

# GPU化MOS統計降尺度建模系統設計與實作

馮智勇<sup>1</sup> 陳昱璵<sup>1</sup> 賈愛玫<sup>2</sup> 許乃寧<sup>2</sup>  
多采科技有限公司<sup>1</sup> 中央氣象局預報中心<sup>2</sup>

## 摘要

模式輸出統計法(Model Output Statistics, MOS)為氣象與氣候領域降尺度應用常見方法，透過逐格點建置真實場/分析場與模式輸出層場/重預報資料的複回歸模型，可移除模式系統性偏差而為預報使用。建模過程主要包含讀取或計算模式格點的基本層場與導出量場、內插到真實場/分析場格點與挑選具解釋能力預報因子等三個步驟。然而，考量模式解析度不斷提高且預報延時日益延伸，建置MOS降尺度計算量與成本必然持續增加。

本研究藉由分析建模過程所需資料與前進選擇法計算步驟，設計適合採用平行計算技術的建模系統，並應用NVIDIA CUDA架構建寫CUDA Fortran程式實作GPU化MOS建模系統，而可藉由一般筆記型電腦配置的GPU硬體(GeForce GTX 850M)於4.3秒內建置7,280個格點複線性回歸方程式，相較於使用CPU單執行緒(Intel® Core™ I5-4210M)所需約55秒的計算時間，可獲得約為12.7倍加速比；若以桌上型主機配置的Geforce GTX 780 Ti硬體，平均可於12.3秒內完成預報中心圖型預報編輯器(GFE)範圍內的67,600個格點模型建置，相較於使用Intel® Core™ I7-4820K單執行緒平均計算時間261.7秒，則有21倍的加速比。

實作GPU Computing平行技術除可縮短統計回歸模型發展與測試的時程外，也有助於開發即時搜尋、類比歷史觀測/模式重預報資料相似案例的智慧型專家系統以利提高預報技術。

關鍵字：GPU Computing，CUDA Fortran，MOS

## 一、前言

隨著氣象與氣候數值模式的發展興起無縫隙預報概念(Seamless prediction of weather and climate)，氣象模式開始產製14天甚至16天的預報資訊，而氣候模式亦致力提升第2~4週短期氣候尺度預報能力。然而，數值動力模式受限於僅為真實環境的近似模擬，必然因物理行為參數化過程、時間及空間解析度不足以代表真實情況而造成模式直接輸出結果(Direct Model Output)存在系統性偏差，需要予以移除後方能作為預報使用[7]。

系統性偏差校正方法常見如傳統的模式輸出統計法(Model Output Statistics, 以下簡稱MOS)[5]、類比法(Analog Techniques)[8]、系集回歸法(Ensemble Regression)[9]與誤差衰退平均法(Decaying Average, 以下簡稱DCA)[4,6]。其中，DCA是NCEP基於簡易卡門濾波發展作業化即時校正GEFS模式的系統性偏差移除方法，特色在於透過時間序列移動平均的方式捕捉近期預報與觀測資料的偏差值並加以移除，無需儲存模式近期模擬結果或使用重預報資料(Reforecasts)而有別於其他方法。然而，DCA改善效果隨著預報延時而下降，約莫在第7天即無顯著改善，因此第二週預報在春天與秋天等季節轉換時容易有較大的誤差，仍需借助重預報資料維持預報水準至16天[6]。因此，欲藉由數值動力模式輸出提高5~7天或者是第2~4週預報技術水準，必然需要倚賴

模式的重預報資料。是故如何針對大量資料讀取與網格點統計後處理需求，設計並建構高效率模型建置與分析系統將成為動力模式資料分析者所必須面臨的課題。

此外，隨著預報的時間與空間解析度不斷提高，以網格點為依據的統計模型數量也隨之增加，若可藉由實作平行運算技術縮短分析時間勢必有助於加快研發進程。GPU Computing自2008年發展至今已相當成熟，除了可於一般筆記型電腦或者個人電腦使用外，Top 500超級電腦排名前十大者常見配備GPU加速裝置，而2017年6月份的Green 500前十大排名更是有9個配置NVIDIA Tesla P100的GPU硬體。未來將是在綠能、節能減碳與降低維運成本的考量上必須使用的硬體設備。

有鑑於此，本研究以傳統的MOS統計後處理方法為例，依序分析複線性回歸模型建置流程與大量資料使用方法，並提出以GPU為計算資源的高效率GPU化MOS建模系統設計與實作成果。

## 二、MOS模型建置系統設計

### 2.1 複線性回歸方程式建置

[1~3,5]令 $\hat{y}$ 為格點觀測/分析場資料的估計值(可視為預報值或者是模式校正值)， $\beta_i$ 為待求回歸係數， $N_x$ 為預報因子個數，則複線性回歸方程式如下所示。

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_{N_x} x_{N_x} \quad (1)$$

假設共有  $N_{rec}$  筆 XY 配對資料 ( $N_{rec} > N_x$ ) 可供回歸建模，則可利用矩陣將資料表示為  $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta}$ ，其中， $\mathbf{X}$  為由模式資料與常數 1 組成的矩陣，表示為：

$$\mathbf{X}_{N_{rec} \times (N_x+1)} = \begin{bmatrix} 1 & X_{1,1} & X_{1,2} & \dots & X_{1,N_x} \\ 1 & X_{2,1} & X_{2,2} & & X_{2,N_x} \\ 1 & X_{3,1} & X_{3,2} & \ddots & X_{3,N_x} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{N_{rec},1} & X_{N_{rec},2} & \dots & X_{N_{rec},N_x} \end{bmatrix} \quad (2)$$

而  $\boldsymbol{\beta}$  與  $\mathbf{Y}$  分別為回歸係數行向量 (column vector) 與觀測/分析場資料行向量，可表示為：

$$\boldsymbol{\beta}_{(N_x+1) \times 1}^T = [\beta_0, \beta_1, \beta_2, \dots, \beta_{N_x}]$$

$$\mathbf{Y}_{N_{rec} \times 1}^T = [Y_1, Y_2, Y_3, \dots, Y_{N_{rec}}]$$

利用最小誤差平方和法 (Least Square Method) 即可求得係數矩陣估計值為：

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

為了計算  $\mathbf{X}^T \mathbf{X}$  的反矩陣值，現行採用 LU 矩陣分解法得到上三角與下三角矩陣，再透過矩陣相乘運算求解。回歸模型中的預報因子個數  $N_x$  是由使用者預設一個最大數量後，透過前進選擇法 (Forward Selection) 比較不同等壓層的模式輸出基本場 (如溫度) 與導出量 (如溫度平流) 的解釋能力與共線性，逐一挑選解釋能力最佳且與模型現有預報因子共線性最小者為新增的預報因子，直到個數達到  $N_x$  或者是模型解釋能力無顯著增加 (F-test) 為止。因此， $\mathbf{X}^T \mathbf{X}$  的維度會逐步由  $2 \times 2$ 、 $3 \times 3$  增加至最大維度  $(N_x+1) \times (N_x+1)$  為止。

## 2.2 高效率 MOS 模型建置系統設計

MOS 模型建置流程如圖 1 所示，包含讀取已完成間內插至格點處的模式氣象層場資料、讀取格點觀測/分析場資料、複線性回歸模型建置與統計值計算等步驟，最後則輸出回歸模型係數。

上述過程除了模型建置與統計值計算外，大部分均與讀取事前準備好的模式資料有關，包含利用模式基本場計算導出量場以及內插基本場與導出量場至格點處的資料，勢必增加許多需要儲存的中繼檔案而耗費大量儲存空間。另一方面，倘若每次讀取檔案時只取一個格點的資料，則建置  $N$  個格點回歸模型需要開啟同一個檔案  $N$  次，勢必因頻繁的 IO 動作而嚴重影響建模效率。

鑒於上述分析，本研究認為高效率 MOS 模型建置系統的設計原則在於「減少檔案讀取次數」與「實作平行運算技術」，因此發展建模系統運作流程如圖 2 所示，包含必要 (以實線圓角框表示) 的讀取模式輸出基本場資料 (Read XX)、讀取格點觀測/分析場資料 (Read YY) 與合理值域檢查 (YYQC) 與模型建置 (Regression Process) 等步驟，亦有依照使用者設定啟動與否的模式導出量計算與內插 (以虛線圓角框表示)

程序 [1~3]。如此即可在建模前僅需準備好模式輸出的基本場資料檔案，完全無需儲存導出量場檔案與內插格點檔案，而建模時則是透過平行計算技術即時利用基本場計算必要的導出量場與進行內插，達成藉由平行計算技術減緩讀取大量資料的 IO 瓶頸，亦可同時建置  $N$  個複線性回歸方程式，進一步縮短計算時間。

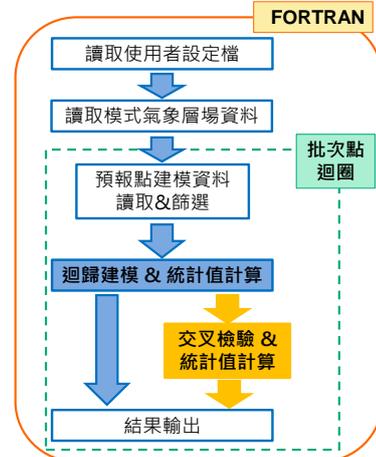


圖 1 MOS 模型建置系統運作流程

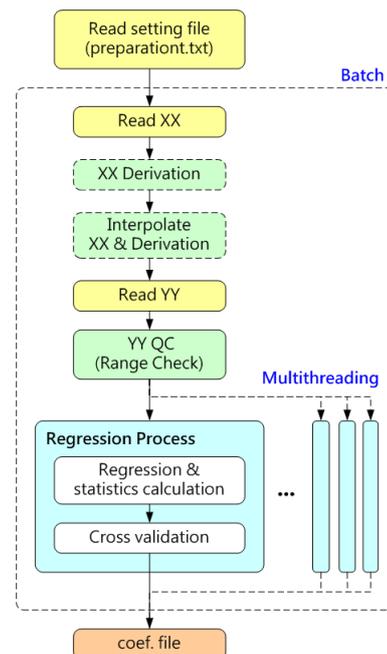


圖 2 高效率 MOS 模型建置系統運作流程

## 三、GPU 化 MOS 建模系統實作

### 3.1 CUDA 技術簡介

CUDA (Compute Unified Device Architecture) 為 NVIDIA [10] 針對旗下圖形處理器所提出「統一計算架構」技術，設計執行緒 (thread)、執行緒區塊 (thread block，簡稱 block) 與執行緒區塊群組 (grid) 三層架構 (如圖 3)，提供使用者以 C、Fortran、Java、Python、.Net 以及 OpenCL 等程式語言管理執行期之執行緒，對應

利用GPU計算核心(thread processor, 或稱CUDA core)、串流複合處理器(streaming multiprocessor)與晶片(device)三層硬體架構進行高速平行計算。

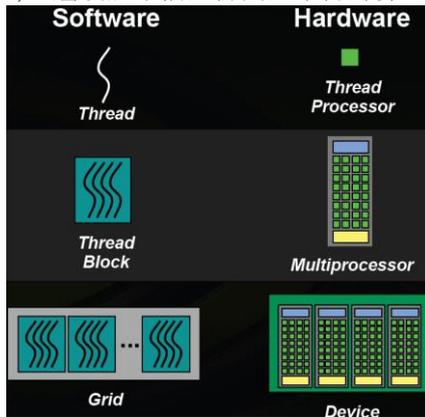


圖3 CUDA技術執行緒管理方式與GPU硬體對應關係

GPU應用程式於作業系統的運作係採GPU協同CPU作業方式, 包含CPU複製系統主記憶體資料到GPU、啟動GPU上之CUDA計算核心、CUDA計算核心平行執行計算程序(kernel), 以及CPU將計算結果複製回系統主記憶體等四個步驟。其中, 在平行執行計算程序步驟中, 每個CUDA計算核心係由GPU上的單一執行緒(thread)所控制, 透過該執行緒唯一編號取得相關資料後, 執行相同之計算程序[10]。

NVIDIA圖形處理器設計包含全域記憶體(global memory)、常數記憶體(constant memory)、材質記憶體(texture memory)、共享記憶體(shared memory)、區域記憶體(local memory)以及暫存器(register)等6種不同類型的記憶體。全域、常數、材質與區域記憶體皆位於DRAM上(稱off-chip位置), 可供不同block中的各執行緒使用; 共享記憶體與暫存器皆位於串流複合處理器上(稱on-chip位置), 共享記憶體執行時期提供位於一個block內的所有執行緒讀寫, 執行緒存取資料速度約為存取全域記憶體的100倍; 暫存器如同於CPU暫存器, 執行緒存取資料速度約與共享記憶體相當, 用以儲存執行緒執行計算程序中區域變數[10]。

### 3.2 MOS建模系統CUDA技術設計實作

藉由NVIDIA GPU硬體設計可知實作CUDA技術建置高速平行計算應用程式的設計原則有二, 一為如何使用GPU硬體提供的眾多執行緒同時進行獨立的計算工作, 二為如何利用存取速度達100倍的暫存器或共享記憶體加速資料傳遞, 必須依應用程式計算瓶頸的差異進行設計。

以圖2的MOS模型建置系統運作流程而言, 可利用GPU進行加速的瓶頸明顯落於複線性回歸方程式建置步驟。由2.1節所述的前進選擇法可知, 每次將選擇具有最佳解釋能力且與現有預報因子共線性最小的層場以擴充方程式預報因子個數, 因此需要逐一比較尚未被挑選的層場。假設現有預報方程式中

已存在3個預報因子, 而層場數量總共為60個, 則挑選第4個預報因子時需要逐一比較其餘27個層場的解釋能力以及共線性程度。為此, 一種實作GPU技術的可能設計是以一個block建置一個格點複線性回歸模型的架構, 將已選入的前3個預報因子資料置於block上的共享記憶體, 則可啟動該block上的27個執行緒同時計算3+1個預報因子方程式, 同時利用了多GPU執行緒與共享記憶體的優勢, 應可大幅縮短計算時間。

然而, 假設使用3年資料共183筆( $N_{rec}$ )XY配對資料, 透過上述運算方式可知, 式(2)需建構27個 $183 \times 5$ 大小的單精確度浮點數矩陣( $\mathbf{X}$ )以及27個 $5 \times 5$ 大小的雙精確度浮點數矩陣( $\mathbf{X}^T \mathbf{X}$ ), 而共線性計算則另外需要27個 $183 \times 4$ 大小的單精確度浮點數矩陣與27個 $4 \times 4$ 大小的雙精確度浮點數矩陣, 共計需要186,732位元組的記憶體。若可同時進行1,000個格點模型建置, 則SM與GPU端全域記憶體需要雙向處理約182,356 KiB的資料交換, 甚至當挑選第10個預報因子時, 需要雙向處理高達351,504 KiB的資料量, 反而因SM與GPU端全域記憶體頻繁進行資料交換形成效率瓶頸(如圖4(a)), 以致計算效率反而遠低於不實作CUDA技術的MOS建模系統[2,3]。

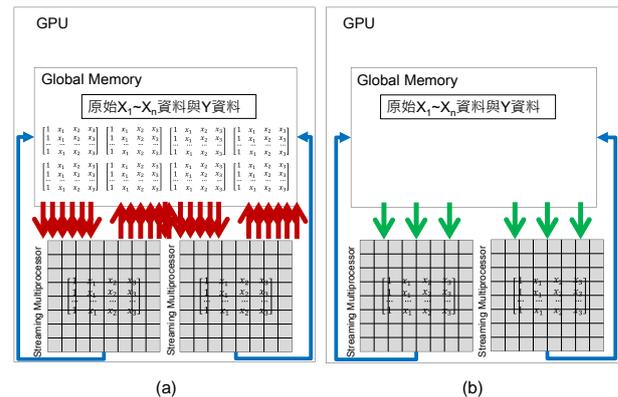


圖4 (a)以執行緒區塊或(b)以執行緒為設計架構的GPU化MOS模型建置系統設計方案比較

為避免受限於記憶體傳輸效率瓶頸, 現階段較適合的設計如圖4(b)所示, 設計位於SM上的每個CUDA core(圖中以灰方格表示)負責一個網格點的建模作業, 自全域記憶體中讀取前進選擇法過程中所需的 $183 \times (30+1) \times 4 = 22,692$ 位元組(22.16 KiB)資料儲存於暫存器, 直到方程式建置完成後才將結果回傳全域記憶體以供下載至CPU端DRAM。此種設計在運算過程中不需要頻繁透過全域記憶體進行資料交換, 而是盡可能將資料儲存於存取速度快的暫存器中, 並且符合大量平行使用執行緒進行模型建置的設計原則[3]。GPU化MOS建模系統程式則是使用PGI CUDA Fortran[11]進行撰寫。

## 四、GPU化MOS建模系統效率測試

以2012~2014年ECMWF數值動力模式28個基本場輸出資料建置GFE範圍2.5公里解析度網格點預報延時000的地面溫度回歸方程式為測試案例，以一張具有5個SM的GeForce GTX 850M裝置為開發環境，令每個SM同時進行32個網格點則可同時進行160個網格點建模作業，可在4.3秒內完成7,280個格點複線性回歸方程式建置作業，相較於使用CPU單執行緒(Intel® Core™ I5-4210M,時脈2.6GHz)所需約55秒的計算時間，可獲得約為12.7倍加速比。

以計算能力與Tesla K40相當的GeForce GTX 780 Ti與採用Intel® Core™ I7-4820K(時脈3.70GHz)進行共67,600個格點模型建置實驗3次如表1所示[3]。在限制記憶體使用量上限為0.5 GiB的情況下，GPU化MOS建模系統將自動分成10個批次進行(第10個批次僅需建置2,080個網格點，因此耗時均低於前9個批次)。綜合3次測試結果可見使用GPU為計算資源的平均耗時12.3秒，I7-4820K平均需要261.7秒，加速比為261.7/12.3=21.28。

表1 GTX 780 Ti與I7-4820K建模耗時比較

批次	測試1		測試2		測試3	
	GPU	CPU	GPU	CPU	GPU	CPU
1	1.225	29.263	1.219	28.820	1.222	29.012
2	1.278	28.426	1.268	28.236	1.286	28.247
3	1.354	28.550	1.335	28.400	1.350	28.433
4	1.322	28.572	1.312	28.479	1.322	28.465
5	1.352	27.750	1.352	27.680	1.357	27.639
6	1.357	27.941	1.345	27.835	1.357	27.796
7	1.297	27.825	1.298	27.708	1.299	27.644
8	1.307	28.195	1.305	28.051	1.305	27.934
9	1.288	28.254	1.291	28.148	1.302	28.148
10	0.531	7.878	0.530	7.869	0.531	7.827
總耗時(秒)	12.31	262.65	12.25	261.22	12.33	261.14

## 五、結論與建議

應用MOS方法建置氣象/氣候模式與觀測值/分析場的複雜性回歸模型時，往往因所需讀取的模式資料量龐大並且格點數眾多，無法快速測試例如模式輸出層場、資料窗大小、內插方式等等建模設定的影響，拖延研發工作進程。本研究提出在建模系統實作時，納入「減少檔案讀取次數」與「實作平行運算技術」的設計原則，有助於開發出高效率MOS模型建置系統。

在減少檔案讀取次數的實作上，可透過讀取最少資料的方式降低IO所需時間，並將如導出量場或內插計算等衍生計算量利用平行運算技術縮短計算時間，因而更進一步提高整體建模系統運算效率。

實作以GPU為計算資源的CUDA架構，GPU化MOS建模系統具有依GPU硬體規格達到加速比達10~20倍的效率提升，相較於實作以CPU為計算資源的OpenMP架構，使用2個CPU執行緒進行平行計算未必可達2倍最佳加速比的情況[1]，GPU Computing確

實具有大幅提升運算效率的能力，值得進一步實作在針對大量網格進行統計後處理的方法。

## 六、參考文獻

- 交通部中央氣象局，2011:「發展鄉鎮逐時天氣預報高解析度網格統計降尺度建置案—統計降尺度天氣預報子系統」委託研究計畫成果報告。
- 交通部中央氣象局，2015:「小區域暨災害性天氣資料分析與應用建置案—統計降尺度天氣預報子系統」委託研究計畫成果報告。
- 交通部中央氣象局，2016:「105年度小區域暨災害性天氣資料分析與應用建置案—統計降尺度天氣預報子系統」委託研究計畫成果報告。
- Cui B., Toth Z., Zhu Y., Hou D., 2012: Bias correction for global ensemble forecast. *Wea. Forecasting*, 27, 396–410, doi:10.1175/WAF-D-11-00011.1.
- Glahn B. and D. A. Lowry, 1972: The use of model output statistics (MOS) in objective weather forecasting. *J. Appl. Meteor.*, 11, 1203–1211
- Guan H., Cui B., Zhu Y., 2015: Improvement of Statistical Postprocessing Using GEFS Reforecast Information. *Wea. Forecasting*, 30, 841–854, doi: 10.1175/WAF-D-14-00126.1
- Hamill T. M., Whitaker J. S., Wei X., 2004: Ensemble reforecasting: Improving medium-range forecast skill using retrospective forecasts. *Monthly Weather Review*, 132, 1434–1447, doi:10.1175/1520-0493(2004)132,1434:ERIMFS.2.0.CO;2
- Hamill T. M., Whitaker J. S., Mullen S. L., 2006: Reforecasts: An important dataset for improving weather predictions. *Bull. Amer. Meteor. Soc.*, 87, 33–46, doi:10.1175/BAMS-87-1-33
- Unger, D. A., H. van den Dool, E. O’Lenic, Collins D., 2009: Ensemble regression. *Mon. Wea. Rev.*, 137, 2365–2379, doi:10.1175/2008MWR2605.1.
- <http://www.nvidia.com/cuda>
- <http://www.pgroup.com>