行政院國家科學委員會專題研究計畫　成果報告

在即時系統中探討具有容錯能力的動態工作排程法

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-009-061-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：國立交通大學資訊工程學系

計畫主持人：陳正

計畫參與人員：李宜軒，張明鈿，陳明志，陳建維，許順閔，柳文斌，李嘉淳，楊偉帆

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 11 月 15 日

行政院國家科學委員會專題研究計畫成果報告

在即時系統中探討有容錯能力的動態工作排程法

A Study of Fault-tolerant Dynamic Task Scheduling Techniques for Real-time System

計畫編號：NSC 92-2213-E009-061
執行期限：92 年 8 月 1 日至 93 年 7 月 31 日

主持人：陳正　　國立交通大學資訊工程學系
計畫參與人員：李宜軒、張明鈿、陳明志、陳建維、許順閔、柳文斌、李嘉淳、楊偉帆
國立交通大學資訊工程學系

一、中英文摘要

在即時系統及多處理機架構下，由於處理器可能發生錯誤的情形，必須使用具有容錯能力的動態工作排程法。現有的方法大部分針對同質性多處理機設計，因此在本年度的計畫中，我們主要將現有方法做延伸改良，使其適用於異質性多處理機系統。同時我們也實作對應的模擬評估環境，評估新方法的執行效能。根據模擬和分析，新方法不僅在效能上符合預期，在執行效率上也不會有太大的負擔，確實可適用於動態排程。

關鍵詞：*異質性多處理機、即時系統、工作排程、動態排程、容錯能力*

Abstract

In real-time system and heterogeneous multiprocessor architecture, since processor may fail unexpectedly, fault-tolerant dynamic task scheduling techniques are necessary and important. Unfortunately, most of existed related techniques are only designed for homogeneous multiprocessor architecture. In this project, we extend existed techniques and propose some fault-tolerant dynamic task scheduling algorithms for real-time system and heterogeneous multiprocessor. Additionally, we implement corresponding simulation environments to evaluate our algorithms. According to simulation results, all proposed algorithms are quite effective and efficient, which is suitable for dynamic scheduling in real-time system.

Keywords: *Heterogeneous Multiprocessor, Real-time System, Task Scheduling, Dynamic Scheduling, Fault-tolerance*

二、計畫緣由與目的

隨著硬體製程的進步，將多顆處理器實作在單顆晶片上的技術已漸趨成熟，間接帶動了多處理機系統的發展 [1]。基於多處理機系統強大的運算能力及可靠度 (reliability)，許多必須即時 (real-time) 完成的應用程式就可以採用此架構執行，以降低所需的執行時間 [2-4]。

即時系統有執行時間的限制，其正確性 (correctness) 的判斷不只依據應用程式的執行結果，還與結果產生的時間有關 [2, 5-6]。即時系統依應用程式性質可分為硬性、韌性及軟性 (hard/firm/soft real-time system) 三種類型，其中硬性即時系統最為嚴謹，它執行的所有工作都必須在期限 (deadline) 內完成，若無法完成則必須盡早退回 (reject)，否則會有災難性 (catastrophic) 的影響 [2, 5-6]。既然如此，容錯能力 (fault-tolerance) 便成為此系統的重要探討議題之一 [2, 4-6]。

處理器發生錯誤的情形一般分成永恆

(permanent) 和短暫 (transient) 二種，前者必須實際替換損壞的硬體才能解決，後者可在一段時間後自行復原，而實際上大部分的錯誤均屬於後者 [7]。在多處理機架構下，容錯能力可由增加備用處理器 (spare processor)，或用排程法將工作備多次由不同處理器分別執行等二種方式來達成 [2, 4, 6-7]。但是使用備用處理器缺點不少，因此目前具容錯能力的即時系統多採用工作備份執行的模式。

根據我們收集的資料，現有用於即時系統具有容錯能力的動態工作排程法，大部份針對同質性多處理機架構設計，甚少考慮異質性多處理機架構 [2-6]。本實驗室之前曾經在異質性多處理機架構上做過工作排程的相關研究，因此在本年度的計畫中，我們將現有方法做延伸改良，在即時系統及異質性多處理機架構下，設計具有容錯能力的動態工作排程法。

除了工作排程法的設計分析，我們也實作對應的模擬評估環境，配合二套工作產生器 (task generator)，測試評估新提出的方法。有關我們提出的方法及測試結果，將分別敘述如下。

三、結果與討論

在介紹我們提出的方法之前，先說明動態工作排程法大致的執行步驟。我們採用集中排程 (centralized scheduling) 的架構，意即有一個排程器 (scheduler) 負責接收動態產生的工作，並將工作分配給其它處理器執行。由於工作會不定時到來，排程器可以先將工作儲存再週期性 (periodic) 排程，或是當有工作到來時隨機排程。每個工作均包含以下三個屬性 (attribute)：到達時間 (ready time)、在不同處理器的執行時間 (computation time) 及執行期限 (deadline)，並假設所有工作彼此獨立 (independent) 沒有資料相依性 (data dependence)。若是排程器發現某工作不能在其執行期限內順利完成，便會將該工作退回 (reject)，避免錯誤發生。

在實際狀況下，由於處理器發生錯誤的機率不高，因此大部分的容錯模組 (fault model) 均假設同一時間只可能有一個處理器損壞，且在下個處理器發生錯誤之前有足夠時間做處理 [2-6]。在本計畫中我們延用這個假設並選用 PB 容錯模組，由排程器將每個工作複製成 primary 和 backup 二個備份，分配給不同處理器執行；既然二個處理器不會同時發生錯誤，primary 和 backup 二個備份中至少有一個可以正確執行，進而達到容錯排程的目的。

我們一共提出三個具容錯能力的動態工作排程法。第一個方法名為 Heterogeneous Distance Myopic Algorithm (HDMA)，是將原本用於同質性多處理機的 Distance Myopic Algorithm [2]，延伸使用於異質性多處理機架構。HDMA (DMA) 屬於序列排程 (list scheduling)，並加入 feasibility check window 及 strongly feasible 的關念，使其具有 look-ahead 的功能。Feasibility check window 的大小可由使用者設定，代表 look-ahead 深度；而一個部分排程若稱為 strongly feasible，則表示它加入任何一個未排程的工作備份都可以找到 feasible 排程。HDMA (DMA) 的執行過程可分為三個步驟：首先將所有工作備份依據輸入的 distance 值排成單一佇列，再將工作佇列前數個備份放入 feasibility check window，最後使用整合型經驗函式 (integrated heuristic function) 從 feasibility check window 選擇最佳工作備份加入排程，同時將工作佇列前端的備份移入 feasibility check window，檢查目前部分排程對所有 feasibility check window 中的工作備份是否為 strongly feasible。此時若不能產生 strongly feasible 排程，則會以整合型經驗函式次佳的工作備份取代原來的最佳備份，看是否能使結果為 strongly feasible。若是遲遲無法找到適合的工作備份加入，才會退回某個工作讓排程動作持續進行。HDMA 基本上承襲 DMA 的執行步驟，不同之處在於整合型經驗函式的定義。HDMA 為了考慮一個工作在不同處理器上執行時間的不同，以「最晚完成時間」取代原來的「最早開始時間」來做選擇。根

據我們的測試評估，HDMA 的執行效能優於其他針對異質性多處理機架構設計的方法；同時由於 HDMA 只改變了經驗函式的計算方式，在執行效率上與 DMA 相較並不會增加額外的負擔，確實也是個適用於即時系統的動態工作排程法。HDMA 的相關研究成果已發表於會議論文 [8]。

第二個提出的方法名為 Fault-Tolerant Myopic Algorithm (FTMA)，改良自之前的 HDMA，同樣針對異質性多處理機架構而設計。在測試 HDMA 時我們發現，由於 HDMA 將所有 primary 和 backup 備份全部排在同一個工作佇列，每排完一個備份之後便將佇列最前端的工作備份移進 feasibility check window，因此會出現一個工作的 primary 和 backup 二個備份同時出現在 feasibility check window 中的情形。為了避免 backup 備份不必要的執行，大部分採用 PB 容錯模組的排程法都會令 backup 備份，必須待對應的 primary 備份實際排入排程之後才能加入排程，HDMA (DMA) 亦有同樣的限制。由此可知，即使二個備份同時存在 feasibility check window，也只有 primary 備份可以經由整合性經驗函式的選擇加入排程。既然 feasibility check window 的大小代表 look-ahead 深度，若是存在不能加入排程的 backup 備份等於是浪費了 feasibility check window 的空間，因此我們設計了 FTMA，改進這個隱性的缺失。FTMA 與 HDMA 的不同之處主要有二個，首先是將 primary 和 backup 備份分開排成二個工作佇列，其次是當有工作備份加入排程之後，在二個工作佇列的第一個備份之間，用整合型經驗函式選擇較佳者移入 feasibility check window。如此一來，不僅不會浪費 feasibility check window 的空間，也不需要輸入 distance 值來排工作佇列，間接解決了 HDMA 中不易選擇 distance 和 feasibility check window 大小配對的問題。根據模擬環境的測試評估，FTMA 的效能明顯優於 HDMA；而且 FTMA 在 feasibility check window 變大時效能並未明顯提升，意即使用 FTMA 時 feasibility check window 不必太大，便可在較低的執行複雜度之下，達到不錯的效能。至於 FTMA 的執行效率也與 HDMA 相似，同樣是個適用於即時系統的動態工作排程法。FTMA 的研究成果業已發表於會議論文 [8]。

我們提出的第三個方法是 Density first with minimum Non-overlap scheduling Algorithm (DNA)，主要定義一個 density 數值當成經驗函式決定工作加入排程的優先順序，以及名為 minimum non-overlap 的機制來做 backup 工作備份的排程。DNA 也屬於序列排程，執行過程可分成三個步驟：依經驗函式選擇工作，將該工作的 primary 和 backup 二個備份加入排程，並更新未排程工作的經驗函式值。在選擇工作方面，大部分方法均使用工作的執行期限當經驗函式，或是如同 HDMA (DMA) 般使用執行期限和最晚完成時間 (最早開始時間) 組成整合型經驗函式；但我們認為，若是一個工作從目前時間到其執行期限之間「剩餘的可執行時間」越短，它應該越優先加入排程，才能有效降低整體工作被退回的機率。因此在 DNA 中，我們根據這個想法定義 density 值，並選擇 density 值最大的工作加入排程。接著在工作排程方面，不同於 HDMA (DMA) 將一個工作的 primary 和 backup 備份當成二個獨立的工作來處理，DNA 選擇同時排程一個工作的二個備份。至於排程的原則，primary 備份延用常見的 ASAP，backup 備份則是另外設計 minimum non-overlap 機制。現有使用 PB 容錯模組的方法，為了提高整體排程成功率，讓 backup 備份有條件重疊是常見的方式；而我們設計的 minimum non-overlap 機制則是令 backup 備份排在「產生最少 non-overlap 時段」的處理器，進一步降低 backup 備份佔用的排程時段。在二個備份都順利排程之後，重新計算未排程工作的 density 值，以便進行下一次的工作選擇。根據對應模擬評估環境的測試，DNA 雖然沒有 look-ahead 的功能，但其執行效能仍較 HDMA 及 FTMA 為佳 (圖 1-8)；也因為不用考慮產生排程是否為 strongly feasible，DNA 的執行效率明顯低於 HDMA 及 FTMA，適合用於需要動態排程的即時系統。此部分相關研究成果可參考 [9]。

四、計畫結果自評

在本年度的計畫中，我們針對即時系統及異質性多處理機架構，在考慮處理器可能發生錯誤的情形之下，提出三個具容錯能力的動態工作排程法，並實作對應的模擬評估環境，配合二套工作產生器做測試評估。總體而言，本計畫大致達成以下幾點目標：

1. 提出 HDMA, 將原本針對同質性多處理機架構設計的動態工作排程法 Distance Myopic Algorithm，延伸使用於異質性多處理機架構。
2. 提出 FTMA，改進 HDMA (DMA) 設計上的缺失，在不增加執行複雜度的情形下，達到更佳的執行效能。
3. 提出 DNA，根據提高整體排程成功率的目標，設計新的經驗函式以及 backup 工作備份排程機制，用更低的執行複雜度，得到比 HDMA 和 FTMA 更佳的執行效能。
4. 實作以上三個排程法對應的模擬評估環境，配合二套工作產生器，証明提出的方法均能達到預期的效能。

由以上幾點可知，本計畫確實能將即時系統及異質性多處理機架構上的某些排程議題做詳細深入的探討，改良既有方法的效能，嘗試提出新的方法，並設計實作模擬評估環境加以測試評估。整體看來，我們提出的方法無論在效能效率上都能達到預期的結果，有效降低即時工作被退回的機率。這些研究成果，有部分已發表於會議論文，後續發展及其他相關排程議題我們也將持續研究。

五、參考文獻

[1] Peng Yang, Chun Wong, Paul Marchal, Francky Catthoor, Dirk Desmet, Diederik Verkest, and Rudy Lauwereins, “Energy-aware Runtime Scheduling for Embedded-multiprocessor SOCs”, *IEEE Design & Test of Computers*, Vol. 18, Issue 5, pp. 46-58, Sep.-Oct. 2001.

[2] G. Manimaran and C. Siva Ram Murthy, “A Fault-tolerant Dynamic Scheduling Algorithm for Multiprocessor Real-time Systems and Its Analysis”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 9, No. 11, pp. 1137-1152, Nov. 1998.

[3] G. Manimaran and C. Siva Ram Murthy, “Dynamic Scheduling of Parallelizable Tasks and Resource Reclaiming in Real-time Multiprocessor Systms”, *Proc. 4th International Conference on High Performance Computing*, pp. 206-211, Dec. 1997.

[4] Tatsuhiro Tsuchiya, Yoshiaki Kakuda, and Tohru Kikuno, “A New Fault-tolerant Scheduling Technique for Real-time Multiprocessor Systems”, *Proc. of 2nd International Workshop on Real-time Computing Systems and Applications*, pp. 197-202, Oct. 1995.

[5] Xiao Win, Hong Jiang, and David R. Swanson, “An Efficient Fault-tolerant Scheduling Algorithm for Real-time Tasks with Precedence Constraints in Heterogeneous Systems”, *Proc. of International Conference on Parallel Processing*, pp. 360-368, 2002.

[6] Sunondo Ghosh, Rami Melhem, and Daniel Mosse, “Fault-tolerance Through Scheduling of Aperiodic Tasks in Hard Real-time Multiprocessor Systems”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 8, No. 3, pp. 272-284, March 1997.

[7] Seong Woo Kwak and Byung Kook Kim, “Task-scheduling Strategies for Reliable TMR Controllers Using Task Grouping and Assignment”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 49, No. 4, pp. 355-362, Dec. 2000.

[8] Yi-Hsuan Lee, Ming-Dien Chang, and Cheng Chen, “Effective Fault-tolerant Scheduling Algorithms for Real-time Tasks on Heterogeneous Systems”, *Proc. of National Computer Symposium*, Dec. 2003.

[9] Ming-Dien Chang, **A Fault-tolerant Dynamic Scheduling Algorithm for Real-time Systems on Heterogeneous Multiprocessor**, Master Thesis, National Chiao-Tung University, June 2004.

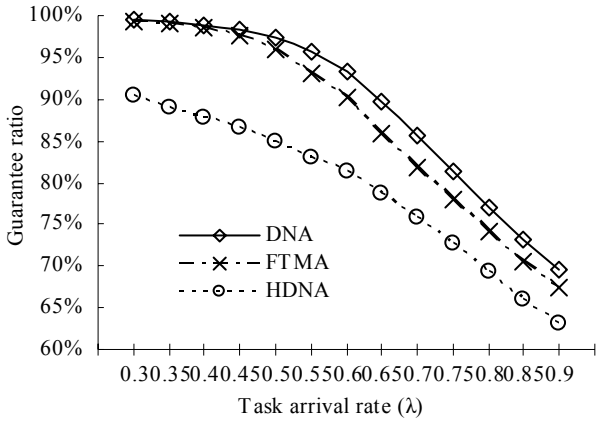

Figure 3. Effect of task load (R = 3, P = 8).

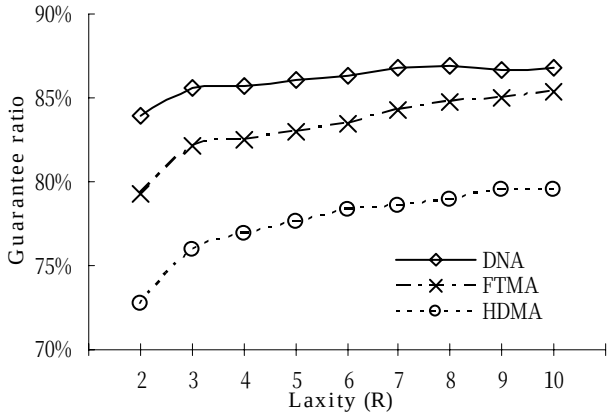

Figure 4. Effect of laxity (λ = 0.7, P = 8).

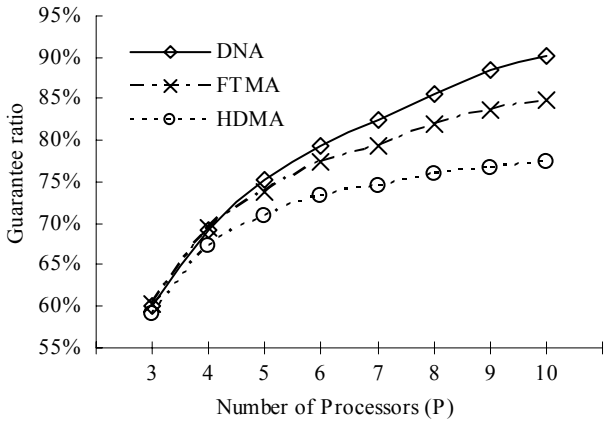

Figure 5. Effect of number of processor (λ = 0.7, R = 8).

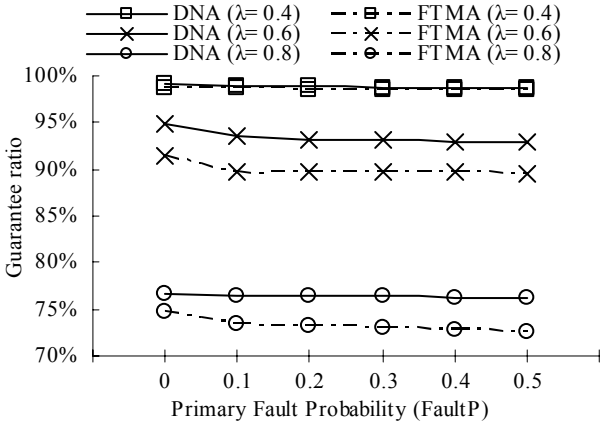

Figure 6. Effect of FaultP with various λ (R = 3, P = 8).

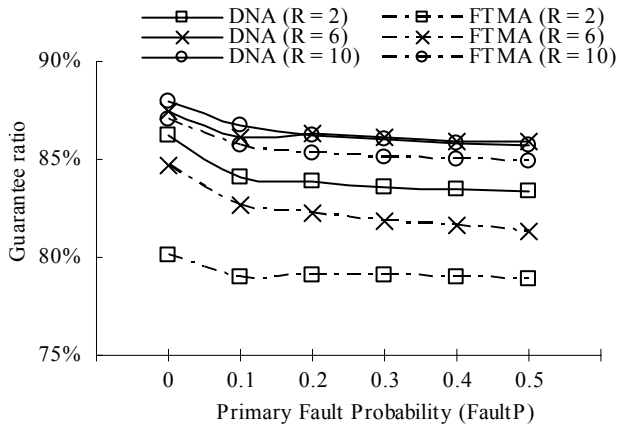

Figure 7. Effect of FaultP with various R (λ = 0.7, P = 8).

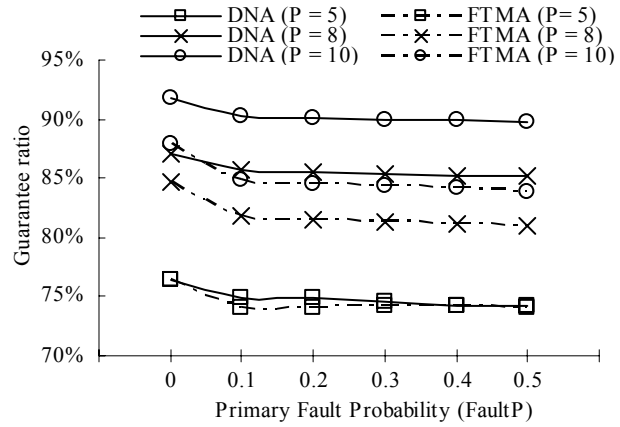

Figure 8. Effect of FaultP with various P (λ = 0.7, R = 3).