氣象學報第四十卷第四期(83年12月)

在Cray YMP 8i上增進有限區域預報模式向量化及平行處理之研究

葉天降 簡宏偉
中央氣象局資訊中心

摘 要

中央氣象局發展第二期數值天氣預報計畫，於民國八十一年引進具6個處理器之Cray YMP 8i超級電腦，而於此電腦發展預報模式。本研究嘗試從改善程式之向量化及平行處理度來增進有限區域預報模式之執行效率。

研究過程主要靠flowview、atexpert及atscope來分析程式各部分之執行效率，而後透過相關性分析等做進一步改善。主要之進行方式是針對模式中所需耗費最久之副程式逐一的先設法儘量向量化，而後再改善程式之平行度。主要完成之工作包括詳細檢查整個程式，儘量消除不需的計算，並以暫存空間消除資料相關性或省去重複計算，重新安排迴路運算之計算次序，以系統應用程式取代向量相關之計算，inline處理呼叫次數很多之副程式，進行程式資料相關性分析並加入指引等工作以節省程式計算所需時間。

經過這些改變，使得原來進行48小時預報約需13300秒之程式降至約需9700秒，而原來之程式平均並行使用2.2個中央處理器，經改善後之程式則約可並行使用3.5個中央處理器，如此使得原來大概要6000秒才能執行完畢的程式現在大概只要3000秒。

一、前 言

中央氣象局推展第二代數值天氣預報作業系統時引進了Cray YMP 8i（簡稱Cray8i）含6個中央處理器(central process unit;CPU)之超級電腦，而每一CPU都有向量(vector)處理的能力。

在Cray 8i電腦上中央氣象局建立了全球波譜預報模式、有限區域預報模式、颱風路徑預報模式、最佳內插客觀分析以及各預報模式之非線性正模初始處理等，其中以預報模式因爲要用小時距對許多格點（或波數）做長時間積分，因此所需CPU時間最長。由於中央氣象局是首次使用多CPU的電腦而且在向量化處理方面也並沒有足夠經驗，同時由於大多數的使用者及程式設計人員習慣於使用一般傳統之純量(scalar)、循序處理電腦，其程式執行模式亦受限於是項程式設計模式，所以一旦將原有程式移至另一雖具有快速向量處理與多CPU電腦後，其執行效率並不如預期的好。此時一個完善的前端處理器或編譯程式就扮演極重要的角色，而得將一個單元、循序的程序轉換成適合於在某一特殊機器上相對應的快速程式。然而，再好的編譯程式，也比不上使用者本身在設計時即考慮到平行處理問題來得有效。

本研究係針對中央氣象局第二代數值天氣預報作業所發展眾多系統中擇取計算時間需要很久，而在向量化及平行處理又不儘理想之有限區域預報模式爲對象，在經過對Cray8i電腦系統及向量化、平行處理相關概念之了解後，進行程式之改善工作。第二節將簡述 Cray 8i與中央氣象局第二代有限區域預報模式之特性，改進之過程與結果將在第三節

討論，結論與心得則在第四節。

二、Cray YMP 8i與中央氣象局第二代有限區域預報模式之特性

中央氣象局所引進之Cray YMP 8i/632型超級電腦，係由Cray Research Inc.所生產，最大容量可擁有8個CPU及128百萬字元(mega words;MW)的記憶體，在本研究進行時配備有6個CPU及32MW記憶體。

Cray YMP 機型屬於共用記憶體之多處理器系統，各CPU均可透過一交換網路而共享主記憶體，而每一個CPU則是具有管線處理能力的向量處理器(pipelined vector processor)，同時每一CPU都有很多的功能單位（如加、減）。

以向量處理器而言，它和主記憶體間資料傳輸的方式可分爲二類（詳見Polychronopoulos, 1988; Zima and Chapman 1990; Hwang, 1993），其中一類資料由主記憶體直接進入CPU處理，而後再由CPU將欲寫回的資料，直接寫入主記憶體中。由於主記憶體存取的時延較長，因此該類電腦的管線(pipeline)之長度也較長。如以往中央氣象局所有之Cyber205屬此類。另一類是在主記憶體和CPU間放入暫存器(register file)。所有的讀出和寫入記憶體均須透過此暫存器。因此，一次記憶體存取的長度限制於讀暫存器的長度。但相對的，其管線的長度減少，且可增加管線鏈結的程度。通常其暫存器之長度爲64位元，Cray YMP屬這一類電腦。

除了資料由主記憶體和CPU間的傳輸外，記憶體之分組也影響了整個運算中的速度。一般而言，主記憶體可均分爲數個記憶模組。而不同的記憶模組可同時被存取，以Cray YMP而言，共有128個記憶體模組，也就是說，可以同時有128個資料項目被存取。但是，記憶模組的數目最好和執行迴路運算之間隔互質，以減少記憶存取失敗的頻率。

除了以上的限制外，現今在Cray 8i上所使用的作業系統爲UNICOS，它是UNIX的加強版，但卻使用實際記憶體的管理方法，因此，使用者程式的大小受限於實際記憶體的大小。

中央氣象局第二代區域預報模式是一垂直共20層在蘭伯特(Lambert)投影面計算之疊合式(nested)模式，其中粗網格組之格距爲60公里，含161×121點，而細網格組之格距爲20公里，含91×91點。而在設計上是使用FORTRAN77語言，粗細網格組則儘可能共用副程式。

概略來說，此模式之流程如圖一，主要耗時的

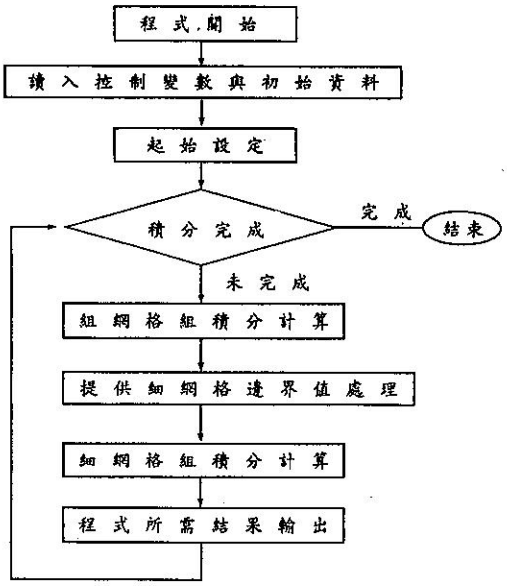

圖 1. 中央氣象局第二代有限區域預報模式運算流程。

是在執行時間的積分部分，而各積分計算則包含物理與動力效應兩大部分，其中物理效應包含積雲參數效應、地表邊界效應、大尺度降水效應、絕對不穩定調整等，詳細之積分情形如圖二。而爲了滿足積分之穩定性，粗網格之積分時距設定爲120秒，細網格之時距爲40秒。對48小時之預報，圖一之迴路共需執行1440次。

對一般使用者以高階語言（如FORTRAN）所撰寫的程式，通常之電腦系統會透過編譯系統將高階語言程式轉換爲可執行檔，Cray 8i也是如此，只不過因爲Cray8i有向量及平行處理能力，爲減少程式發展人員在設計程式時要分心考慮到向量處理及平行處理，因此Cray 8i之編譯系統較一般電腦之編釋系統爲複雜，Cray8i完整之FORTRAN編譯系統分爲前置處理(FORTRAN Preprocessor;FPP)，中期處理(FORTRAN Midprocessor;FMP)，編譯(FORTRAN

Compiler)以及載入(Loader)等四部分。其中，編譯和載入二部分和一般電腦系統之編譯載入功能相近。FPP及FMP則是針對協助使用者應用向量與平行處理而設計，其中 FPP主要在解析並有能力改善使用者之程式，使得程式執行時能提高單元與向量

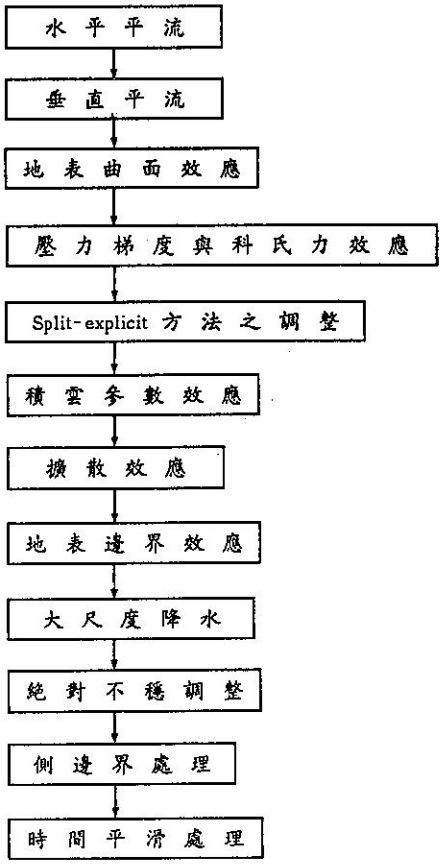

圖 2. 中央氣象局第二代有限區域預報模式之計算次序。

處理之效率。同時，FPP也會分析並決定程式之可並行執行區域，FMP則將程式之可並行執行區域依可使用CPU之數量進一步轉換程式，使程式能由多CPU並行處理。

由於Cray 8i有相當成功的前置及中期處理單元(FPP及FMP)，因此一個結構較簡單的程式即使設計者並未考慮到向量化，平行處理，但經Cray 8i完整的編譯過程，在執行時也往往能相當有效的被向量、平行處理，然而，對一個結構較複雜的程式，則需設計者在撰寫程式時加入特殊之指引(directive)，才能有效地被執行。

而分析程式是否有效的被執行方面，CRAY公司提供flowview、atscope、atexpert、 hpm、prof、perfview、procstat等工具以協助程式發展者分析程式之特性，而以 flowview、atscope和atexpert三者和往後之討論較有關係，其中flowview可產生關於程式執行時各副程式的相關資訊如總執行時間，呼叫次數，平均執行時間，執行時間百分比等；atscope則可協助使用者判定一變數之屬性，在一迴路內變數之屬性改變可以影響此迴路之向量與平行處理能力；而atexpert 則可顯示各副程式在多重處理器同時執行之平行度。

三、改進過程與結果

原始之程式經Cray 8i編譯自動強化向量與平行處理後以時距2分鐘進行48小時之預報積分時，模式執行所需之總時間大約需13300秒，以6個CPU執行之回應時間大約是6000秒，也就是平均大概只以2.2個CPU來執行，因此程式執行之並行度不佳，由flowview等工具分析之資料（節錄如表一）顯示最花時間之計算在副程式CHEF、VQSAT2、PBLEND、PBLINT、CUPARA等副程式（各副程式之主要功能如附錄一）。

針對計算所需時太久，本研究所採取之改進方法首先著重在增進程式之向量處理，其中主要修改之原則爲：

㈠多層迴路中最內層之迴路儘量爲最長之迴路。

㈡限制了迴路運算向量化之計算，儘可能將它分離於迴路之外，使其它之計算能向量化。

㈢可以合併之計算則將它們合併於同一迴路內，以節省計算時間。

㈣儘量增長迴路的長度，如原來爲二維的計算修改爲一維，可節省計算時間。

㈤儘量減少呼叫副程式。

依以上原則，我們詳細的研讀程式並進行程式改進，其中注意力尤其著重在占了總時間較多的副程式，我們發覺CHEF副程式所以需時很多的主要原因之一爲有二個迴路運算中最短維度之計算被置於最內層，而VQSAT2副程式則主要是因爲被呼叫

太多次，我們將其修改原來在迴路內呼叫之方法爲以陣列暫存，而後在VQSAT2副程式內進行迴路運算，而PBLEND及PBLINT主要費時是因爲進行溫度與位溫互變時需要有X之Y次方計算，這種計算需要長時間，同時有一些計算可以提出在迴路運算之外（如除以X，可以改成乘以X分之一），而經過比原來約13300秒節省了3600秒，約爲原來的0.73倍。

而由flowview的資料（表1.）顯示，此時程式之執行雖然經向量化而增速，CPU並行處理方面仍然不好，由1個CPU執行之時間比例，高平均而言，大致只使用了約2.2個CPU，同時flowview之資

表 1. 各階段區域模式進行48小時預報之程式執行情形之分析

	原有程式	第一次修改後	第二次修改後	第三次修改後
計算總時(秒)	13300	10800	9620	9780
CPU 使用情形(使用 CPU 個數之總秒數)	1 ： 4165 2 ： 147 3 ： 151 4 ： 400 5 ： 813 6 ： 491	1 ： 3636 2 ： 102 3 ： 101 4 ： 302 5 ： 846 6 ： 505	1 ： 3512 2 ： 105 3 ： 86 4 ： 364 5 ： 759 6 ： 347	1 ： 1017 2 ： 143 3 ： 117 4 ： 274 5 ： 830 6 ： 599
平均 CPU 使用個數	2.19	2.30	2.19	3.52
適於 IN-LINING 之副程式	QSAT 、 ESAT	QSAT 、 ESAT	QSAT 、 ESAT	
最費時之前五個副程式	CHEF VQSAT2 PBLEND PBLINT CUPARA	CHEF DIFF CUPARA PBLINT PBLEND	CHEF DIFF CUPARA VERADV ESAT	CHEF CIFF CUPARA HORADV MIXPBL

註:CPU總數≒(1×cpu1＋2×cpu2＋3×cpu3＋4×cpu4＋5×cpu5＋6×cpu6)

這樣之初步改進後，相同積分計算所需之時間由原來約13300秒降至10800秒，同時最費時之計算也依序改變爲CHEF、DIFF、CUPARA、PBLINT及PBLEND等副程式（詳如表一），由此可見程式設計者注意遵行向量化程式設計撰寫原則的重要性。

再進一步之程式改進仍然是著重於計算占時較多的副程式，其中我們發覺，由於Cray8i記憶體(32MW)在執行區域模式時（約需20MW）仍有剩餘，因此，可將程式中原本重複計算的部分利用暫存空間存起來，以減少計算。同時，原以設定DATA值再內插（如飽和水汽壓之計算）之方法，改成直接由六次曲線函數（高次曲線函數使值較精確，由加、減與乘所組合之運算要較如次方等非基本運算者要快）計算要稍微快一些，經過這些改進後程式之執行由原來約10800秒降至約9700秒（表1.），料也顯示可將QSAT、ESAT兩副程式以IN-LINING方式處理，以進一步縮短計算所需時間。因此，再下來之程式改進工作主要爲：

㈠將副程式儘量以IN-LINING處理。

㈡嘗試使用Cray提供之系統程式取代部分計算。

㈢進行程式分析，以提高程式並行處理之能力。

在進行第一項改進時，最適於IN-LINING處理的是獨立的副程式，其被呼叫之次數很多，而每次執行之時間又很短的。另外一種情況是，在迴路內呼叫副程式或副程序時使得整個迴路運算不能向量化。在本研究中我們依照flowview分析後之建議，將QSAT與ESAT兩副程式以IN-LINING處理。其作法是首先在編譯指令CFT77中宣布使用IN-LINING之功能，並且在程式裏加上CDIR$INLINE之指引於QSAT與ESAT之前。

在第二項工作之執行上，主要是針對區域模式中有距陣相乘的計算，這些數學的計算軟體Cray公司已提供有高效率的系統應用程式可供呼叫，目前我們已將距陣相乘的部分以呼叫系統應用程式SGEMM來取代（稍後將提到在副程式FSMODE內因爲距陣長度不一致而導致無法以呼叫SGEMM取代）。

第三部分提高並行處理能力之工作則爲經程式分析劃分並行區域，這其中，相關性分析及排程問題則爲兩個最有用之觀念與技巧。相關性分析大致可分爲資料相關性分析與控制相關性分析（或迴路相關性分析）兩部分。資料相關性分析在討論兩相鄰指令資料之依存情形，以FORTRAN指令而言，等號左邊之變數值取決於等號右邊各變數之值，因此如果我們有兩指令（或擴充爲兩組指令集）S及S'，且S在S'前執行，若以USE表示一指令（或一組組令）所有右邊之變數所成之集合，而DEF表示一指令（或一組指令）所有左邊之變數所成之集合。

如果$DEF(S) \cap USE(S') \neq 0$則資料存在正相關性，如果$USE(S) \cap DEF(S') \neq 0$則資料間存在反相關性，而如$DEF(S) \cap DEF(S') \neq 0$則資料間存在輸出值相關性，一般Cray 8i編譯系統會依照這種資料相關性分析結果決定是否並行處理，當有資料相關性存在時往往限制了並行處理能力，需要予以排除。

控制相關性分析主要以迴路所產生之相關性分析爲主，如果在同一迴路內之指令並無資料相關性，但如將迴路做LOOP UNROLLING時則有可能產生相關性（稱 loop carried dependence）。而這些分析方法主要是對其迴路運算指標加以分析，較常用的有gcd test及Banerjee analysis (Banerjee, 1988; Zima and Chapman, 1990)。

關於排程問題在Polychronopoulos(1988)中有較詳細討論，而我們之詩論主要針對迴路運算在執行時如何分配到各處理器而言，因此，對單一CPU之機器並沒有這方面之問題。以Cray8i電腦而言，其排程之方式主要可分三種：

㈠爲內定值，其內定值一次只分派一個iteration到一個CPU，也就是說如果有一個64次的迴路運算，若有6個CPU則CPUi將執行第i＋6j次的計算，其中$1 \leq i \leq 6$，而$0 \leq j \leq 64/6+1$。

㈡爲guided-self-scheduling，這是依據每次各CPU之使用情況將迴路分配至各 CPU，對各CPU之分配數並不一定每次相同，基本上這種方法是較佳之方法，因其考慮到CPU間負載之平衡。

㈢Vector，這種方法相似於guided-self-scheduling，只不過其小之分配數爲64個，這在迴路內之內體運算較簡單時是有效的方法。

在我們的研究裡，我們總共對區域模式中12個需要執行較久的副程式，依以上之相關性分析及考慮排程問題進行改善，這12個副程式分別是DIFF、FSMODE、 CHEF、MIXPBL、 VERADV、HORADV、CUPARA、CURV、LSPRCP、PRCOR、CLOUDY及INTSTA。於此，我們僅以 FSMODE及HORADV兩個副程式爲例，進一步說明我們對程式改進之作法。

對於FSMODE之改進方面，原來程式如附錄二，由迴路運算上劃分總共可得16小節（S_1到S_{16}），而此16小節其執行順序與相關情形如圖三（因主要在做資料相關性分析，並未進一步了解各小節之實際功能），圖中B_1至B_4分別代表4組區塊，其中依常理B_1和B_2之各計算沒有相依情形應可平行處理，然而B_1和B_2卻使用相同之陣列變數D1，D2及DUMMY爲暫存空間，而無法並行處理，對於這種缺點可以增加變數之方法使它們可以並行處理，而S_4則僅做初始化動作可以將之移至最前面。

在經增加變數及重組後，我們可將FSMODE（程式見附錄三）劃分爲6個並行區域(PARALLEL REGION,R_1至R_6)如圖四，這些並行區域內之計算單元可以同時被多CPU來執行而和次序無關，各並行區域之範圍在程式內以DO ALL或PARALLEL及END PARALLEL來界定。此兩者之不同點在於DO ALL使用於緊接著的迴路，而PARALLEL至 END PARALLEL內則可含數個迴路和其它計算單元。以FSMODE內R_5而言，即含了三個可並行處理之迴路區塊P_6、P_7及P_8。

在程式之修改上則要引用Cray編譯系統的指引來達到劃分並行區域等之目的，這其中又分爲CDIR$（對於編譯器），CFPP$（對FPP）及CMIC$

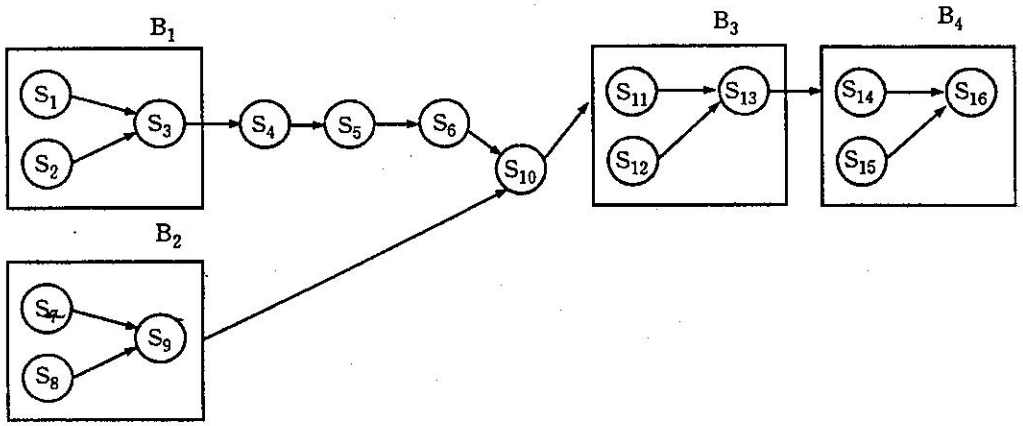

圖 3. 副程式FSMODE之執行循序結構。

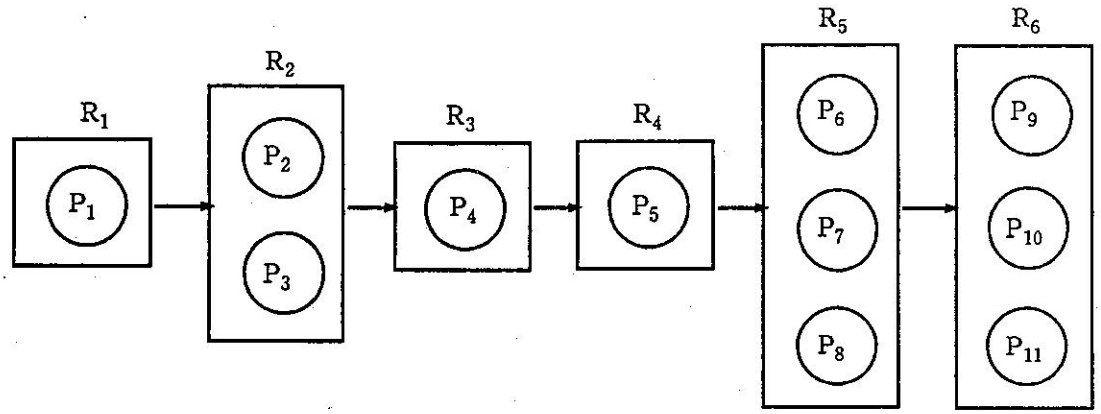

圖 4. :修改後FSMODE副程式之執行循序結構。

（對FMP）三類。在R_1裡我們共用了3個指引，其中CFPP$ UNROLL 4R指示FPP在這副程式中當遇到迴路運算的長度小於等於4時就將該LOOP做UNROLLING。

CFPP$PRIVATE ARRAY使FPP將下面緊接的迴路運算，當在並行處理時各CPU間的資料彼此無關，不需做同步協調。

CFPP$SELECT(CONCUR)使FPP在下面緊接的迴路中，選擇以迴路爲整體單位送至各CPU執行，通常平行化是針對最外層的迴路，向量化是針對最內層的迴路，透過此指引可改變這個規則。

在R_2中，我們利用PARALLEL及END PARALLEL來宣告一PARALLEL REGION，逐行之意義爲：

CMIC$PARALLEL AUTOSCOPE 使FMP認知爲PARALLEL REGION的開始，AUTOSCOPE則啓動編譯程式在測試平行區域時變數的內隱規則。

CMIC$1 SHARED使FMP在設定CPU的私有變數時，讓各CPU有獨立的值。

CMIC$CONTROL告訴FMP整個平行區域中用來做控制的變數名稱，當編釋程式遇到有未經宣告的變數時，即以CONTROL內宣告變數爲依據以判定其屬性。

CMIC$ DO PARALLEL GUIDED宣告下面的段落可以被多CPU並行處理，其執行迴路運算分配到各CPU之方式採前述之guided-self-scheduling的方法。

CFPP$PRIVATE ARRAY相似於 CMIC$ PRIVATE

CFPP$NODEPCHK使FPP在緊接著的迴路中不須做

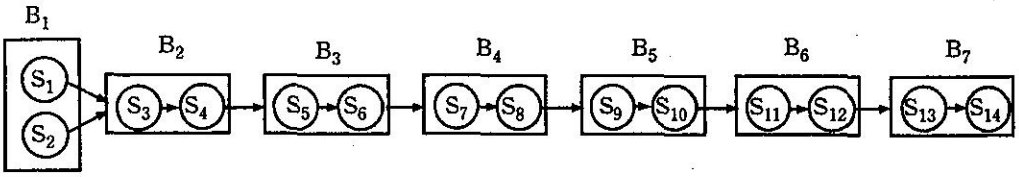

圖 5. :副程HORADV執行循序結構

相關性分析。

因此使用時設計者必須自行分析，確保真的沒有相關生存在，否則結果將產生錯誤。

CMIC$ END DO爲 DO PARALLEL區域之結束。

在此平行區域R_2中，共有二個平行段落P_2及P_3，其執行順序可在執行時才決定，並可平行處理。平行區域R_3和R_2相似，唯一的不同只在於平行度僅及於緊接其後的那一個迴路。

在FSMODE副程式中另外加入之指引是CMIS$ DO ALL AUTOSCOPE GUIDED，它是告訴FMP這一段是PARALLEL REGION，並採用guided-self-scheduling的方式，並且在遇到有未宣告變數之屬性時使用內隱之規則。

總結而言，對FSMODE副程式，主要是透過程式分析後以增加區域變數之方式消除相關性而提高副程式執行時之平行度。而在R_6中的三個區塊原本可用呼叫系統提供之應用程式SGEMM而取代，但因區塊中之陣列大小不定，以致無法使用，根據實驗，呼叫SGEMM的速度約爲該四層迴路運算的三分之一，因此如何進一步改進資料之結構，使能以SGEMM取代，成爲下一階段的研究重點。

關於對副程式HORADV之改進方面，HORADV之原來程式如附錄四，執行順序與相關性如圖五。由於S_3必須用到S_1及S_2的輸出，是以S_3必須在S_1及S_2後執行，但B_2到B_7原本可以互相獨立，但因爲共用了陣列變數DUMMY，以致無法並行執行，因此這程式仍可以增加區域變數之使用來提高平行度。而實際上在三維變數DUMMY裡，僅用到前面二維，因此B_2到B_7若使用不同之第三維變數（K值）則可清除B_2到B_7等6區塊之相關性，改善後之程式（見附錄五）之執行結構如圖六。由圖六可知

在新的程式中，共分爲R_1、R_2及R_3等3個平行區域，10個平行區塊（P_1至P_{10}），已有相當高的平行度。而由此部分程式之修改經驗，我們也體會到善

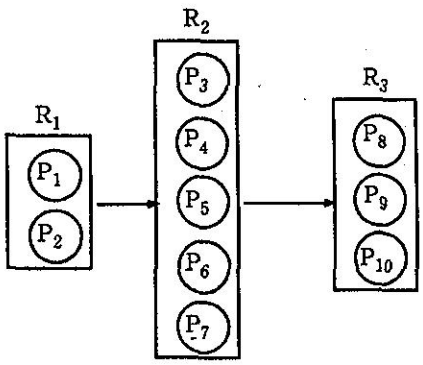

圖 6. :修改後HORADV副程式之執行循序結構

用記憶體之重要性，在沒有新增記憶體的情況下，改變暫存空間之使用方式卻提高了程式之平行處理能力。

另外值得一提的是，在進行此部分程式修改之時，我們亦進行了決定採用並行處理之測試，在每一平行區域之前我們設定一個標準值，倘若在執行時迴路之執行次數低於這個值則不做並行處理。由Cray所提供資料可知Cray YMP在做並行處理的標準值約爲800個計算單位，而這標準值(T)之求法爲

T＝800/CHIME

其中CHIME＝（區域中執行"＋"和"－"之總數，區域中"*"和3倍"/"的總數，區域中記憶體存入的次數，（區域中記憶體讀的次數＋1）/2）以上各項數值中之最大(MAX)值。

依此求法，在HORADV副程式中各PARALLEL REGION之T值爲

R_1:800/MAX(4,4,2,9/2)≅ 200

R_2:800/MAX(4,1,1,3/2)≅ 200（但我們取高標準400)

R_3:800/MAX(6,3,1,4)≅ 133（取高標準200）

同時，因爲在R_1中，迴路內的運算較簡單，我們採用Vector的方式，亦即每次以最少64個迴路內體運算的方式做guided-self-scheduling，而在R_2及R_3

中，因迴路內運算較繁雜，故採用一般guided-self-scheduling的方式。

而在完成以上增進12個副程式平行度後，程式之總執行時間（如表一）稍爲增加，然而卻有效地使一個CPU同時執行之時間則分別由約758秒增加至約829秒以及由346秒增加至599秒。如此使得平均程式之執行由原先使用約2.19個CPU提昇至約3.52個CPU，並使得48小時之預報由原先約5170秒縮短至約3000秒。

四、結論與心得

中央氣象局在全面業務電腦化第二期計畫下引進了Cray YMP 8i，並在其上發展各預報模式，由於此電腦爲具6個CPU的超級電腦，因此如何充分地應用這新電腦所具快速之向量處理及多CPU並行處理的能力，成爲發展預報系統外之另一重要工作。

本研究首先在熟悉CRAY 8i之特性、其所提供各種程式執行之分析輔助工具及應用軟體後進行對原區域預報模式程式之分析，而後引用向量處理與平行處理之要點，針對程式中需時最久之部分進行改善，改善之步驟則在首先提高程式之向量處理能力，而後引用系統應用程式呼叫及IN-LINNING處理，最後透過減少相關性後提高程式運算之平行度。

經過這項研究改善，使得程式原先對48小時預報總共約需13300秒,而並行度約2.2CPU，縮短爲總需時約9700秒，並提高並行度達3.5CPU，這結果使原先約需6000秒完成之工作降至約3000秒，執行時間縮短爲原來之一半。而其中增加向量化部分，大致可使預報總需時降至9600秒，即約爲原來之2/3。不過，提高向量化並未有效改善程式之可並行度。

在研究的過程中，我們也有以下一些心得：

㈠Cray YMP在編譯系統方面，對向量化已較成熟，只要在撰寫程式時遵照增進向量處理之常規，則可得到相當好的效率，而系統所提供的一些工具也可以相當有效地協助使用者分析程式之計算效率。在平行處理方面，現有之編譯系統相較下效能並不好，需要在撰寫程式時加入適當指引，才能有效提高計算效率。

㈡在程式記憶體之管理方面，我們發覺一般程式設計者仍會採用以往單一CPU宣告一大塊COMMON BLOCK做爲共用區之習慣，而在此COMMON BLOCK內含許多暫存需要的陣列，以共用記憶空間。然而這種設計方法，往往增加了資料之相關性，而減低程式之並行度，我們的建議是使用者應儘量宣告必要的共用變數於COMMON BLOCK，而將許多的暫存用陣列宣告爲區域變數。

㈢在程式中EQUIVALENCE的宣告，往往使程式分析的困難度增加，應儘可能少用。

㈣在陣列的處理方面，在程式中有許多地方都存有邊界值的問題，若能進一步妥善處理這些邊界，當可再提高其效率。

㈤區域模式裡因含有粗及細兩組格點，而兩者格點數並不相同，但又大部分使用同一副程式，這樣的安排有時會因迴路之計算，並不是對所有宣告陣距大小之全部，而影響了向量處理或呼叫系統軟體。因此，對程式整個資料及程式架構做進一步分析與改變，將可進一步提高程式之執行效率。

五、致　　謝

本文在中央氣象局研究發展專題CB83-1A05支助下完成，期間承蒙戴金生博士夫婦、彭順台教授及李尙武博士提供建議，謹此致謝。

六、參考文獻

Banerjee, U., 1988: Dependence analysis for supercomputing. Kluwer Academic Publishers, 155pp.

Hwang, K., 1993: Advanced computer architecture: Parallelism, scalability, programmability. McGraw-Hill, 770pp.

Polychronopoulos, C.D., 1988: Parallel programming and compilers. Kluwer Academic Publishers, 240pp.

Zima, H., and B. Chapman, 1990: Supercompilers for parallel and vector computers. ACM Press, Addison Wesley, 376pp.

附錄一：有限區域預報模式部分副程式之主要功能。

CHEF：以完成圖二之積分計算。

CLOUDY：積雲參數化處理中，計算雲內之溫度濕度等結構。

CUPARA：處理郭氏積雲參數化。

CURV：地表曲率效應。

DIFF：擴散處理。

ESAT：計算飽和水汽壓。

FSMODE：移動較快波動部分之修正。

HORADV：水平平流。

MIXPBL：處理混合邊界層。

PBLINT：溫度換算位溫爲地表邊界層之前置處。

PBLEND：爲地表邊界層之後續處理（主要將位溫換爲溫度）

QSAT：計算飽和比濕。

VERADV：垂直平流。

VQSAT2：計算飽和比濕。

LSPRCP：處理大尺度降水。

PRCOR：計算壓力梯度與科氏力作用。

INTSTA：模式變數單位轉換。

附錄二：改進前之FSMODE副程式

```
      SUBROUTINE FSMODE(U1,V1,T1,PS1,U2,V2,T2,PS2,M,N,M1,N1)
      include 'HXNXKK'
      PARAMETER (K1=KK-1,KM=3)
      DIMENSION U1(M1,N,KK),V1(M,N1,KK),T1(M,N,KK),PS1(M,N),
     1          U2(M1,N,KK),V2(M,N1,KK),T2(M,N,KK),PS2(M,N)
      COMMON/BLK1/DELX,DELY,DELXSQ,DELYSQ,DELT,R,CP,G,
     1 HXM(MX,NX),HXU(MX,NX),HXV(MX,NX),corols(mx,nx),
     2 THEAN(KK),SIGMA(KK),RTSTAR(KK),PSTAR(KK),DEL2(KK),INT(KM)
      COMMON/BLK2/AN1(K1,KK),AN2(KK),AM1(KK,KK),AM1INV(KK,KK),AM2(KK,KK)
     1    ,AM3(KK,KK),AM3INV(KK,KK),AM4(KK,KK),AM5(KK)
     2    ,AM6(KK,KK),AM7(KK)
      COMMON/BLK3/EV(KK),VEC(KK,KK),VECINV(KK,KK)
      COMMON/BLK4/F1(MX,NX,KM),F2(MX,NX,KM),PMODE(MX,NX,KM),
     1   DMODE(MX,NX,KM)
      COMMON/BLK9/ D1(MX,NX),D2(MX,NX),DUMMY(MX,NX,KK)
      COMMON/BLK2022/HMMX(Mx,Nx),HMMY(Mx,Nx),
     1              HUUX(Mx,Nx),HUUY(Mx,Nx),
     1              HVVX(Mx,Nx),HVVY(Mx,Nx),
     3              HmX(Mx,Nx),HmY(Mx,Nx),
     3              HUX(Mx,Nx),HVY(Mx,Nx),
     4              DHV(Mx,Nx),DHU(Mx,Nx),
     5              HMMXX(Mx,Nx),HMMYY(Mx,Nx),
     1              HUUXX(Mx,Nx),HUUYY(Mx,Nx),
     1              HVVXX(Mx,Nx),HVVYY(Mx,Nx)
      DO 197 J=2,N1
      DO 197 I=1,M1
S1    D1(I,J)=(PS1(I+1,J)+PS1(I,J))*0.5*HxU(i,J)
  197 CONTINUE
      DO 198 J=1,N1
      DO 198 I=2,M1
S2    D2(I,J)=(PS1(I,J+1)+PS1(I,J))*0.5*HXV(i,J)
  198 CONTINUE
      DO 200 K=1,KK
      DO 200 J=2,N1
      DO 200 I=2,M1
S3    DUMMY(I,J,K)= (D1(I,J)*U1(I,J,K)-D1(I-1,J)*U1(I-1,J,K))*hmmx(I,j)+
     1              (D2(I,J)*V1(I,J,K)-D2(I,J-1)*V1(I,J-1,K))*hmmy(i,J)
  200 CONTINUE
      DO 204 K=1,KM
      DO 204 J=1,N
      DO 204 I=1,M
S4    F1(I,J,K)=0.0
      F2(I,J,K)=0.0
      DMODE(I,J,K)=0.0
      PMODE(I,J,K)=0.0
  204 CONTINUE

      print *, 'Start calling fsmode'

      DO 205 KA=1,KM
      DO 205 K=1,KK
S5    DO 205 J=2,N1
      DO 205 I=2,M1
      F1(I,J,KA)=F1(I,J,KA)-VECINV(KA,K)*DUMMY(I,J,K)
  205 CONTINUE
      DO 203 K=1,KM
      DO 203 J=2,N1
S6    DO 203 I=2,M1
      DMODE(I,J,K)=-F1(I,J,K)
  203 CONTINUE
      DO 172 J=2,N1
      DO 172 I=1,M1
S7    D1(I,J)=(PS2(I+1,J)+PS2(I,J))*0.5*HxU(i,J)
  172 CONTINUE
      DO 173 J=1,N1
      DO 173 I=2,M1
S8    D2(I,J)=(PS2(I,J+1)+PS2(I,J))*0.5*HXV(i,J)
  173 CONTINUE
      DO 206 K=1,KK
      DO 206 J=2,N1
S9    DO 206 I=2,M1
      DUMMY(I,J,K)=(D1(I,J)*U2(I,J,K)-D1(I-1,J)*U2(I-1,J,K))*hmmx(I,j)+
     1             (D2(I,J)*V2(I,J,K)-D2(I,J-1)*V2(I,J-1,K))*hmmy(i,J)
  206 CONTINUE
      DO 208 KA=1,KM
      DO 208 K=1,KK
S10   DO 208 J=2,N1
      DO 208 I=2,M1
      DMODE(I,J,KA)=DMODE(I,J,KA)-VECINV(KA,K)*DUMMY(I,J,K)
  208 CONTINUE
      DO 210 K=1,KM
      DO 210 J=2,N1
S11   DO 210 I=2,M1
      F2(I,J,K)=-AM7(K)*PS1(I,J)
  210 CONTINUE
      DO 212 K=1,KK
      DO 212 J=2,N1
S12   DO 212 I=2,M1
      DUMMY(I,J,K)=PS1(I,J)*T1(I,J,K)
  212 CONTINUE
      DO 215 K=1,KM
      DO 215 KA=1,KK
S13   DO 215 J=2,N1
      DO 215 I=2,M1
      F2(I,J,K)=F2(I,J,K)-AM6(K,KA)*DUMMY(I,J,KA)
  215 CONTINUE
      DO 220 K=1,KM
      DO 220 I=1,M
S14   DO 220 J=1,N
      PMODE(I,J,K)=AM7(K)*(PS1(I,J)-PS2(I,J))
  220 CONTINUE
      DO 222 K=1,KK
      DO 222 J=1,N
S15   DO 222 I=1,M
      DUMMY(I,J,K)=(PS1(I,J)*T1(I,J,K)-PS2(I,J)*T2(I,J,K))
  222 CONTINUE
      DO 225 K=1,KM
      DO 225 KA=1,KK
S16   DO 225 J=1,N
      DO 225 I=1,M
      PMODE(I,J,K)=PMODE(I,J,K)+AM6(K,KA)*DUMMY(I,J,KA)
  225 CONTINUE

      RETURN
      END
```

附錄三：經改善平行處理後之FSMODE副程式

```
      SUBROUTINE FSMODE(U1,V1,T1,PS1,U2,V2,T2,PS2,M,N,M1,N1)
      include 'HXNXKK'
      PARAMETER (K1=KK-1,KM=3)
      DIMENSION U1(M1,N,KK),V1(M,N1,KK),T1(M,N,KK),PS1(M,N),
     1          U2(M1,N,KK),V2(M,N1,KK),T2(M,N,KK),PS2(M,N)
      COMMON/BLK1/DELX,DELY,DELXSQ,DELYSQ,DELT,R,CP,G,
     1 HXM(MX,NX),HXU(MX,NX),HXV(MX,NX),corols(mx,nx),
     2 THEAN(KK),SIGMA(KK),RTSTAR(KK),PSTAR(KK),DEL2(KK),INT(KM)
      COMMON/BLK2/AN1(K1,KK),AN2(KK),AM1(KK,KK),AM1INV(KK,KK),AM2(KK,KK)
     1    ,AM3(KK,KK),AM3INV(KK,KK),AM4(KK,KK),AM5(KK)
     2    ,AM6(KK,KK),AM7(KK)
      COMMON/BLK3/EV(KK),VEC(KK,KK),VECINV(KK,KK)
      COMMON/BLK4/F1(MX,NX,KM),F2(MX,NX,KM),PMODE(MX,NX,KM),
     1   DMODE(MX,NX,KM)
      COMMON/BLK9/ D1(MX,NX),D2(MX,NX),DUMMY(MX,NX,KK)
      COMMON/BLK2022/HMMX(Mx,Nx),HMMY(Mx,Nx),
     1              HUUX(Mx,Nx),HUUY(Mx,Nx),
     1              HVVX(Mx,Nx),HVVY(Mx,Nx),
     3              HmX(Mx,Nx),HmY(Mx,Nx),
     3              HUX(Mx,Nx),HVY(Mx,Nx),
     4              DHV(Mx,Nx),DHU(Mx,Nx),
     5              HMMXX(Mx,Nx),HMMYY(Mx,Nx),
     1              HUUXX(Mx,Nx),HUUYY(Mx,Nx),
     1              HVVXX(Mx,Nx),HVVYY(Mx,Nx)

      DIMENSION D1197(MX,NX),DZ198(MX,NX),DUMMYG0(MX,NX,KK)
      DIMENSION DUMMYD6(MX,NX,KK), DUMMY12(MX,NX,KK)

CFPP$ UNROLL 4 R

CFPP$ PRIVATEARRAY
      DO 204 K=1,KM
CFPP$ SELECT (CONCUR)
      DO 204 J=1,M
      DO 204 I=1,M
P1    F1(I,J,K)=0.0
      F2(I,J,K)=0.0
      DMODE(I,J,K)=0.0
      PMODE(I,J,K)=0.0
  204 CONTINUE
```

```
CMIC$ PARALLEL AUTOSCOPE
CMIC$1 SHARED (N1,M1,D1197,PS1,HXU,HXV,D2198)
CMIC$2 SHARED (D1,D2,PS2)
CMIC$3 PRIVATE (I,J)
CMIC$4 CONTROL (I,J)

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
CFPP$ NODEPCHK
      DO 197 J=2,N1
P2    DO 197 I=1,M1
      D1197(I,J)=(PS1(I+1,J)+PS1(I,J))*0.5*HXU(I,J)
      D1(I,J)=(PS2(I+1,J)+PS2(I,J))*0.5*HXU(I,J)
  197 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
CFPP$ NODEPCHK
      DO 198 J=1,N1
P3    DO 198 I=2,M1
      D2198(I,J)=(PS1(I,J+1)+PS1(I,J))*0.5*HXV(I,J)
      D2(I,J)=(PS2(I,J+1)+PS2(I,J))*0.5*HXV(I,J)
  198 CONTINUE
CMIC$ END DO

CMIC$ END PARALLEL

CMIC$ DO ALL AUTOSCOPE GUIDED
CMIC$1 SHARED (DUMMY00,D1197,D2198,U1,V1,HMMX,HMMY,KK,N1,M1)
CMIC$2 SHARED (DUMMY06,D1,D2,U2,V2,DUMMY12,T1,PS1)
CMIC$3 PRIVATE (I,J,K)
CMIC$4 CONTROL (I,J,K)

CFPP$ PRIVATEARRAY
      DO 200 K=1,KK
CFPP$ NODEPCHK
      DO 200 J=2,N1
      DO 200 I=2,M1
      DUMMY00(I,J,K)= (D1197(I,J)*U1(I,J,K)-D1197(I-1,J)*U1(I-1,J,K))
     1              *hmmx(i,j)+
P4   1               (D2198(I,J)*V1(I,J,K)-D2198(I,J-1)*V1(I,J-1,K))
     1              *hmmy(i,j)
      DUMMY06(I,J,K)=(D1(I,J)*U2(I,J,K)-D1(I-1,J)*U2(I-1,J,K))
     1              *hmmx(i,j)+
     1              (D2(I,J)*V2(I,J,K)-D2(I,J-1)*V2(I,J-1,K))*hmmy(i,j)
      DUMMY12(I,J,K)=PS1(I,J)*T1(I,J,K)
  200 CONTINUE
c     print *, 'Start calling fsmode'

CMIC$ DO ALL AUTOSCOPE GUIDED
CMIC$1 SHARED (KK,N1,M1,F1,VECINV,DUMMY00)
CMIC$2 PRIVATE (I,J,K)
CMIC$3 CONTROL (I,J,K)

CFPP$ UNROLL 3 L
      DO 205 KA=1,KM
      DO 205 K=1,KK
CFPP$ NODEPCHK
P5    DO 205 J=2,N1
      DO 205 I=2,M1
      F1(I,J,KA)=F1(I,J,KA)-VECINV(KA,K)*DUMMY00(I,J,K)
  205 CONTINUE

CMIC$ PARALLEL AUTOSCOPE
CMIC$1 SHARED (N1,M1,DMODE,F2,F1,AM7,PS1,PMODE,PS2,AM7)
CMIC$2 SHARED (M,N,KK,DUMMY,T1,T2)
CMIC$3 PRIVATE (I,J)
CMIC$4 CONTROL (I,J)

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
CFPP$ NODEPCHK
      DO 203 J=2,N1
      DO 203 I=2,M1
P6    DO 203 K=1,KM
      DMODE(I,J,K)=-F1(I,J,K)
      F2(I,J,K)=-AM7(K)*PS1(I,J)
  203 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
CFPP$ NODEPCHK
      DO 220 I=1,M
P7    DO 220 J=1,N
      DO 220 K=1,KM
      PMODE(I,J,K)=AM7(K)*(PS1(I,J)-PS2(I,J))
  220 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
      DO 222 K=1,KK
CFPP$ NODEPCHK
P8    DO 222 J=1,N
      DO 222 I=1,M
      DUMMY(I,J,K)=(PS1(I,J)*T1(I,J,K)-PS2(I,J)*T2(I,J,K))
  222 CONTINUE
CMIC$ END DO

CMIC$ END PARALLEL

CMIC$ PARALLEL AUTOSCOPE
CMIC$1 SHARED (KK,N1,M1,N,M,DMODE,VECINV,DUMMY06)
CMIC$2 SHARED (F2,AM6,DUMMY12,PMODE,DUMMY)
CMIC$3 PRIVATE (I,J,KA)
CMIC$4 CONTROL (I,J,KA)

CMIC$ DO PARALLEL GUIDED
CFPP$ UNROLL 3 L
      DO 208 KA=1,KM
      DO 208 K=1,KK
P9    CFPP$ NODEPCHK
      DO 208 J=2,N1
      DO 208 I=2,M1
      DMODE(I,J,KA)=DMODE(I,J,KA)-VECINV(KA,K)*DUMMY06(I,J,K)
  208 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL GUIDED
CFPP$ UNROLL 3 L
      DO 215 K=1,KM
      DO 215 KA=1,KK
P10   CFPP$ NODEPCHK
      DO 215 J=2,N1
      DO 215 I=2,M1
      F2(I,J,K)=F2(I,J,K)-AM6(K,KA)*DUMMY12(I,J,KA)
  215 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL GUIDED
CFPP$ UNROLL 3 L
      DO 225 K=1,KM
      DO 225 KA=1,KK
CFPP$ NODEPCHK
P11   DO 225 J=1,N
      DO 225 I=1,M
      PMODE(I,J,K)=PMODE(I,J,K)+AM6(K,KA)*DUMMY(I,J,KA)
  225 CONTINUE
CMIC$ END DO

CMIC$ END PARALLEL

c     do 997 i=1,10
c        print *,dmode(i,4,4)
c 997 continue

      RETURN
      END
```

附錄四：改進前之HORADV副程式

```
      SUBROUTINE HORADV(U1,U2,V1,V2,T1,T2,PS1,M,N,M1,N1)
      include 'HXNKKK'
      PARAMETER (MX1=MX-1,NX1=NX-1,K1=KK-1)
      PARAMETER (KM=3)
      DIMENSION U1(M1,N,KK),U2(M1,N,KK),V1(M,N1,KK),V2(M,N1,KK),
     1   T1(M,N,KK),T2(M,N,KK),PS1(M,N)
      COMMON/BLK1/DELX,DELY,DELXSQ,DELYSQ,DELT,R,CP,G,
     1 HXM(MX,NX),HXU(MX,NX),HXV(MX,NX),corols(mx,nx),
     2 TMEAN(KK),SIGMA(KK),RTSTAR(KK),PSTAR(KK),DEL2(KK),INT(KM)
      COMMON/BLK9/ D1(MX,NX),D2(MX,NX),DUMMY(MX,NX,KK)
      COMMON/BLK2022/HMMX(Mx,Nx),HMMY(Mx,Nx),
     1              HUUX(Mx,Nx),HUUY(Mx,Nx),
     1              HVVX(Mx,Nx),HVVY(Mx,Nx),
     1              HmX(Mx,Nx),HmY(Mx,Nx),
     3              HUX(Mx,Nx),HVY(Mx,Nx),
     4              DHV(Mx,Nx),DHU(Mx,Nx),
     5              HMMXX(Mx,Nx),HMMYY(Mx,Nx),
     1              HUUXX(Mx,Nx),HUUYY(Mx,Nx),
     1              HVVXX(Mx,Nx),HVVYY(Mx,Nx)

      M2=M-2
      N2=N-2
      DO 200 K=1,KK
      DO 90 J=2,N1
      DO 90 I=1,M1
S1    D1(I,J)=(PS1(I+1,J)+PS1(I,J))*HxU(I,J)*U1(I,J,K)
   90 CONTINUE
      DO 95 J=1,N1
S2    DO 95 I=2,M1
      D2(I,J)=(PS1(I,J+1)+PS1(I,J))*HXV(I,J)*V1(I,J,K)
   95 CONTINUE
      DO 100 J=2,N1
S3    DO 100 I=2,M1
      DUMMY(I,J,K)=(D1(I,J)+D1(I-1,J))*(U1(I,J,K)+U1(I-1,J,K))
  100 CONTINUE
      DO 105 J=2,N1
S4    DO 105 I=2,M2
      U2(I,J,K)=U2(I,J,K)-(DUMMY(I+1,J,K)-DUMMY(I,J,K))*huux(i,j)/8.
  105 CONTINUE
      DO 110 J=1,N1
S5    DO 110 I=2,M2
      DUMMY(I,J,K)=(D2(I+1,J)+D2(I,J))*(U1(I,J,K)+U1(I,J+1,K))
  110 CONTINUE
      DO 115 J=2,N1
S6    DO 115 I=2,M2
      U2(I,J,K)=U2(I,J,K)-(DUMMY(I,J,K)-DUMMY(I,J-1,K))*huuy(i,j)/8.
  115 CONTINUE
      DO 120 J=2,N2
S7    DO 120 I=1,M1
      DUMMY(I,J,K)=(D1(I,J+1)+D1(I,J))*(V1(I+1,J,K)+V1(I,J,K))
  120 CONTINUE
      DO 125 J=2,N2
S8    DO 125 I=2,M1
      V2(I,J,K)=V2(I,J,K)-(DUMMY(I,J,K)-DUMMY(I-1,J,K))*hvvx(i,j)/8.
  125 CONTINUE
      DO 130 J=2,N1
S9    DO 130 I=2,M1
      DUMMY(I,J,K)=(D2(I,J)+D2(I,J-1))*(V1(I,J,K)+V1(I,J-1,K))
  130 CONTINUE
      DO 135 J=2,N2
S10   DO 135 I=2,M1
      V2(I,J,K)=V2(I,J,K)-(DUMMY(I,J+1,K)-DUMMY(I,J,K))*hvvy(i,j)/8.
  135 CONTINUE
      DO 140 J=2,N1
S11   DO 140 I=1,M1
      DUMMY(I,J,K)=D1(I,J)*(T1(I+1,J,K)+T1(I,J,K)-2.0*TMEAN(K))
```

```
  140 CONTINUE
      DO 145 J=2,N1
      DO 145 I=2,M1
      T2(I,J,K)=T2(I,J,K)-(DUMMY(I,J,K)-DUMMY(I-1,J,K))*hmmx(i,j)/4.
  145 CONTINUE
      DO 150 J=1,N1
      DO 150 I=2,M1
      DUMMY(I,J,K)=D2(I,J)*(T1(I,J+1,K)+T1(I,J,K)-2.0*THEAN(K))
  150 CONTINUE
      DO 155 J=2,N1
      DO 155 I=2,M1
      T2(I,J,K)=T2(I,J,K)-(DUMMY(I,J,K)-DUMMY(I,J-1,K))*hmmy(i,j)/4.
  155 CONTINUE
  200 CONTINUE
      RETURN
      END
```

附錄五：經改善平行處理後之HORADV副程式

```
      SUBROUTINE HORADV(U1,U2,V1,V2,T1,T2,PS1,M,N,M1,N1)
      include 'MXNXKK'
      PARAMETER (MX1=MX-1,NX1=NX-1,K1=KK-1)
      PARAMETER (KM=3)
      DIMENSION U1(M1,N,KK),U2(M1,N,KK),V1(M,N1,KK),V2(M,N1,KK),
     1   T1(M,N,KK),T2(M,N,KK),PS1(M,N)
      COMMON/BLK1/DELX,DELY,DELXSQ,DELYSQ,DELT,R,CP,G,
     1 HXM(MX,NX),HXU(MX,NX),HXV(MX,NX),corols(mx,nx),
     2 THEAN(KK),SIGMA(KK),RTSTAR(KK),PSTAR(KK),DELZ(KK),INT(KM)
      COMMON/BLK9/ D1(MX,NX),D2(MX,NX),DUMMY(MX,NX,KK)
      COMMON/BLK2022/HMMX(Mx,Nx),HMMY(Mx,Nx),
     1              HUUX(Mx,Nx),HUUY(Mx,Nx),
     1              HVVX(Mx,Nx),HVVY(Mx,Nx),
     3              HmX(Mx,Nx),HmY(Mx,Nx),
     3              HUX(Mx,Nx),HVY(Mx,Nx),
     4              DHV(Mx,Nx),DHU(Mx,Nx),
     5              HMMXX(Mx,Nx),HMMYY(Mx,Nx),
     1              HUUXX(Mx,Nx),HUUYY(Mx,Nx),
     1              HVVXX(Mx,Nx),HVVYY(Mx,Nx)

      M2=M-2
      N2=N-2

      DO 200 K=1,KK

CMIC$ DO ALL AUTOSCOPE VECTOR IF (N1*M1 .GT. 200)
CMIC$1 SHARED (D1,B2,N1,M1,U1,V1,HXU,HXV,PS1,K)
CMIC$2 PRIVATE (I,J)
CMIC$3 CONTROL (I,J,K)
      DO 90 J=1,N1
      DO 90 I=1,M1
P1    D1(I,J)=(PS1(I+1,J)+PS1(I,J))*HxU(i,J)*U1(I,J,K)
      D2(I,J)=(PS1(I,J+1)+PS1(I,J))*HXV(I,J)*V1(I,J,K)
   90 CONTINUE

CMIC$ PARALLEL AUTOSCOPE IF (N1*M1 .GT. 400)
CMIC$1 SHARED (DUMMY,D1,D2,U1,V1,THEAN,T1,N1,M1,N2,M2,K)
CMIC$2 PRIVATE (I,J)
CMIC$3 CONTROL (I,J,K)

CMIC$ DO PARALLEL VECTOR
CFPP$ PRIVATEARRAY
      DO 100 J=2,N1
P2    DO 100 I=2,M1
      DUMMY(I,J,1)=(D1(I,J)+D1(I-1,J))*(U1(I,J,K)+U1(I-1,J,K))
  100 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL VECTOR
CFPP$ PRIVATEARRAY
      DO 110 J=1,N1
P3    DO 110 I=2,M2
      DUMMY(I,J,2)=(D2(I+1,J)+D2(I,J))*(U1(I,J,K)+U1(I,J+1,K))
  110 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL VECTOR
CFPP$ PRIVATEARRAY
      DO 120 J=2,N2
P4    DO 120 I=1,M1
      DUMMY(I,J,3)=(D1(I,J+1)+D1(I,J))*(V1(I+1,J,K)+V1(I,J,K))
  120 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL VECTOR
CFPP$ PRIVATEARRAY
      DO 130 J=2,N1
P5    DO 130 I=2,M1
      DUMMY(I,J,4)=(D2(I,J)+D2(I,J-1))*(V1(I,J,K)+V1(I,J-1,K))
  130 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL VECTOR
CFPP$ PRIVATEARRAY
      DO 140 J=2,N1
P6    DO 140 I=1,M1
      DUMMY(I,J,5)=D1(I,J)*(T1(I+1,J,K)+T1(I,J,K)-2.0*THEAN(K))
  140 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL VECTOR
CFPP$ PRIVATEARRAY
      DO 150 J=1,N1
P7    DO 150 I=2,M1
      DUMMY(I,J,6)=D2(I,J)*(T1(I,J+1,K)+T1(I,J,K)-2.0*THEAN(K))
  150 CONTINUE
CMIC$ END DO

CMIC$ END PARALLEL

CMIC$ PARALLEL AUTOSCOPE IF (N1*M1 .GT. 200)
CMIC$1 SHARED (DUMMY,U2,V2,T2,N1,M1,N2,M2,K)
CMIC$2 SHARED (HUUX,HUUY,HVVY,HVVX,HVVY,HMMX,HMMY)
CMIC$3 PRIVATE (I,J)
CMIC$4 CONTROL (I,J,K)

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
      DO 105 J=2,N1
P8    DO 105 I=2,M2
      U2(I,J,K)=U2(I,J,K)-((DUMMY(I+1,J,1)-DUMMY(I,J,1))*huux(i,j)
     1                    +(DUMMY(I,J,2)-DUMMY(I,J-1,2))*huuy(i,j))*0.125
  105 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
      DO 125 J=2,N2
P9    DO 125 I=2,M1
      V2(I,J,K)=V2(I,J,K)-((DUMMY(I,J,3)-DUMMY(I-1,J,3))*hvvx(i,j)
     1                    +(DUMMY(I,J+1,4)-DUMMY(I,J,4))*hvvy(i,j))*0.125
  125 CONTINUE
CMIC$ END DO

CMIC$ DO PARALLEL GUIDED
CFPP$ PRIVATEARRAY
      DO 145 J=2,N1
P10   DO 145 I=2,M1
      T2(I,J,K)=T2(I,J,K)-((DUMMY(I,J,5)-DUMMY(I-1,J,5))*hmmx(i,j)
     1                    +(DUMMr(I,J,6)-DUMMY(I,J-1,6))*hmmy(i,j))*0.25
  145 CONTINUE
CMIC$ END DO

CMIC$ END PARALLEL

  200 CONTINUE

c     do 997 i=1,10
c       print *, t2(i,4,4)
c 997 continue

      RETURN
      END
```

PRELIMINARY RESULTS ON THE IMPROVEMENT OF THE VECTORIZATION AND PARALLELIZATION OF A LIMITED-AREA FORECAST MODEL ON THE CRAY YMP COMPUTER

T.C. Yeh H.W. Jyan

Central Weather Bureau
Taipei, Taiwan, R.O.C

ABSTRACT

To improve the computational efficiency is one of the essential tasks for an operational numerical weather prediction center to be a success. The Central Weather Bureau (CWB) developed a limited-area forecast model in their second phase numerical weather prediction project. This paper reports some results of an attempt to improve the computational efficiency of the model on the six-processor CrayYMP 8i super-computer

The model source codes were first been examined to eliminate the unnecessary computations. Some temporary memory spaces were added to store values for saving computations. System provided utilities (such as flowview, atexpeut and atscope) then were applied to analyze the computational efficiency and data dependency. Code refinements then were made for the subroutines with the poorer computational efficiency.

After the study, the central processor unit (CPU) time for executing a 48-hour model forecast has been decreased from 13,300 seconds to 9,700 seconds. In the mean time, the parallelization has been increased from the averaged using 2.2 CPU to 3.5 CPU.