**實例講解JVM參數調優的八條經驗**

要想配置好JVM參數，需要對年輕代、年老代、救助空間和永久代有一定瞭解，還要瞭解jvm內存管理邏輯，最終還要根據自己的應用來做調整。關於JVM參數上網一搜就能搜出一大把，也有很多提供實踐的例子，我也按照各種例子測試過，最終還是會出現問題。

經過幾個月的實踐改善，我就網站（要求無停滯時間）的jvm參數調優給出以下幾條經驗。

1：建議用64位操作系統，Linux下64位的jdk比32位jdk要慢一些，但是吃得內存更多，吞吐量更大。

2：XMX和XMS設置一樣大，MaxPermSize和MinPermSize設置一樣大，這樣可以減輕伸縮堆大小帶來的壓力。

3：調試的時候設置一些打印參數，如-XX:+PrintClassHistogram -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+PrintHeapAtGC -Xloggc:log/gc.log，這樣可以從gc.log裡看出一些端倪出來。

4：系統停頓的時候可能是GC的問題也可能是程序的問題，多用jmap和jstack查看，或者killall -3 java，然後查看java控制台日誌，能看出很多問題。有一次，網站突然很慢，jstack一看，原來是自己寫的URLConnection連接太多沒有釋放，改一下程序就OK了。

5：仔細瞭解自己的應用，如果用了緩存，那麼年老代應該大一些，緩存的HashMap不應該無限制長，建議採用LRU算法的Map做緩存，LRUMap的最大長度也要根據實際情況設定。

6：垃圾回收時promotion failed是個很頭痛的問題，一般可能是兩種原因產生，第一個原因是救助空間不夠，救助空間裡的對象還不應該被移動到年老代，但年輕代又有很多對象需要放入救助空間；第二個原因是年老代沒有足夠的空間接納來自年輕代的對象；這兩種情況都會轉向Full GC，網站停頓時間較長。第一個原因我的最終解決辦法是去掉救助空間，設置-XX:SurvivorRatio=65536 -XX:MaxTenuringThreshold=0即可，第二個原因我的解決辦法是設置CMSInitiatingOccupancyFraction為某個值（假設70），這樣年老代空間到70%時就開始執行CMS，年老代有足夠的空間接納來自年輕代的對象。

7：不管怎樣，永久代還是會逐漸變滿，所以隔三差五重起java服務器是必要的，我每天都自動重起。

8：採用並發回收時，年輕代小一點，年老代要大，因為年老大用的是並發回收，即使時間長點也不會影響其他程序繼續運行，網站不會停頓。

我的最終配置如下（系統8G內存），每天幾百萬pv一點問題都沒有，網站沒有停頓，2009年網站沒有因為內存問題down過機。

1. $JAVA\_ARGS .= " -Dresin.home=$SERVER\_ROOT -server
2. -Xms6000M -Xmx6000M -Xmn500M -XX:PermSize=500M -XX:MaxPermSize=500M
3. -XX:SurvivorRatio=65536 -XX:MaxTenuringThreshold=0 -Xnoclassgc
4. -XX:+DisableExplicitGC -XX:+UseParNewGC -XX:+UseConcMarkSweepGC
5. -XX:+UseCMSCompactAtFullCollection -XX:CMSFullGCsBeforeCompaction=0
6. -XX:+CMSClassUnloadingEnabled -XX:-CMSParallelRemarkEnabled
7. -XX:CMSInitiatingOccupancyFraction=90 -XX:SoftRefLRUPolicyMSPerMB=0
8. -XX:+PrintClassHistogram -XX:+PrintGCDetails -XX:+PrintGCTimeStamps
9. -XX:+PrintHeapAtGC -Xloggc:log/gc.log ";

說明一下， -XX:SurvivorRatio=65536 -XX:MaxTenuringThreshold=0就是去掉了救助空間；

-Xnoclassgc禁用類垃圾回收，性能會高一點；

-XX:+DisableExplicitGC禁止System.gc()，免得程序員誤調用gc方法影響性能；

-XX:+UseParNewGC，對年輕代採用多線程並行回收，這樣收得快；

帶CMS參數的都是和並發回收相關的，不明白的可以上網搜索；

CMSInitiatingOccupancyFraction，這個參數設置有很大技巧，基本上滿足(Xmx-Xmn)\*(100-CMSInitiatingOccupancyFraction)/100>=Xmn就不會出現promotion failed。在我的應用中Xmx是6000，Xmn是500，那麼Xmx-Xmn是5500兆，也就是年老代有5500兆，CMSInitiatingOccupancyFraction=90說明年老代到90%滿的時候開始執行對年老代的並發垃圾回收（CMS），這時還剩10%的空間是5500\*10%=550兆，所以即使Xmn（也就是年輕代共500兆）裡所有對象都搬到年老代裡，550兆的空間也足夠了，所以只要滿足上面的公式，就不會出現垃圾回收時的promotion failed；

SoftRefLRUPolicyMSPerMB這個參數我認為可能有點用，官方解釋是softly reachable objects will remain alive for some amount of time after the last time they were referenced. The default value is one second of lifetime per free megabyte in the heap，我覺得沒必要等1秒；

網上其他介紹JVM參數的也比較多，估計其中大部分是沒有遇到promotion failed，或者訪問量太小沒有機會遇到，(Xmx-Xmn)\*(100-CMSInitiatingOccupancyFraction)/100>=Xmn這個公式絕對是原創，真遇到promotion failed了，還得這麼處理。

JVM配置參數中文說明：

-----------------------------------------------------------------------

1、-Xmixed mixed mode execution (default)

混合模式執行

2、-Xint interpreted mode execution only

解釋模式執行

3、-Xbootclasspath:<directories and zip/jar files separated by ;>

set search path for bootstrap classes and resources

設置zip/jar資源或者類（.class檔）存放目錄路徑

3、-Xbootclasspath/a:<directories and zip/jar files separated by ;>

append to end of bootstrap class path

追加zip/jar資源或者類（.class檔）存放目錄路徑

4、-Xbootclasspath/p:<directories and zip/jar files separated by ;>

prepend in front of bootstrap class path

預先加載zip/jar資源或者類（.class檔）存放目錄路徑

5、-Xnoclassgc disable class garbage collection

關閉類垃圾回收功能

6、-Xincgc enable incremental garbage collection

開啟類的垃圾回收功能

7、-Xloggc:<file> log GC status to a file with time stamps

記錄垃圾回日誌到一個檔。

8、-Xbatch disable background compilation

關閉後台編譯

9、-Xms<size> set initial JAVA heap size

設置JVM初始化堆內存大小

10、-Xmx<size> set maximum JAVA heap size

設置JVM最大的堆內存大小

11、-Xss<size> set java thread stack size

設置JVM棧內存大小

12、-Xprof output cpu profiling data

輸入CPU概要表數據

13、-Xfuture enable strictest checks, anticipating future default

執行嚴格的代碼檢查，預測可能出現的情況

14、-Xrs reduce use of OS signals by JAVA/VM (see documentation)

通過JVM還原操作系統信號

15、-Xcheck:jni perform. additional checks for JNI functions

對JNI函數執行檢查

16、-Xshare:off do not attempt to use shared class data

盡可能不去使用共用類的數據

17、-Xshare:auto use shared class data if possible (default)

盡可能的使用共用類的數據

18、-Xshare:on require using shared class data, otherwise fail.

盡可能的使用共用類的數據，否則運行失敗

The -X options are non-standard and subject to change without notice.