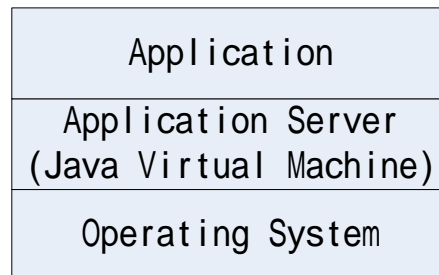


如何分析 Java 應用系統效能與問題

Java 應用系統效能對系統管理者而言是一個很重要的課題，在現今的多層式系統架構下系統環境複雜，應用系統的效能變慢或是發生了問題影響了系統運作服務，如何有效且快速找出應用系統效能瓶頸與問題點，這對系統管理者而言是一大考驗。

在面對處理這類似的問題，無法單純就應用系統的 Log 或是 Server 的 Log 可以找到解答的，這個必需要有三方面的資訊，如下圖所示：



圖(一) 系統架構

因為應用系統運作不外乎是由上面三個元素組成，因此其中一個元素的效能變慢或是發生問題，都會影響到整個應用系統的運作；一般而言這三個元素的資訊都是以文字檔案方式記錄儲存，分析效能問題是將這三個元素的訊息一一加以解讀，才能找出真正的效能問題點。

除了上述的分析方法外，另外就是透過系統效能監看工具協助問題的分析，各項資訊都被整合在統一的介面上，且記錄的訊息以圖形化方式呈現，可以有效且快速定出問題點。若能整合上述的工具，在面對效能問題分析時可以快速且有效找出問題點；**Quest Application Management** 正是這樣的產品組合，讓我們可以在分析效能問題時透過圖形化的介面，找出效能的瓶頸與問題點。

在本篇文章將以我們平時協助客戶找出效能分析的步驟與方法為出發點，說明我們是如何分析效能與問題點，通常是會配合使用一些相關的系統監看分析工具(如：HP J Tune、IBM GA、Quest Foglight 等)，以協助我們分析所得到的相關訊息，迅速的找到並確認效能問題點，進而快速排解問題。

分析前的資料準備

相關的訊息是很重要的，缺少這些相關的訊息即便有工具可以使用，但也是無法做分析系統效能與問題；在圖(一)可以清楚知道應用系統的組成有分三個元素，因此分析問題的第一個步驟必需要產生這三個元素的相關訊息，而這三個元素要產生的資訊要有那些才足夠呢？在下面一一列出來：

- 1 應用程式
 - 1.1 Log Files
Application 所產生的相關運作訊息。
 - 1.2 架構
- 2 應用伺服器(Java Virtual Machine)
 - 2.1 Server
 - 2.1.1 Log Files
Application Server 的運作訊息
 - 2.1.2 Server Configuration
Application Server 的系統設定檔案
 - 2.2 Java Virtual Machine
 - 2.2.1 Hotspot arguments
Java Virtual Machine 所設定的記憶體大小與相關參數資訊
 - 2.2.2 GC Log Files
GC 運作訊息透過設定將訊息輸出至一個檔案
 - 2.2.3 Thread Dumps
產生 Java Virtual Machine 的 Thread 的運作狀況
- 3 作業系統
 - 3.1 CPU Utilization
 - 3.2 Memory Utilization

上述的資訊是用來分析系統運作狀況，應用程式的 Log 與架構資訊是要了解系統運作方式，以了解可能會造成系統瓶頸與效能不佳的原因，其他的資訊為應用程式的執行環境運作資訊，分析這些資訊是為了確保應用程式是在一個運作狀況良好的執行環境下執行；GC Log 與 Thread Dump 的資訊有些是必需透過參數設定的方式讓它固定產生出，作業系統的 CPU 與 Memory 的使用狀況是要透過系統記錄的方式或是透過效能監看工具(如 Quest Foglight)產生該項資訊，而應用伺服器的 Log 與 Configuration 為系統運作時本身就會俱備的資訊，分析上述資訊，以確認運作環境的執行效能達到最佳狀況，以提供應用程式在上面執行運作。

開始分析

準備好上面的資訊後則開始進入效能分析，我們會先分析底層的運作，主要是底層的運作效能好，則在上面執行的應用程式至少也會有一定的運作效能，反之連底層都運作效能不佳，則在上面執行的應用程式可能也不會有良好的效能，首先會先從 JVM 開始分析。

JVM 分析

JVM 的運作效能我們是以其 GC 的運狀況來衡量其運作效能，並配合作業系統的 CPU 與 Memory 的使用狀況，以了解整個 JVM 的運作是否已達最佳。分析 JVM 的效能，從實務的經驗做出下面幾點的分析準則：

- 1 JVM 的 Total Heap 的設定應為不得超過實體記憶體體的 3/4，一般實務而言盡量讓作業系統能有至少保有 512MB 的記憶體，以確體作業系統運作正常，這能才能讓 JVM 運作正常則 Application Server 也才能有好的運作效能。
- 2 JVM 的 New Size 的設定應為 Total Heap 的 1/4 至 1/2 間，不得高於 1/2 否則將會造成 JVM 運作不穩定的問題。我們實務上都設定為 1/3，而 JVM 預設(-server 模式)為 1/4。
- 3 Minor GC 的時間應維持小於 1 秒，而 Major(Full) GC 最佳的時間為低於 3 秒，若高於 3 秒但小於 5 秒則屬於一般正常狀況；而 Heap 的使用最好能保有約 30%左右的剩餘空間確保穩定運作。
- 4 Full GC 發生的次數應愈低愈好，最好不得高於 Minor GC 發生的次數；Full GC 的次數過高也會影響到 CPU 的 Loading 會升高；若 Full GC 的次數高於 Minor GC，則代表 Heap 的空間配置不佳，應要調整空間配置，可以參考準則 2 的調整。
- 5 若 Full GC 的時間過長，則可以調整 GC 的運作模式，實務經驗會採用 through put 與 low pause 兩種運作模式，兩種運作模式都是以 Parallel 的方式行 GC。採用這兩種模式將會影響到 CPU 的 Loading 會略微升高，一般實務觀察統計約在 10%以下。

分析 JVM 的 GC 運作效能，由於 GC Log 所產生的內容是以文字檔案儲存，因此可以直接透過文字編輯器開啓進行分析，由於 GC 所產生資訊的多寡將會隨著 JVM 執行時間的長短而不同，一般而言系統運作時間夠長的話，通常 GC Log 的資訊將會超過幾千行，如果單用文字編輯器分析將是件很累人的事，當然 JVM 各家廠商都有慮到這一點，因此都有提供相對應的分析工具，以協助做 GC 的運

作效能分析,如 HP Jtune、IBM GA、Sun JVMStat 與 Open Source 的 GC Viewer 等都是相關的 GC 分析工具,其相關的下載資訊如下:

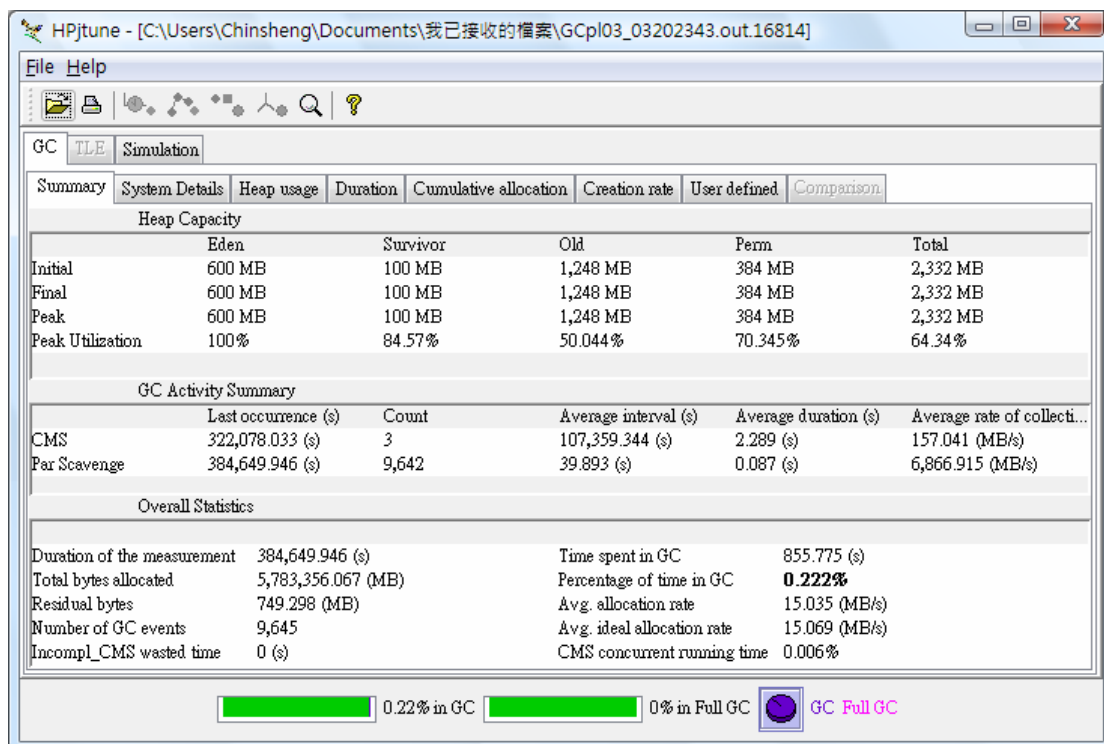
HP JTune: <http://www.hp.com/products1/unix/java/java2/hpjtune/>

IBM GA: <http://www.alphaworks.ibm.com/tech/pmat>

Sun JVMStat: <http://java.sun.com/performance/jvmstat/>

Tagtraum GCViewer: <http://www.tagtraum.com/gcviewer-download.html>

分析 GC Log 訊息是衡量 Minor/Full GC 的次數與 GC 的平均花費時間,以這兩個數據做為 JVM 運作效能分析依據,按照上述準則 3 與 4 並配合分析工具來衡量 GC 的運作狀況 (如圖二與圖三),若能達準則 3 與 4 的條件,則代表 JVM 的運作效能是良好的,若未能達到準則 3 與 4 的條件,則代表 GC 的運作是有校調的空間,因此可以再做 GC 運作模式的調整,可以按照準則 5 的兩種 GC 運作模式進行調校;調整完 GC 運作模式後,仍需繼續觀察 GC 的運作狀況確定能符合準則 3 的條件,這樣便是完成 GC 的校調。



圖(二) 用 HP JTune 分析 GC 運作效能

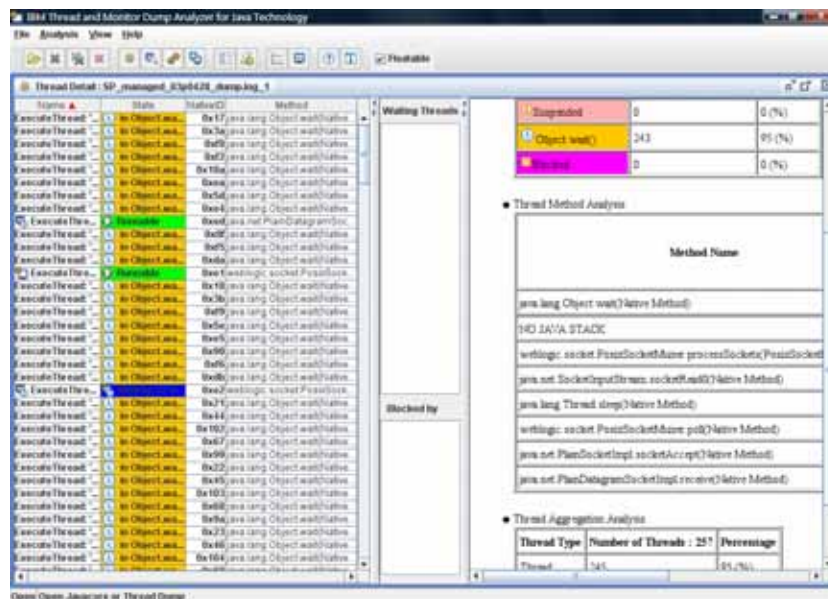


圖(三) Foglight 監看系統資源

另外 Thread Dump 的分析，主要是分析當 CPU 的 Loading 過高時，了解整個 JVM 是在執行那些程式，而 Thread Dump 資訊與 GC Log 一樣都是以文字檔案方式儲存，因此分析時也可以直接打開文字編輯器進行分析，當然也有可以分析工具可以協助做分析，如下：

IBM JCA:

http://alphaworks.ibm.com/tech/jca?open&S_TACT=106AH21W&S_CMP=AWLP



圖(四) IBM JCA 分析工具

May 08 M-Power eNew

本篇文章版權為倍力資訊股份有限公司所有，未經書面同意，嚴禁複製、轉載

應用伺服器分析

當 JVM 與作業系統相關的效能已調整後，再來便是分析應用伺服器上的運作效能，通常要監看應用伺服器的各項資源運作效能，通常可以直接透過應用伺服器的監看功能了解其運作狀況，最方便的方式是採用應用伺服器效能監看工具協助分析應用伺服器效能，例如：Quest Foglight 或 PerformaSure，透過監看工具記錄整個應用伺服器的各項資源的狀況，在監看分析工具上分析整個應用伺服器的各項資源運作狀況與效能，以確定其各項資源的運作效能，如下圖(二)是透過 Quest Foglight 監看 Server 運作狀態，可以在其監控畫面中馬上看到那一項系統資源效能不佳，並且發出警訊(顯示黃燈)讓我們知道該項資源運作狀況不佳的。



圖(五) Foglight 監看畫面

透過 Quest Foglight 監看工具了解是屬於那一個資源效能不佳，如圖(五)所示，Foglight 警示在應用伺服器的 JDBC Connection 中的 Statement Cache 的 Miss Rate 達 83%，Miss Rate 過高將會影響到對資料庫存取的效能，一般實務建議 Miss Rate 最好是維持在 20%以下，因此要對應用伺服器的 Connection Pool 進行調整，以提升系統效能。應用伺服器上有許許多多的資源，如：EJB、JMS、JTA、等等，這些都是應用程式運作所會使用到的資源，若資源運作狀況

不佳也會影響到整個系統的運作，因此讓各項資源能達到一定的效能，這樣便可以有效提升系統效能。

結語

經本文說明後分析應用系統效能與做適當的校調不是件難事，是可以靠系統的 Log 資訊與相關系統資訊進行分析，可以找出系統效能問題點；若能有系統監看分析工具(如：Quest Foglight 與 PerformaSure)的配合使用，則可以讓分析應用系統效能更佳容易，也能很快速找到真正的效能問題點；另外善用系統效能監看工具所提供預警的功能，警示出效能不佳的系統資源，以及早知道系統效能瓶頸，做適當調整以及早發現效能問題並及早排解，以維持應用系統穩定並且提供一個運作良好的應用系統。