明確指定節點IP
```

典型症狀與解決方法：

```
# 症狀
[ERR] Node ... is not empty. Either the node already knows other
nodes

# 解決步驟：
1. 檢查每個節點的 cluster-announce-ip 是否正確
2. 確保所有端口在防火牆中開放（6379和16379）  #此問題在大多數默認部署時會有遺漏，需要留意。
3. 清除舊集群配置：rm nodes.conf，然後重新創建集群
```

2.4 系統配置

為了緩存系統的穩定高效，有些時候我們需要將調優的範圍延伸到系統級別。 以下的參數調整可供參考。

1. vm.overcommit_memory - 內存超配策略

```
# 當前值查看
cat /proc/sys/vm/overcommit_memory
```

```
# 永久修改（/etc/sysctl.conf）
vm.overcommit_memory = 1
```

一般來說，該參數的默認值為0，它的含義是內核檢查可用內存，如果不足可能拒絕分配（可能導致AMDC 服務fork失敗） 而其他值 1 --總是允許分配（即使超出物理內存） 2 --嚴格限制，分配不能超過物理內存 + swap 對此，我們的建議是設置為1，否則AMDC執行bgsave時，fork子進程可能因「內存不足」而失敗（實際不需要雙倍內存，但內核會檢查）

2. 透明大頁（Transparent HugePages）

```
# 檢查狀態
cat /sys/kernel/mm/transparent_hugepage/enabled

# 禁用命令
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag

# 永久禁用（CentOS/RHEL 7+）
grubby --update-kernel=ALL --args="transparent_hugepage=never"
```

建議禁用，否則可能導致bgsave時fork延遲增加數倍，內存使用量可能增加30%-40%，運行中可能產生不可預測的延遲峰值

3. net.core.somaxconn - 連接隊列大小

```
# 查看當前值
cat /proc/sys/net/core/somaxconn

# 永久修改
net.core.somaxconn = 65535
```

對於這個參數，我們的建議是增大，因為AMDC的tcp-backlog配置（默認511）受此參數限制，高並發時，如果隊列太小，客戶端會收到「Connection refused」。典型場景：秒殺活動時，大量連接瞬間到達。

4. ulimit文件描述符限制

```
# 查看AMDC進程限制
cat /proc/$(pidof amdc-server)/limits | grep "open files"

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

該參數的默認值通常為1024，其作用為限制單個進程能打開的文件數（每個連接佔用一個文件描述符），我們建議增大，因為：每個客戶端連接消耗1個文件描述符，且持久化也會打開文件（RDB/AOF）默認1024只能支持約1000個並發連接。

生產建議：至少65535，大型集群可能需要更高。也存在現在大多數客戶端都自帶連接池，實際的消耗需要具體情況具體分析。

5. 網絡緩衝區調整

```
# 永久修改 (/etc/sysctl.conf)
net.core.rmem_max = 134217728
net.core.wmem_max = 134217728
net.ipv4.tcp_rmem = 4096 87380 134217728
net.ipv4.tcp_wmem = 4096 65536 134217728
```

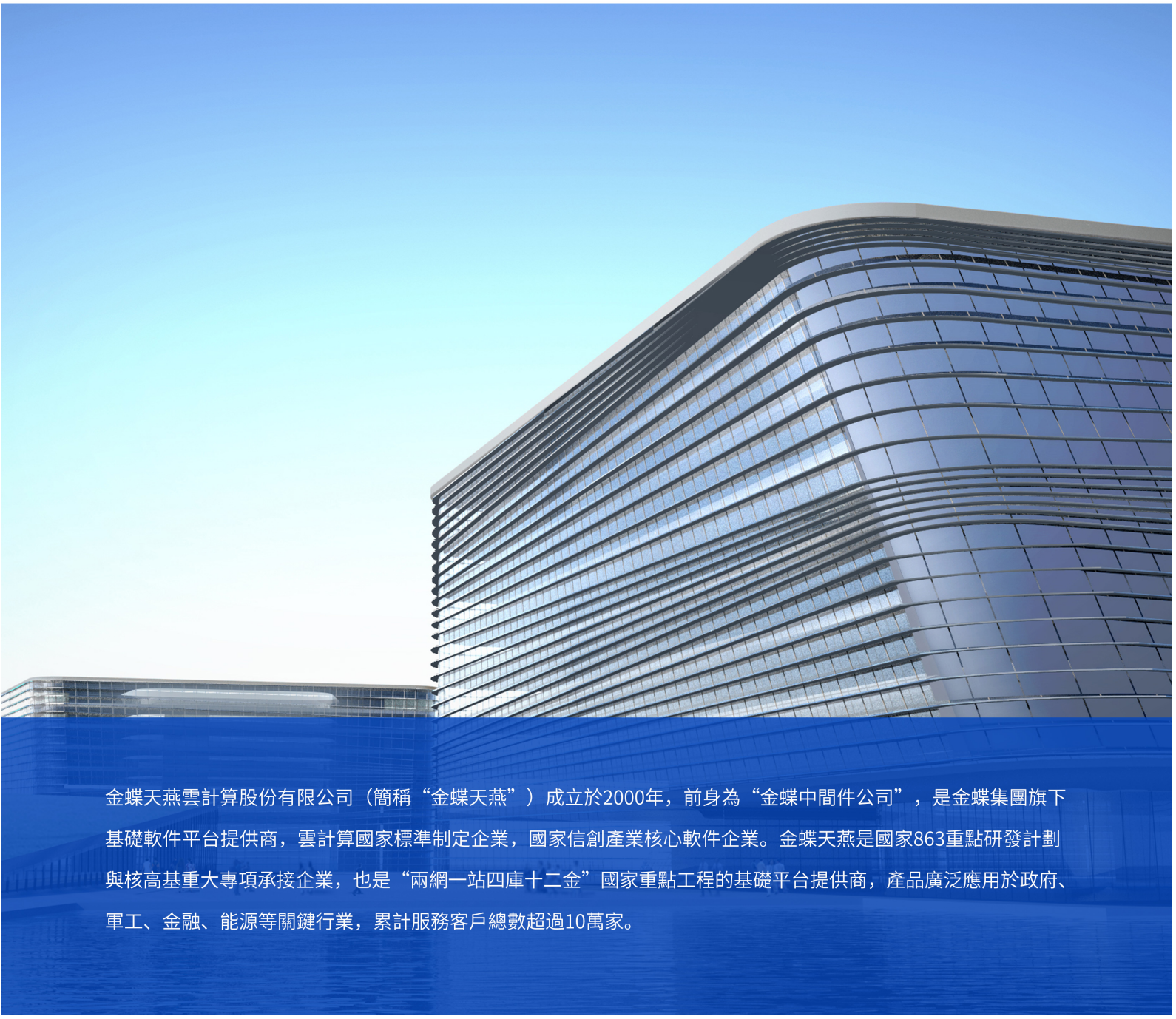
上述的參數作用主要是用於控制TCP發送和接收緩衝區大小 因為默認值一般較小，所以我們建議是增大，因為我們的服務可能需要返回大量的數據，其中要麼是大value，要麼是批量操作，緩衝區太小的話會導致TCP窗口的縮小，降低吞吐量，尤其是在高帶寬、高延遲的網絡中更加重要。

生產建議：

- rmem_max/wmem_max：`128MB`
- tcp_rmem/tcp_wmem：最小`4KB`，默認`64KB/85KB`，最大`128MB`

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。

Apusic 金蝶天燕

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