

# 行政院國家科學委員會專題研究計畫成果報告

## 數值預測器之研發

### The Design of Advanced Value Predictors

計畫編號：NSC 90-2213-E-032-019

執行期限：90年8月1日至91年7月31日

主持人：莊博任 淡江大學電機系

計畫參與人員：蕭永宗、邱育賢、徐敬岳  
淡江大學電機系

#### 一、中文摘要

微處理器架構在設計上，著重在同時利用數個執行單元執行多道指令，其效率與指令層平行度有高度的相關性。數值預測因此益顯重要。

資料相關性會限制住指令層平行度，若某個指令具有資料相關性，它必須等待前一個指令的執行結果產生之後才能執行。處理器如果能事前先預測結果，就可以同時處理多個指令，不會造成管線浪費，效能自然隨之提升。

我們的研究重點即在改善數值預測的準確度，以增進處理器的運算效能。經由深入的探討與改進。我們發現區域及迴圈性質為數值的主要特性。由於目前使用的數值預測器普遍存有改進空間，因此我們在合理的成本之下，採用可觀察數值之區域性及迴圈性質的預測器替代之，以求得一更合適有效的預測器架構，使預測準確度獲得提升。

**關鍵詞：**指令層平行處理、數值預測、資料相依、數值之迴圈與區域性、數值軌跡、預測準確度、模擬及成本效能評估。

#### Abstract

Value prediction, a technique to break data dependency, is important in enhancing instruction-level parallelism and processor performance. A new value predictor utilizing both the loop and locality properties of data values is proposed in this research to pursue desirable prediction accuracy at reasonable cost. The proposed value predictor, called the

Dynamic Loop&Locality-based (DLL) predictor, makes predictions by dynamically practicing the loop or locality-based prediction policy according to the state. Aided by some simple designs, the DLL predictor proves to be effective and cost-efficient. A new performance measure, *accuracy improvement per cost* or briefed as the A/C ratio, is introduced in the research to secure more comprehensive experimental evaluation of the DLL and other value predictors. As the results collected from extensive simulation runs indicate, compared with the other value predictors of different features, the proposed DLL predictor produces better A/C ratio in almost all situations due to flexible application of different prediction policies as well as reduced cost.

**Keywords:** Instruction-level parallelism, value prediction, data dependency, loop and locality properties, value history, prediction accuracy, simulation and cost-performance evaluation.

#### 二、計劃緣由與目的

近代的微處理器架構在設計上，著重在同時利用數個執行單元執行多道指令。這種架構的效率，與指令層平行度有高度的相關性。數值預測的重要性，因此益受重視。

一般的處理器指令需要等待先前指令的運算結果產生之後才能執行，使得指令經常耗費在無謂的等待，管線也因此而停滯，此種現象稱為資料相關性。資料相關

性會限制住指令層平行度，使得處理器無法同時處理多個指令。處理器如果能在結果產生之前，事先獲得結果給下一個指令使用的話，就不會造成管線的浪費，效能自然隨之提升。

數值預測是一種藉由預測指令的結果來打破產生數值的指令與消耗數值的指令之間的相關性之技術。其主要功能是藉由記錄過去的歷史資料來預測指令所產生的結果，藉此越過內部指令資料間的依賴關係，使有依賴關係的指令群達到平行處理的效果。

1996 年 last outcome predictor [1]被提出來，它利用簡單的方法捕捉指令重複產生的數值，也開啟了數值預測的研究之門；此後數年間針對 data value 之特性，各種預測器紛紛被提出，預測準確度的研究改進不斷推陳出新。原本用於分支預測的 two-level predictor 也被用於數值預測[2]，它能夠藉由捕捉更複雜的重覆 patterns 增進準確度。隨後各式截取多種預測器優點的混合型預測器，也被廣泛討論[3]。

由於目前普遍使用的各種數值預測器都存在改進空間，對數值預測法做了極為深入的探討後，我們研究如何改善數值預測的準確度，以增進處理器的運算效能，進而提昇整體系統效能。

在實際做法上，我們針對數值的兩項特性來改進數值預測器：

1. 區域 (locality) 特性：last outcome scheme [1]是第一個針對此特性的預測器，架構簡單且成本不高，但準確度卻不能夠符合一個高效能處理器所需；FCM[4,5]和 two-level predictor[2]也是針對此一特性所設計的預測器，它們都有雙層次的設計且預測相當準確，但其硬體成本較高且運作複雜為缺點。因此我們設計出一折衷的預測方法，使其兼顧成本而且維持一定的準確度。
2. 迴圈 (loop) 特性：我們知道程式中經常有迴圈的出現，因此數值也具有迴圈性質。模擬數據顯示 stride predictor [6,7]的預測正確率很高，表示數值中迴圈佔有一定的比率，但是一般的預測器都沒有特別針對迴圈的性質加以設計，因此我們在預測器中加入了偵測迴圈性質

的機制，使其能有效預測出迴圈的出現，以提升預測準確度。

### 三、結果與討論

我們所提出的方法，包含一個新設計的預測器 Dynamic Loop&Locality-based predictor (簡稱 DLL 預測器)，以及一個新的分發策略 (簡稱 LRU info Dispatch)。

#### (一) DLL 預測器

DLL 預測器的運作方式如下：指令以 program counter 索引到 entry，比較 tag 相同後，檢查 state 欄位，若為 steady，表示前兩次出現的 data values 有等差的現象，則預測下一個值為 value+stride；若 state 欄位是 locality，表示之前的 data values 沒有等差的現象，則找出 counters 欄位中哪一個 counter 的值最大，並檢查此 counter 值是否有超過 threshold 值。若有超過 threshold 值，就以此 counter 對應的 data value 為預測值；若沒有超過 threshold 則不預測。

當 outcome 可得時，更新分為兩部份，針對迴圈性質部份的 Stride、State 欄位需要更新：stride 欄位存放 outcome 與 value 的差值，接著再將 outcome 存入 value 欄位，而 State 欄位的改變是以目前的 stride 和前次的 stride 是否相等為改變的依據，相等時 state 改變成 loop，不相等時則 State 欄位改為 locality。更新的另一部份包括針對 locality 性質的 LRU info、Data Values、Counters 欄位：由 LRU info 可知 data values 中何者最舊，最舊者用 outcome 替換掉。Counters 方面，每當 outcome 可得時，DLL 所儲存的 unique values 與 outcome 比對，比對相等者的計數值加 3(如果加 3 以後超過飽和值，則以飽和值計)，比對不相等者的計數值減 1(若它們都非零的話)。

#### (二)LRU info Dispatch

假設每個 entry 存放 4 個 unique values，這 4 個 values 分別用 index 表示，即 0, 1, 2, 3。LRU info 欄位有 4 個 fields，每個 field 佔 2 個 bits：

LRU			MRU
-----	--	--	-----

LRU 放的是第一個要被替代的 index，MRU 是最後才被替代的 index。一開始，LRU info 及 data values 欄位是空的，當 outcome 產生後，這筆資料填入 data values 欄位的第一個：

Index :	0	1	2	3
Data values :	V1	Null	Null	Null

LRU info 欄位做以下的安排：

1	2	3	0
LRU			MRU

這樣可以達到當第 2, 3, 4 個 unique value 出現的話，index 1 的值最先被替代，再來是 index 2 和 3 的值被替代。最後被替代的是 V1，因為它是 MRU。

假設 V5 出現前，Data values 和 LRU info 如下：

Index :	0	1	2	3
Data values :	V1	V2	V3	V4

LRU info :	2	3	1	0
LRU			MRU	

第 5 個 unique value (V5) 出現後，index 2 要被替代，替代完畢後 index 2 變為 MRU，其餘的 index 左移：

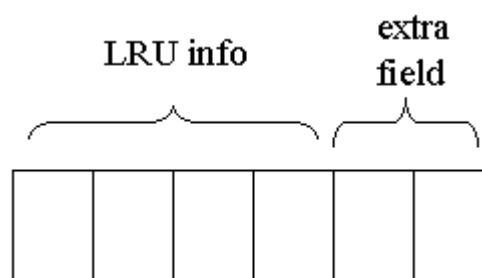
Index :	0	1	2	3
Data values :	V1	V2	V5	V4

LRU info 欄位做以下的安排：

3	1	0	2
LRU			MRU

由以上可知，LRU info 也是一種 pattern，但是存放 4 個 values 時，LRU info 也存放 4 個 index (一個 index 需 2bits)，而這樣只有 8 個 bits，只能索引到  $2^8=256$  個 entries (256 個太少，造成的干擾效應會很大)。與 two-level VHP 的 12 個 bits (可索引到  $2^{12}=4096$  個 entries) 有不小的差距。

所以增加一個叫 extra\_field 的欄位，存放 2 個 index (4 bits)。在 LRU info 更新前，先將 extra\_field 的資料左移，再將 LRU index 存放到 extra\_field 的右邊那格 (即低位元欄位)。這樣的設計也是存 6 個 index，共 12 bits 與 two level VHP 一樣。LRU info 分發的架構如下圖。



LRU info 分發的架構圖

針對所提出的方法，我們做了模擬評估，評估的標準除了預測正確率外，還加入成本的考量。綜合各項模擬結果。我們發現能夠兼顧 data value 的 locality 及 loop 特性的預測器才能在各項 benchmarks 中有較高的正確率，如 FCM、hybrid 及 DLL 預測器。這時成本的考量就顯得特別重要，由於 FCM 使用非常多的硬體成本記錄歷史資料，雖然正確率最高，但在考量成本的情況下顯得不盡理想。Hybrid predictor

在針對 locality 特性的花費的成本較 DLL 預測器為多，但是所能提升的效能有限，因此總體來看，效能成本比以 DLL 預測器較高。

#### 四、計畫成果自評

以往在研究數值預測問題時，大多數的文獻都將著眼點放在正確率的提升，因而忽略了成本對此問題的影響。在考量成本的情況下，我們定義了一個評比預測器的公式，藉由此公式可以反映出預測器與所達到的正確率是否合乎效能成本比，而不是一味的使用複雜的方式及增加成本來達到提高正確率的目的。

模擬結果顯示，我們所提出的 DLL 預測器，可以在合理的硬體成本下，達到水準以上的預測正確率，並且有著優異的效能成本比。而我們設計 LRU info 分發的本意在充分利用現有資源，達到減低分發時所需的成本，但我們同時發現 LRU info 分發也可適用於其它使用 LRU policy 更新 data values 的預測器。將 DLL 預測器和 LRU info 分發技術結合運用後，我們也發現符合預期想法的結果，也就是在合理的分發成本下，有效提高了 DLL 預測器內的 locality-based predictor 之預測正確率。

本研究計劃所預計達成的目標都能圓滿完成，此一研究成果並已整理成篇，投付國際研討會審理中。今後的預期目標與研究方向在於發展出更符合效能成本比的 locality-based 和 loop-based predictor，並找出一個更有效的結合式預測器的預測方式，進一步改進並提昇效能成本比和預測準確度。

#### 五、參考文獻

- [1] M. H. Lipasti and J. P. Shen, "Exceeding the Dataflow Limit via Value Prediction," *Proc. 29th Annual IEEE/ACM Int'l Symp. on Microarchitecture*, Nov. 1996, pp. 226-237.
- [2] T.-Y. Yeh and Y. N. Patt, "Alternative Implementations of Two-Level Adaptive Branch Prediction," *Proc. 19th Int'l Symp. on Computer Architecture*, May 1992, pp. 124-134.
- [3] K. Wang and M. Franklin, "Highly Accurate Data Value Prediction Using Hybrid Predictors," *Proc. 30th Annual IEEE/ACM Int'l Symp. on Microarchitecture*, Nov. 1997, pp. 281-290.
- [4] Y. Sazeides and J. E. Smith, "The Predictability of Data Values," *Proc. 30th Annual IEEE/ACM Int'l Symp. on Microarchitecture*, Nov. 1997, pp. 248-258.
- [5] B. Goeman, H. Vandierendonck, and K. DeBoschere, "Differential FCM: Increasing Value Prediction Accuracy by Improving Table Usage Efficiency," *Proc. 7th Int'l Symp. on High-Performance Computer Architecture*, Feb. 2001, pp. 207-216.
- [6] F. Dahlgren and P. Stenstrom, "Evaluation of Hardware-based Stride and Sequential Prefetching in Shared-memory Multiprocessors," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 7, No. 4, pp. 385-398, Apr. 1996.
- [7] T. Chen and J. L. Baer, "Effective Hardware-Based Data Prefetching for High-Performance Processors," *IEEE Trans. on Computers*, Vol. 44, No. 5, pp. 609-623, May 1995.