

新手學 SQL Server 2008「資源管理員(Resource Governor)」

(1)

作者：陳俊宇
審稿：楊先民
文章編號：S100109502
出刊日期：2010/01/19

對於資料庫管理系統而言，基本上，只要通過帳戶與權限的檢核後，系統就該全力執行所交付的任務，回傳其執行的結果，但在真實世界裡，僅透過帳戶密碼機制來進行管理，肯定是不足的。以下提供幾個案例，例如：

- 前端程式沒有進行最佳化，因故造成浪費系統資源的問題。
- 早上 08:00 資料庫系統要對外提供營運，但是系統維護作業、轉檔作業尚未完成，並且耗用了大量的系統資源。
- 如何面對過度耗用資源程式的執行？砍掉？
- 如何確保指定的程式有足夠的系統資源可供使用？
- 如何面對「阻絕服務攻擊(Denial of Service, DoS)」？

在 SQL Server 2008 版本，新增加了「資源管理員 (Resource Governor)」功能，讓資料庫管理師可以有效的控制伺服器上的硬體資源之使用，並對前述的問題，提供了解決方案。

認識「資源管理員 (Resource Governor)」

SQL Server 2008 所提供的「資源管理員」，讓管理人員可以配置 CPU 與記憶體資源給各個應用程式來使用。讓硬體資源可以被有效的運用。

「資源管理員」提供以下的功能：

- 在「資源集區」內，設定可使用的 CPU 使用量與記憶體配置量。
- 在「工作負載群組」內，可以設定其「相對重要性」。
- 使用「分類函數」，將各個連線(在 SQL Server 內稱為「工作階段 (session)」)，分類到指定的工作群組。
- 可監視在「資源集區」、「工作負載群組」內的資源使用量。

本文相關資訊

資料庫類型：SQL Server 2008

本文相關資訊：

簡介 SQL Server 2008「資源管理員(Resource Governor)」的功能與實作方式。

本文難易等級：

易 難

DB World 資料庫專家電子雜誌

總編：張智凱
主編：楊先民
社務顧問：邱世萍
技術編輯：羅慧真
編輯顧問：許薰尹 羅慧真
胡百敬 許嘉仁 申建忠
楊先民 陳俊宇 高光弘
鄭淑芬 何致億
美術設計：陳淑娟 顏廷廷
行銷企劃：陳秀慧 許克臻
林曉郁 林昕頻
發行服務：謝佩珊
發行所：毅達行銷顧問
股份有限公司
客戶服務：service@dbworld.com.tw



本電子刊物之所載標誌名稱分屬各該公司所有，非經授權請勿轉載使用，版權所有。如經查證依法律追訴。

Oracle OCP 認證最吸金， 幫我薪資加值 **28%**

Oracle OCP的三大好處

1. 更多的工作機會-前進大企業最佳跳板
2. 更高的薪資成長-突破現有薪資門檻
3. 更高的身價保障-持認證人數比例低

企業對OCP人才需求量大
根據IDC 2009年的調查，Oracle資料庫以39%的市佔率遙遙領先28%的IBM及24%的微軟。

現在學，搭上企業升級潮
甲骨文對Oracle資料庫的技術支援即將在2010年7月終止，搶先考取10g及11g OCP來應戰。

選擇恆逸，讓您實力認證兼具
恆逸為原廠授權教育訓練中心，原廠認證師資採用系統化的授課方式，讓您學到最核心的應用技術。

SYSTEMEX making it happen 精誠資訊 | **UCOM 恆逸** 教育訓練中心 | **ORACLE** APPROVED EDUCATION CENTER

所以資料庫管理人員可以利用「資源管理員」來解決資源使用上的相關問題，例如：

- 處理「runaway query(失控的查詢)」。
- 所謂的「runaway query」是指需要耗費長時間來執行的程式，甚至超過「查詢最佳化工具(query optimizer)」所判斷的預期執行時間，造成系統資源被耗盡，嚴重超載。例如：因故造成無窮迴圈運算等。
- 因為「resource contention(資源爭用)」，導致無法預估執行作業的效能。
- 這是因為在伺服器上，多半是混雜不同類型的工作負載，例如：在 OLTP 系統上以為 ERP 為主的程式，與資料倉儲系統上以報表查詢為主的程式，基本上應該要隔離分開，若同時在同一台伺服器執行運算，將造成系統資源的爭用，導致效率不彰與無法預估其執行效能。
- 設定各個工作負載的執行優先權(priority)。
- 在「資源管理員」內，可以指定「工作負載」的「相對重要性」，藉此設定此「工作負載」內的程式之處理速度高於其他「工作負載」的程式，或是在發生資源爭用時，能優先提供資源給此「工作負載」的程式來執行。這裡要說明的是，指定「相對重要性」為「HIGH」時，是指設定提高其處理速度，而不是說讓此程式先執行完成後，才執行「相對重要性」為「MEDIUM」或「LOW」的程式。

「資源管理員」的架構

若是由連線登入、進行分類與運作處理等的觀點來看，其運作處理流程，簡述如下，並請參考下圖 1 所示：

- 應用程式連線登入到 SQL Server 「執行個體」。
- 系統自動利用「分類函數」，對各個「工作階段」進行分類，分配到指定的「工作負載群組」。
- 各個「工作組群組」再依據其上層的「資源集區」所配置的系統資源，提供給各個「工作階段」來使用。

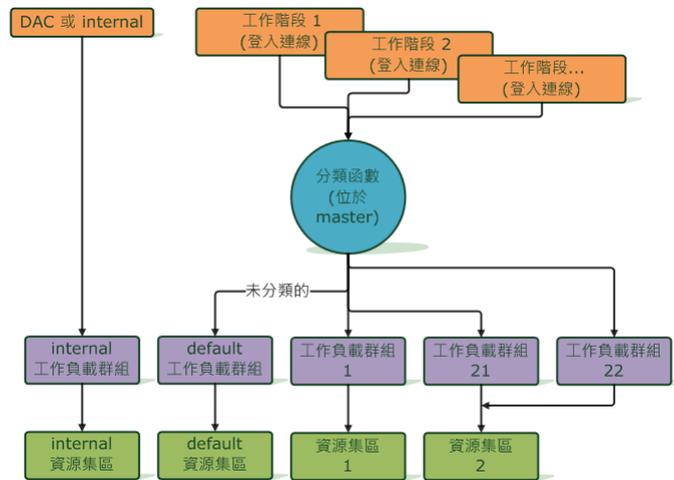


圖 1：「資源管理員」的運作處理流程

有關於建置「資源管理員」的流程，請參考如下：

- 先建立「資源集區」，設定可使用的實體資源。
- 在建立「工作負載群組」，設定其「相對重要性」。
- 最後，建立「分類函數」，用來將各個「工作階段」歸類到指定的「工作負載群組」。

以下為「資源管理員」內相關物件的架構，請參考下圖 2 所示：

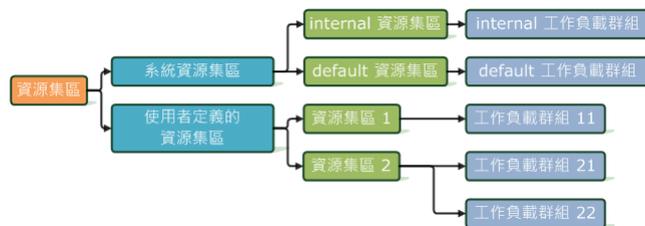


圖 2：「資源管理員」的物件架構

在圖 2 中，資料庫管理師可以建立多個「資源集區」，而在各個「資源集區」下，可以建立多個所需的「工作負載群組」。

認識「資源集區」

所謂的「資源集區(Resource Pool)」，是指此 SQL Server 執行個體可使用的硬體資源，目前可用於分配的硬體資源是：CPU 使用量、記憶體配置量，但尚未包含磁碟 I/O 的使用量。例如：伺服器上有 16 GB 的記憶體，但因故設定此「執行個體」僅能使用其中 12 GB 的記憶體，則在「資源集區」內也僅能對此 12 GB 的記憶體進行分配。

在「資源集區」內，預設已經建立了兩個「系統資源集區」，分別是：「default(預設)」與「internal(內部)」，以及兩個「工作負載群組」，分別是：「default(預設)」與「internal(內部)」，請參考下圖 3 所示：



圖 3：內建的「系統資源集區」與其「工作負載群組」

在「資源集區」部分，可以分成以下的類別：

- 「internal 集區」。
- 「default 集區」。
- 使用者定義的資源集區。

「internal 資源集區」

「internal 資源集區」是指「執行個體」本身所耗用的系統資源，在「internal 資源集區」內，僅包含「internal 工作負載群組」，也不可以變更此「資源集區」設定。在「internal 資源集區」部分，其資源的耗用量是沒有限制的，在此「集區」內任何負載都被視為對 SQL Server 運行所需的資源負載，而且當「internal 資源集區」資源不足時，「資源管理員」將向其他「資源集區」直接施壓索取資源，即便是違反了其他「資源集區」上所設定的資源配置。「internal 資源集區」與「internal 工作負載群組」的資源使用量，與使用者「工作階段」可使用的資源使用量是分開計算的(以百分比方式計算)。請參考下圖 4 所示：



圖 4：不允許改變「internal 資源集區」

「資源管理員」不會對「專用管理員連接(DAC)」進行資源負載控制，有關於「專用管理員連接(DAC)」的「工作階段」部分，則自動隸屬於「internal 資源集區」內的「internal 工作負載群組」。

「default 資源集區」

所謂的「default 資源集區」，是預先定義的第一個「使用者資源集區」。預設在「default 資源集區」內，僅包含一個「default 工作負載群組」。管理人員無法刪除此預設「資源集區」，但是可以變更此「資源集區」的相關設定，在「default 資源集區」下，還可以再建立「使用者定義的工作負載群組」。

設定「資源集區」可用的系統資源

在每個「資源集區」上，設定可使用的資源部分，採取百分比的方式來設定：

- CPU 的 MIN 與 MAX。
- 記憶體 MIN 與 MAX。

要提醒您的是：MIN 值是指對此資源設定最小保證可用的使用量，也就是說，至少可以使用的資源量。MAX

值則是指最大可以使用的資源量。因為 MIN 值是採取保證可用的設計方式，所以在各個「資源集區」內，其 MIN 值加總起來的值，不得超過 100%，請參考下圖 5 所示：



圖 5：各個「資源集區」的 CPU 之 MIN 值，總和超過 100%的限制

在圖 5 中，在 CPU 或是記憶體部分，其 MIN 值都有加總起來不得超過 100%的限制。在 CPU 使用量的 MAX 值部分，則建議設定在 MIN 值與 100%之間的合理值，「資源管理員」對 MAX 值部分，並沒有強制檢查，這是因為 MAX 值部分是指最大可使用的資源量，而非保證可用的資源量。

在設定記憶體的 MAX 值部分，可以使用「可靠性與效能監視器」進行觀察，請參考下圖 6 所示：

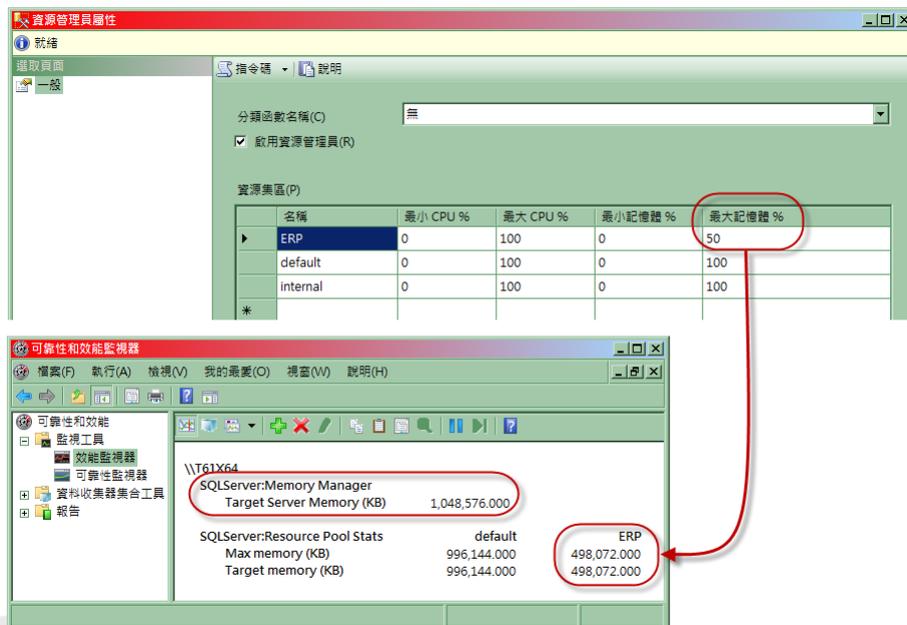


圖 6：使用「效能監視器」觀察設定記憶體的 MAX 值

在圖 6 中，此「執行個體」，在「最大伺服器記憶體」部分，特別設定為 1024 MB，資料庫管理師可以使用效能計數器：Target Server Memory (KB)，觀察 SQL Server 伺服器可用的動態記憶體總數。使用「資源管理員」，建立新的「資源集區」：ERP，並設定記憶體的 MAX 值為 50%，使用在 SQL Server:Resource Pool Stats 物件下的 Max memory (KB)與 Target memory (KB) 效能計數器，將觀察到此「資源集區」：ERP，可擁有的最大記憶體數量(以 KB 為單位)與此「資源集區」正嘗試取得的目標記憶體數量(以 KB 為單位)，差不是「最大伺服器記憶體」的一半。

在設定記憶體的 MIN 值部分，請參考下圖 7 所示：

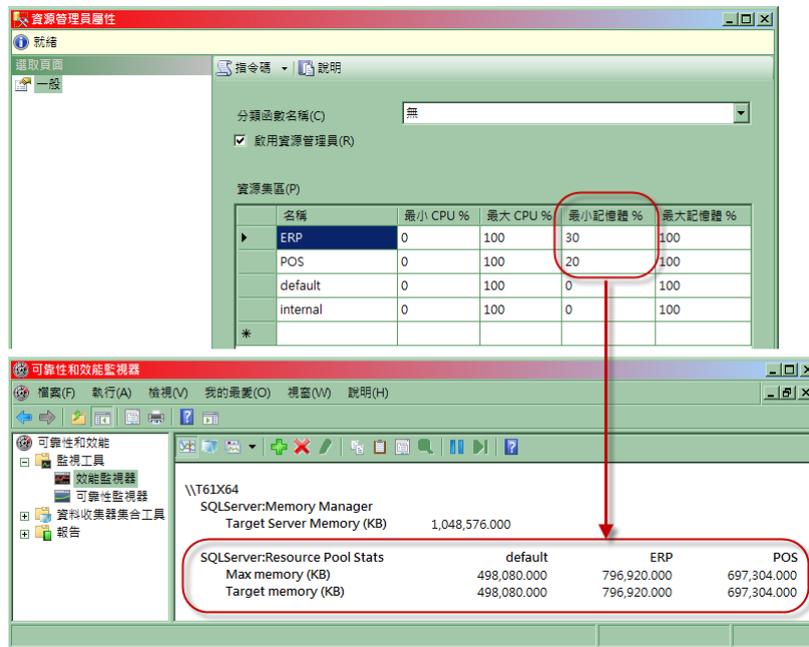


圖 7：使用「效能監視器」觀察設定記憶體的 MIN 值

在圖 7 中，設定此「執行個體」的「最大伺服器記憶體」部分為 1024 MB。使用「資源管理員」，建立新的「資源集區」：ERP 與 POS，並分別設定記憶體的 MIN 值為 30%、20%。利用「可靠性與效能監視器」將可以觀察到「資源集區」：ERP 與 POS，最大可用的記憶體分別約是：780 MB、681 MB，但在「default 資源集區」部分，最大可用的記憶體僅剩下約 487 MB。這是因為 MIN 值是指：對此資源設定最小保證可用的資源量，而在「資源集區」：ERP 與 POS 上，已經分別設定 MIN 值，所以在「default 資源集區」部分，則是剩下： $100 - (30 + 20) = 50(\%)$ 可以使用。

但在圖 7 中，所顯示各個「資源集區」最大可用的記憶體部分，實際上並非此「資源集區」所能夠使用的最大記憶體配置量，這是因為所有的「資源集區」都是由「資源管理員」所控制，而在「資源集區」內，又細分為兩個部分：

- 其中一個是不會與其他「資源集區」重疊使用，這就是設定 MIN 值所達成的最小保證可用之功能。
- 另一個部分，則會與其他「資源集區」共用，用於支援最大可能的資源耗用量。

有關於在「資源集區」共用的部分，是指可用資源的使用範圍(前提是如果資源可使用的話)。也就是說，當有資源被耗用時，它們就會移至指定的「資源集區」內，而且無法被共用。但當指定的「資源集區」中沒有任何要求時，則設定給此「資源集區」的資源，將可以釋放給其他「資源集區」來使用，採取這樣做法將可以改善資源的使用率。

若要計算出各個「資源集區」實際有效的 MAX 值，可以參考以下的公式：

- $\text{Min}(X, Y)$ 代表較小的 X 和 Y 值。
- $\text{Sum}(X)$ 代表所有集區之 X 值的總和。
- 全部共用的 % = $100 - \text{sum}(\text{MIN} \%)$ 。
- 有效的 MAX % = $\text{min}(X, Y)$ 。

- 共用的% = 有效的 MAX % - MIN %。

在以下的範例中，建立兩個使用者自訂的「資源集區」，並且分別設定了 MIN 值為 20%與 70%，MAX 值為 100%與 70%，請參考以下表 1 的說明：

集區名稱	MIN 值	MAX 值	計算出有效的 MAX 值	計算出共用的 %	說明
default	0	100	30	30	計算出有效的 MAX 值： $\min(100, 100 - (20 + 50)) = 30$ 。 計算出共用的%：計算出有效的 MAX 值 - MIN = $30 - 0 = 30$ 。
集區 1	20	100	50	30	計算出有效的 MAX 值： $\min(100, 100 - 50) = 50$ 。 計算出共用的%：計算出有效的 MAX 值 - MIN = $50 - 20 = 30$ 。
集區 2	50	70	70	20	計算出有效的 MAX 值： $\min(70, 100 - 20) = 70$ 。 計算出共用的%：計算出有效的 MAX 值 - MIN = $70 - 50 = 20$ 。

表 1：計算有效的 MAX 值與共用的%

延續表 1，若在增建一個「資源集區」：集區 3，設定其 MIN 值為 5%，MAX 值為 100%，則其他「資源集區」有效 MAX 值與共用%部分，系統將自動進行調整，請參考以下表 2 的說明：

集區名稱	MIN 值	MAX 值	計算出有效的 MAX 值	計算出共用的 %	說明
default	0	100	25	25	計算出有效的 MAX 值： $\min(100, 100 - (20 + 50 + 5)) = 25$ 。 計算出共用的%：計算出有效的 MAX 值 - MIN = $25 - 0 = 25$ 。
集區 1	20	100	45	25	計算出有效的 MAX 值： $\min(100, 100 - 55) = 45$ 。 計算出共用的%：計算出有效的 MAX 值 - MIN = $45 - 20 = 25$ 。

集區 2	50	70	70	20	計算出有效的 MAX 值： $\min(70, 100 - (20 + 5)) = 70$ 。 計算出共用的%：計算出有效的 MAX 值 - MIN = $70 - 50 = 20$ 。
集區 3	5	100	30	25	計算出有效的 MAX 值： $\min(100, 100 - (20 + 50)) = 30$ 。 計算出共用的%：計算出有效的 MAX 值 - MIN = $30 - 5 = 25$ 。

表 2：計算有效的 MAX 值與共用的%

要提醒您的是：

在表 1 與表 2 中，並未包含「internal 資源集區」，這是因為「internal 資源集區」其資源的耗用量是沒有限制的，也無法設定其 MIN 值與 MAX 值。

以下列舉數種特殊的「資源集區」之情境：

- 每個「資源集區」皆明確指定 MIN 值，而且其加總值為 100%時，在此種情況下，有效的 MAX 值就等於 MIN 值。這就相當於將伺服器的資源，劃分成許多獨立非重疊的部分一樣。
- 若將每個「資源集區」的 MIN 值設定為 0%，而 MAX 值都設定為 100%時，則每個「資源集區」將會爭用全部可用的資源。

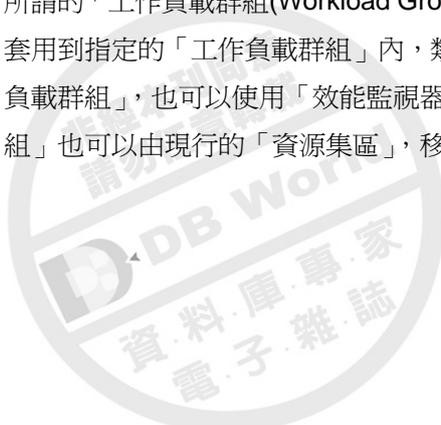
「資源管理員」與 CPU

對於 CPU 資源的管理部分，「資源管理員」所採取的運作方式，是對此「執行個體」所能夠使用的全體 CPU 資源進行調配，將 CPU 資源平均分散給每個 CPU「排程器(scheduler)」來處理。請注意，這裡是對各個 CPU 排程器，採取平均分散的方式來配置。

所以在「工作階段」上所執行的程式，是否能以平行處理方式，執行多個執行緒，將會影響到「資源管理員」的分配方式，例如：在多核心 CPU 的環境上，設定兩個「資源集區」的 MAX 值分別為 40%、50%，但是在「工作階段」上所執行的程式，卻僅能使用到個別單一 CPU 排程器方式來執行。在這樣的情況下，即便此程式是屬於耗用大量 CPU 資源的程式，但卻因為其他 CPU 排程器都沒有受到資源負載的影響，所以「工作階段」上的程式仍然可以超所設定的過 MAX 值方式來執行。如果說能夠以平行處理方式來執行多個執行緒，或是在其他「資源集區」有更多的執行請求時，則可以明顯看到「資源管理員」所發揮的管控功能。

認識「工作負載群組」

所謂的「工作負載群組(Workload Group)」，可用於依據「分類函數」所設計的分類準則，將各個「工作階段」套用到指定的「工作負載群組」內，類似於「工作階段」要求的容器。在「資源集區」下，可建立多個「工作負載群組」，也可以使用「效能監視器」來觀察「工作負載群組」的資源耗用量。使用者定義的「工作負載群組」也可以由現行的「資源集區」，移動到另一個「資源集區」下。請參考下圖 8 所示：



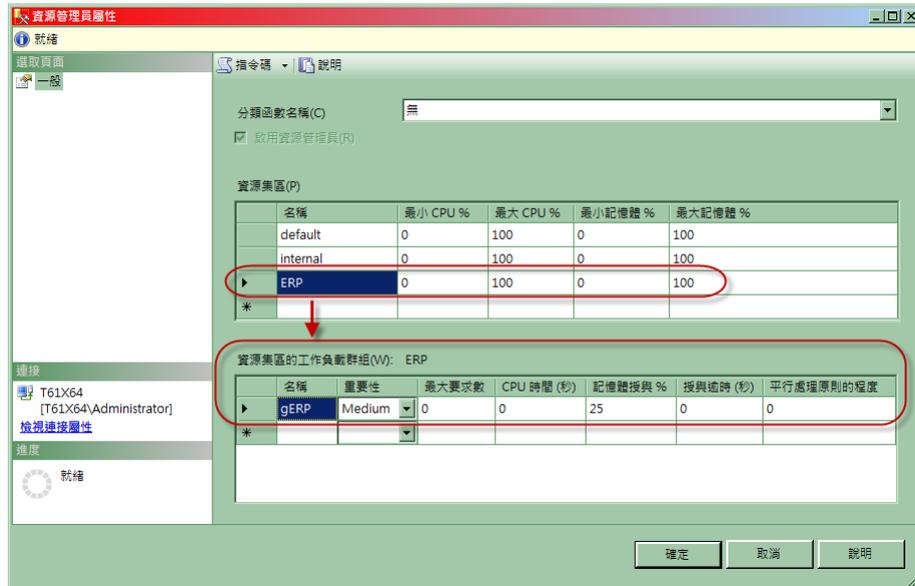


圖 8：檢視「工作負載群組」的設定介面

依據預設值，在「資源管理員」內，預先定義了兩個「工作負載群組」：「default」與「internal」。雖然管理人員對於「internal 工作負載群組」無法變更或是調整，但是仍可進行監視其資源耗用量。當符合以下條件，「工作階段」要求將被分類的「default 工作負載群組」下：

- 沒有「分類函數」，或是未符合分類函數之準則的「工作階段」。
- 嘗試分類在不存在的「工作負載群組」。
- 發生一般分類失敗。

重要性(importance)

指定要求在「工作負載群組」中的相對重要性，預設值為 MEDIUM。可設定為 LOW、MEDIUM 與 HIGH。相對重要性的範圍為本機，且適用於在同一個「資源集區」中的「工作負載群組」。僅適用於查詢的 CPU 分配量。

最大要求數目(GROUP_MAX_REQUESTS)

指定在「工作負載群組」執行時，其所允許的最大的同時要求數。輸入值須為正整數。輸入值的預設值為 0，表示不限制其要求數量。

CPU 時間(秒)(REQUEST_MAX_CPU_TIME_SEC)

指定要求可以使用的最大 CPU 時間量 (以秒為單位)。輸入值須為正整數。輸入值的預設值為 0，這代表沒有限制。「資源管理員」不會在超過最大 CPU 時間時，阻止要求繼續執行。但系統會產生相對事件，在 SQL Server Profiler 或 SQL 追蹤，在「Errors and Warnings 事件類別目錄」下的「CPU Threshold Exceeded 事件類別」，可用來指出「資源管理員」偵測到超出指定之 CPU 臨界值的查詢。

順帶一提，在 SSMS 管理介面上，看到的「CPU 時間(秒)」，但在 CREATE WORKLOAD GROUP 陳述式，使用的參數是：REQUEST_MAX_CPU_TIME_SEC，而且在 SQL Server 2008 線上叢書 (2009 年 1 月) 的文件裡，多半也使用「最大 CPU 時間」這個詞彙。

記憶體授與%(REQUEST_MAX_MEMORY_GRANT_PERCENT)

指定單一要求可由「資源集區」中獲取的記憶體最大數量。這個百分比相對於在「資源集區」內所指定的「最大記憶體%(MAX_MEMORY_PERCENT)」之大小。輸入值須為正整數。允許的輸入值範圍從 0 至 100。輸入值的預設為 25。若將輸入值設定為 0，將會造成在此「工作負載群組」內的「工作階段」，避免執行 SORT

和 HASH JOIN 作業的查詢。

依據 SQL Server 2008 線上叢書 (2009 年 1 月)的說明，建議不要將輸入值設定為大於 70 的值，因為如果有其他的並行查詢正在執行，則伺服器可能沒有足夠的記憶體來因應並行查詢的處理，最後，這可能會導致查詢逾時，產生錯誤 8645：「等候記憶體資源來執行資源集區 xxx 中的查詢時發生逾時。請重新執行查詢」。

如果查詢記憶體需求超過這個參數所指定的限制，伺服器會執行以下作業：

- 如果是使用者定義的工作負載群組，伺服器會嘗試減少查詢的平行處理原則程度，直到記憶體需求低於此限制，或是調整平行處理原則程度等於 1 來處理。如果查詢記憶體需求仍然大於此限制，將會發生錯誤 8657：「無法取得 xxx KB 的記憶體授權，因為它超過工作群組 xxx 和資源集區 xxx 中的最大組態限制。請連絡伺服器管理員，以提高記憶體使用限制」。
- 如果是內部和預設的工作負載群組，伺服器會允許查詢取得所需的記憶體。
- 請注意，如果伺服器沒有足夠的實體記憶體，上述這兩種情況都會導致查詢逾時，產生錯誤 8645。

授與逾時(秒)(REQUEST_MEMORY_GRANT_TIMEOUT_SEC)

指定查詢能夠等候記憶體授權 (指工作緩衝區記憶體(work buffer memory))，變成可用的記憶體之最大等待時間 (以秒為單位)。若是到達記憶體授權的逾時值時，查詢不一定會失敗。通常是在當有太多並行的查詢正在執行時，查詢才會失敗。否則，查詢可能只會得到最小的記憶體授權，導致此查詢效能降低。

平行處理原則的程度(MAX_DOP)

對平行查詢要求，指定其平行處理原則的最大程度(degree of parallelism, DOP)。允許的輸入值之範圍從 0 至 64，輸入的預設值是 0，將使用全域設定，也就是在「執行個體」上「max degree of parallelism(平行處理原則的最大程度)」的設定值。

有關於平行處理原則的最大程度(MAX_DOP)的設定部分，採取以下的方式進行處理：

- 只要 MAX_DOP 沒有超過「工作負載群組」的 MAX_DOP，就會接受使用「查詢提示」的 MAX_DOP。
- 「查詢提示」的 MAX_DOP 永遠會覆寫 sp_configure 的「max degree of parallelism」設定值。
- 「工作負載群組」的 MAX_DOP 會覆寫 sp_configure 的「max degree of parallelism」。
- 如果查詢在編譯時間被標示為序列(serial, MAX_DOP = 1)，則不管「工作負載群組」或 sp_configure 設定為何，都無法在執行階段將該查詢變更回平行處理。
- 設定平行處理原則的最大程度(MAX_DOP)後，在授與記憶體不足的壓力下，僅能其降低平行度。

認識「分類函數」

當進行連線登入時，「資源管理員」將利用「分類函數」，將各個「工作階段」分類到指定的「工作負載群組」內。所謂的「分類函數(Classifier Function)」，是指使用者定義的純量函數(scalar function)，內含將內送「工作階段」指派給「工作負載群組」的程式邏輯。以下是「分類函數」的特性與注意事項：

- 此使用者定義函數，應建置在系統資料庫 master 內。
- 此使用者定義函數，應設計為「結構描述繫結函數」，利用 WITH SCHEMABINDING 子句，可將函數與它參考的任何物件之結構描述繫結在一起，例如資料表、檢視及其他使用者自訂函數。嘗試更改或卸除任何被結構描述繫結函數所參考的物件將會失敗。
- 在「資源管理員」內，每次僅能使用一個使用者定義函數作為「分類函數」。
- 系統會針對每個新的「工作階段」，使用此「分類函數」進行評估，即便有啓用了連接共用(connection pooling)機制也一樣。
- 無法「分類函數」所歸類的「工作階段」，或是傳回 NULL 或不存在群組的名稱時，這些「工作階段」

都將被分配到「default 工作負載群組」。若此「分類函數」因故分類失敗，這些「工作階段」也將被分配到「default 工作負載群組」。

要提醒您的是：

建議啟用「專用管理員連接(DAC)」。因為「專用管理員連接(DAC)」不會受到「資源管理員」的管理限制，這就可以用來監視和疑難排解此「分類函數」。依據預設值，僅能在此伺服器上執行「專用管理員連接(DAC)」，不允許以遠端網路連線方式來執行「專用管理員連接(DAC)」。特別在「容錯移轉叢集(Cluster)」環境上，請記得啟用「專用管理員連接(DAC)」。管理人員可以使用「介面區組態」或是 `sp_configure` 預存程序，設定啟用「remote admin connections」選項。如果發生連「專用管理員連接(DAC)」也無法處理的疑難排解，請改用單一使用者模式來啟動此「執行個體」，再停用此「分類函數」。

在「分類函數」內使用系統函數

在「分類函數」中可以搭配使用以下的系統函數，用來識別各個「工作階段」之特性。請參考下**表 3** 所示：

系統函數	說明
HOST_NAME()	傳回工作站名稱。
APP_NAME()	傳回目前工作階段的應用程式名稱。
SUSER_NAME()	傳回使用者的登入識別名稱。
SUSER_SNAME()	傳回與安全性識別碼 (SID) 相關聯的登入名稱。
IS_SRVROLEMEMBER()	指出目前 SQL Server 登入是不是指定固定伺服器角色的成員。
IS_MEMBER()	指出目前的使用者是否是資料庫角色(包括資料庫固定角色或使用者自訂角色，但不包括伺服器角色)或 Windows 網域群組的成員。
LOGINPROPERTY()	可以使用 DefaultDatabase 與 DefaultLanguage，傳回此登入帳戶的登入預設資料庫與登入預設語言。
ORIGINAL_DB_NAME()	傳回使用者在資料庫連接字串中指定的資料庫名稱。 例如：使用 <code>sqlcmd -d</code> 選項(發出 USE database)或 ODBC 資料來源運算式(設定初始目錄 =databaseName) 所指定的資料庫。
CONNECTIONPROPERTY()	傳回連接屬性的相關資訊，例如：實體傳輸通訊協定、此登入帳戶的驗證配置(例如：SQL、NTLM、KERBEROS 等)、目標伺服器的 IP 位址、連接的目標伺服器 TCP 埠、用戶端的位址等。

表 3：搭配使用的系統函數

要提醒您的是：

使用 HOST_NAME()、APP_NAME() 系統函數是可以回傳相關資料，但是用戶端應用程式也可以設定提供工作站名稱、應用程式名稱，而且可設定不正確的資料。所以請勿依賴 HOST_NAME()、APP_NAME() 當做安全性的功能。

以下為建置分類函數的流程：

- 建立使用者定義的純量函數。
- 向「資源管理員」註冊使用此函數為「分類函數」。
- 更新「資源管理員」在記憶體中組態設定。
 - 可以使用 SSMS 管理工具，或是執行 ALTER RESOURCE GOVERNOR RECONFIGURE 陳述式。
- 使用「專用管理員連接 (DAC)」進行疑難排解。

執行分類的登入程序

啓用「資源管理員」後，每個「工作階段」的登入程序包含以下程序：

- 登入驗證。
- 執行 LOGON 觸發程序。
- 執行分類。

開始分類時，「資源管理員」就會執行「分類函數」並使用此函數所傳回的值，將要求傳送至適當的「工作負載群組」。

「資源管理員」的組態設定

「資源管理員」提供兩種狀態設定：啓用或停用。依據預設值，「資源管理員」的狀態是：停用狀態。對「資源管理員」所執行的組態變更，是不會立即生效。若要變更「資源管理員」的組態設定，請依循以下的流程來進行：

- 發出適當的 ALTER 陳述式或使用 SSMS 管理工具的設定變更，進行調整「資源集區」或「工作負載群組」的組態。這部分是將組態設定的「中繼資料(存放系統資料表內)」進行更新。
- 發出 ALTER RESOURCE GOVERNOR 陳述式或使用 SSMS 管理工具執行「重新設定」選項。這會將「中繼資料」複製成爲其在記憶體中的組態值。

也就是說，變更「資源管理員」的組態設定，需要等到將「中繼資料」複製到記憶體上才能發揮作用。大部分所執行的組態變更設定，都可以即時發揮功用，但主要都是影響到新建立的「工作階段」，對於現有的「工作階段」可能不受到影響，請參考以下表 4 的整理：

元件	說明
分類函數	變更分類函數，將影響到新建立的「工作階段」。但不影響現有的「工作階段」。
資源集區	若爲長時間執行的「工作階段」可能會受到此項變更的影響。新建立的「工作階段」將受到影響。

工作負載群組

現有的「工作階段」不會受到影響，但是新建立的「工作階段」將受到影響。

表 4：調整組態設定對「工作階段」的影響

若原本有啟用「資源管理員」，後來停用「資源管理員」，將產生以下的影響：

- 系統一併停用「分類函數」、「資源集區」與「工作負載群組」等的設定。
- 現有的「工作階段」將不再受到「資源管理員」的管理。

監控「資源管理員」

在使用「資源管理員」時，可以使用以下的方式來觀察其運作：

- 效能計數器。
- SQL 追蹤事件。
- 動態管理檢視與目錄檢視。

取得各個「工作負載群組」和「資源集區」的執行統計資料，是在使用「資源管理員」的管理策略上很重要的部份。收集下來的執行統計資料，可以讓資料庫管理師識別對「資源管理員」的組態設定，是否遵循預期般的規劃，並且及早偵測出可能的問題，以便採取更正動作，並確保可靠且可預測的作業能持續進行。

效能計數器

管理人員可以使用以下的效能物件來收集「工作負載群組」和「資源集區」的統計資料，請參考以下表 4 的說明：

效能計數器名稱	說明
SQLServer:Workload Group Stats	報告每個使用中的「工作負載群組」之統計資料，例如：在此「工作負載群組」中執行的要求數目、要求的 CPU 頻寬使用量、被封鎖的要求數目等。
SQLServer:Resource Pool Stats	報告每個使用中的「資源集區」之統計資料，例如：要求的 CPU 頻寬使用量、可擁有的最大記憶體數量、每秒逾時的記憶體授權數目等。

表 5：可使用的效能計數器

管理人員也可以利用動態管理檢視 `sys.dm_os_performance_counters` 來查詢計數器的現行值。

SQL 追蹤事件

以下針對「資源管理員」可使用的 SQL 追蹤事件，請參考以下表 6 的說明：

事件名稱	說明
CPU Threshold Exceeded	指出「資源管理員」偵測到某個查詢已經超過 <code>REQUEST_MAX_CPU_TIME_SEC</code> 中的 CPU 臨界值之設定。 此事件的偵測間隔為五秒。也就是說，如果某個查詢超過指定的 CPU 臨界值限制達五秒以上，就會產生此事件。不過，如果某個查詢超過指定的臨界值少於

	五秒，根據查詢的時間和上一次偵測清除的時間，可能會導致遺漏偵測。
PreConnect:Starting	指出 LOGON 觸發程序或「資源管理員」的「分類函數」已經開始執行。
PreConnect:Completed	指出 LOGON 觸發程序或「資源管理員」的「分類函數」已經完成執行。

表 6：可使用的 SQL 追蹤事件

動態管理檢視與目錄檢視

以下列舉數個常用的「動態管理檢視」與「目錄檢視」，可用於觀察「工作負載群組」和「資源集區」的統計資料和組態資料。兩者最大的差異在於，「目錄檢視」是用來查詢其儲存在中繼資料(資料表)內的組態設定，若要檢視現行在記憶體上的組態設定與統計資料，請使用「動態管理檢視」，請參考以下表 7 與表 8 的說明：

名稱	說明
sys.resource_governor_configuration	傳回儲存在中繼資料內的「資源管理員」之組態。
sys.resource_governor_resource_pools	傳回儲存在中繼資料內的「資源集區」之組態。
sys.resource_governor_workload_groups	傳回儲存在中繼資料內的「工作負載群組」之組態。

表 7：與「資源管理員」有關的「目錄檢視」

名稱	說明
sys.dm_resource_governor_workload_groups	傳回「工作負載群組」的統計資料以及在目前在記憶體中組態。
sys.dm_resource_governor_resource_pools	傳回「資源集區」目前的狀態之相關資訊、目前的組態與其統計資料。
sys.dm_resource_governor_configuration	傳回「資源管理員」目前在記憶體中組態狀態。

表 8：與「資源管理員」有關的「動態管理檢視」

「資源管理員」的限制條件與注意事項

在 SQL Server 2008 與 SP1 版本的「資源管理員」部分，有以下的限制條件：

- 「資源管理員」僅是用於 SQL Server Database Engine，無法應用於 Analysis Services、Integration Services 和 Reporting Services。
- 「資源管理員」是以 SQL Server「執行個體」為管理單位。個別的「執行個體」需要分別進行設定「資

源管理員」。

- 「資源管理員」可以配置的硬體資源是：CPU 與記憶體，尚無法控制磁碟 I/O 資源。這點十分可惜，在部分案例裡，磁碟 I/O 仍是資料庫系統的瓶頸，例如：查詢程式沒有使用到索引，造成「資料表掃描(TABLE SCAN)」等，導致其他程式沒有足夠的磁碟 I/O 資源可用。
- OLTP 工作負載。「資源管理員」可以管理 OLTP 工作負載，但是這些類型的查詢（通常持續時間會很短）在 CPU 上的時間不一定都夠長而足以套用頻寬控制。這樣可能會扭曲針對 CPU 使用量百分比傳回的統計資料。
- 不適合建置過多的「資源集區」。請善用在「資源集區」建立「資源群組」做分類。在目前版本上，至多可以自行建立 18 個「資源集區」，若將內建的「default」與「internal」兩個「系統資源集區」一併列計算，至多是可以使用 20 個「資源集區」來進行設計與規劃資源的分配。但是「資源群組」部分，則無此限制。請參考下圖 9 所示：



訊息 10909，層級 16，狀態 1，行 1
無法建立資源集區。資源集區數上限不能超過目前的限制 20，其中包括預先定義的資源集區。

圖 9：建立超過容量限制的「資源集區」，所遇到的錯誤訊息

安全性的注意事項

有關於安全性的注意事項，請參考如下：

- 若要變更「資源管理員」組態設定，使用者必須擁有 CONTROL SERVER 權限。
- 若要使用動態管理檢視來觀察「資源管理員」正在使用中的組態設定，使用者必須擁有 VIEW SERVER STATE 權限。

「資源集區」和「工作負載群組」的名稱之安全性

所有「資源集區」和「工作負載群組」的名稱都是公開的。因此，建立「資源集區」和「工作負載群組」時，建議選擇不會透露在伺服器上執行之應用程式本質等相關資訊的名稱。

結語

在本期中，介紹了「資源管理員(Resource Governor)」的整體架構與相關注意事項，在下一次的文章，將帶領各位按部就班的來建置「資源管理員(Resource Governor)」。

DB World

◀ 本文作者簡介 ▶▶▶

陳俊宇
Derrick Chen

現任職恆逸資訊
教育訓練處資深講師
擁有 MVP、MCT、MCTS、MCITP、MCPD 等認證，並為專欄作家，最新著作：SQL Server 2008 管理實戰 - 營運管理篇。作者部落格為：
德 瑞 克 (Derrick)
<http://sharederrick.blogspot.com/>