

行政院國家科學委員會專題研究計畫 期中進度報告

子計畫二：能監控服務品質之轉碼技術研發(1/3)

計畫類別：整合型計畫

計畫編號：NSC93-2213-E-002-126-

執行期間：93年08月01日至94年09月30日

執行單位：國立臺灣大學電信工程學研究所

計畫主持人：陳銘憲

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 94 年 11 月 4 日

中文摘要：

本計畫為整合型計畫「多媒體共享之網路服務平台」之子計畫。其名稱為「能監控服務品質之轉碼伺服器技術研發」。在第一年度中，我們參考了有線及無線網路流量的模型，以及傳統轉碼伺服器的技術，而設計出適合處理多媒體文件的轉碼伺服器的架構，而根據此一架構，我們並將此一轉碼伺服器處理多媒體文件的機制，轉化為傳統演算法中，最具複雜度之困難問題之一，即所謂之「0-1 背包問題」(0-1 Knapsack Problem)。此一問題，在演算法中，最困難之處在於，用背包在裝物件時，該物件具備「不可分割」的特性，因此，為了求出最佳解，我們設計了一套利用「動態規劃法」(Dynamic Programming)概念的演算法，藉由這套演算法，我們可以決定出，轉碼伺服器在處理某個特定的多媒體文件時，該如何進行「轉碼」，或是將其存放在伺服器中的「快取磁碟機」中，使得無線網路的用戶端，在從轉碼伺服器要求多媒體文件時，能讓平均等候時間降至最低，而讓用戶端享有更好的服務品質。

關鍵字：多媒體文件，轉碼伺服器，背包問題，動態規劃

英文摘要：

This report belongs to the sub-project “A QoS-Aware Transcoding Proxy using On-Demand Broadcasting”, which is under the integrated project

“Design of Service Platform for Sharing Multimedia Documents”. In the first year, we have designed an effective architecture of a transcoding proxy for multimedia documents according to the network flow model in the wired and wireless environment. It is noted that the transcoding technique is capable of trading of object fidelity for size. Therefore, the transcoding proxy for multimedia objects can utilize the network bandwidth very well. According this architecture, we also transform the problem of caching multimedia documents to a “0-1 Knapsack Problem”, which is a well-known NP-Complete problem. According to the characteristics of the problem, we have designed an effective algorithm by using the concept of “Dynamic Programming” so that the problem can be solved in polynomial time. Therefore, a transcoding proxy for multimedia object can perform “transcoding” or “caching” mechanism so as to enhance the quality of mobile users by reducing the average waiting time to the minimum.

Keywords: Multimedia documents, Transcoding Proxy, 0-1 Knapsack Problem, Dynamic Programming

前言：

隨著行動通訊技術的進步，無線網路領域中的諸多議題，對業界及學界而言，已逐漸凌駕於傳統之有線網路，而成為現今電信技術之研究主流。透過無線網路，人們可利用手機、

PDA 或是手提電腦，作雙向資料的傳輸，包括下載交通，股市，及氣象等重要資訊。而近年來所提出的 GPRS，3G 等行動通訊的標準，讓使用者在使用無線網路時，享有更大的頻寬。也因此，傳統純文字資料的下載，已不敷使用者的需求，而逐漸被新興之多媒體資料所取代。在講求效率的今日生活中，如何提供更快速，更便捷的多媒體傳輸服務，已成為爭相研究的課題。

所謂多媒體文件，包括了圖片、聲音以及影像。一般而言，多媒體文件，和傳統純文字文件相比，即使透過相關壓縮技術之處理，仍然具有較大之資料量。例如：以電腦位元組 Bytes 作為計算單位，一份普通的純文字文件，約為 101KB；一張以 JPEG 作壓縮，大小約 1024x768 的彩色圖片約為 102KB，一段以 MPEG 作壓縮，長度為一分鐘的影片，其資料量則高達 104KB。

而在無線網路的環境中，使用者所使用的網路平台包括了手機、PDA，以及筆記型電腦等。不同種類的網路平台，對多媒體文件的品質要求不盡相同。以手機使用者為例，市面上之彩色手機，其解析度為 600x480，並用 16 位元來表示一個點的顏色；在另外一方面，以手提電腦的使用者為例，其解析度可高達 1024x768，並可用 32 位元來表示一個像點。當兩種不同平台之使用者欲接收同一筆影像文件時，讓 1024x768x32 的影像同時傳給手機平台及手提電腦之平台，無疑會造成網路頻寬之浪費。

有鑒於不同的使用者平台，對多媒體文件的品質要求，皆不相同，如

何相多媒體文件做轉碼，儲存，以及傳送給使用者，使得網路流量，以及使用者的等候時間，皆能降至最低，成為一個熱門的研究主題。由於轉碼技術已日趨成熟，現有之 MPEG4, H. 264, JPEG 2000 等規格，皆能將多媒體文件作合適的轉碼，因此，本計畫「能監控服務品質之轉碼伺服器技術研發」的研究重點，便在於儲存技術以及傳送技術的研發以及整合。

研究目的：

本計畫「能監控服務品質之轉碼伺服器技術研發」第一年之研究方向主要在設計具有較佳效能之轉碼伺服器架構。由於轉碼伺服器架設於使用者和遠端伺服器之間，當使用者想存取遠端的多媒體文件時，如果轉碼伺服器存放有使用者所要求之文件，直接將該文件回傳給使用者，便能有效地降低使用者之等候時間，但轉碼伺服器具備有限的磁碟空間，只有部份的文件能被存放在快取磁碟機中。同時，多媒體文件可藉由轉碼技術而轉成不同的版本，傳統的快取置換技術，便無法使用在轉碼伺服器中。此一限制，使得設計出合適的快取置換方式，成為一熱門之研究主題。

文獻探討：

在本篇結果報告中，我們研究的重點在於提出一個快取置換的機制，使得無線網路中的使用者，在存取轉碼伺服器中的多媒體文件時，能夠享有更高品質的服務，亦即使用者存取所花費之時間，可以降至最低。所謂「快取置換」(Cache Replacement)，不論在有線或是無線網路中，都是一個

值得深入研究的主题 [3]，在傳統的 World Wide Web 環境下[7]，有於代理伺服器 (Proxy) 的磁碟空間有限，因此，其代理伺服器設計一套快取置換的機制，不斷更新快取磁碟機中的內容，使得大部份使用者所要求的資料，皆能存放於快取磁碟機中，而讓使用者的等候時間降至最低。

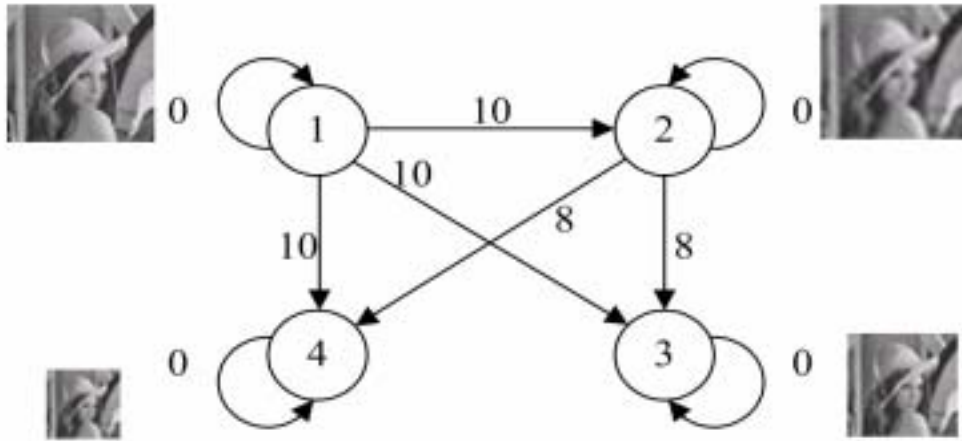
一般而言，在傳統的 Web Proxy 的環境下，常用的快取置換機制可分為 LFU [4] (Lest Frequently Used)，以及 LRU [5][6] (Least Recently Used)。所謂 LFU，其核心概念在於，快取磁碟所存放的，為存取機率較高的資料。至於 LRU，其核心概念則在於，快取磁碟所存放的，為近期常被存取的資料。然而，在轉碼伺服器的環境下，由於代理伺服器具備有將多媒體文件轉碼之特殊功能，一份多媒體文件，便可能有不同種型式之版本，每份版本的檔案大小以及解析度皆不相同。因此，決定一多媒體文件是否該儲存於快取磁碟中，以及該以何種型式存放，便成為一個值得深入考慮的問題。在[1]這份論文中，作者便首度提出了轉碼伺服器的快取置換機制。作者提出了兩種方法：Coverage-Based 以及 Demand-Based Replacement。所謂 Coverage-Based Replacement，代表著轉碼伺服器會將最原始的多媒體文件存於快取磁碟中，當使用者要求某一份多媒體資料時，如果該份文件存放在轉碼伺服器的快取磁碟中，即使版本不相同，轉碼伺服器仍會執行轉碼的動作，將原始版本轉成使用者所要求之版本。至於 Demand-Based Replacement，指的則是轉碼伺服器會存於使用者所要求過的特定版本的資

料，和 Coverage-Based Replacement 相比，此一方法最大的好處在於，由於不用儲存原始版本，因此，轉碼伺服器可以存放較多筆資料。而在[2]這份論文中，作者則是採用「利益」的概念，計算出存放哪一份多媒體文件的哪一份版本，可以享有較大的利益，而轉碼伺服器便根據此一利益法則，來存放多媒體文件，使得整體利益達到最大。

和以上所提及的文獻相比，本份報告之重點，在於改進轉碼伺服器之效能，使得用戶能享用更好的服務品質。以上文獻中，以[2]這篇論文所提出的方法效果最佳。然而[2]這篇論文所採用的，是一簡單的「貪婪演算法」(Greedy Algorithm)，我們這份報告中，則設計出最佳解，使得整體效能達到最高。

研究方法：

首先，為了表達方便，我們定義了如下的記號。假設每一份多媒體文件可以透過轉碼伺服器的技術，轉成 n 種版本，因此我們用 o_{ij} 來表示某一份多媒體文件的某一特定版本，其中 i 用來識別文件，而 j 則用來識別版本，而 $o_{i,0}$ 則代表的是原始版本，也是畫質或解析度最清晰的版本。舉例而言， $o_{1,2}$ 則代表了第一份多媒體文件的第二個版本。而每一份多媒體文件 o_{ij} ，具有兩個屬性，被要求頻率 (r_{ij}) 以及物件大小 (s_{ij})。根據此一轉碼伺服器的特性，我們對一份多媒體文件 i ，可以定義出「轉碼示意圖」(Weighted Transcoding Graph) G_i ，來表示不同版本之間轉碼的相互關係。一個轉碼示意圖的例子如圖一所示：



圖一 轉碼示意圖

而在此一轉碼示意圖中，我們可以用 $V[G_i]$ ，來表示圖中各點，而用 $E[G_i]$ 來表示圖中各邊。而每一個邊，都有一個相對應的值 $w_i(u,v)$ ，來代表從第 u 個版本轉碼到第 v 個版本所需要的時間。由以上之定義，我們可以定義出將某個版本的多媒體文件，存放在快取磁碟中，可以獲得的「個別利益」(Singular Profit)如下：

$$PF(o_{i,j}) = \sum_{(j,x) \in E[G_i]} r_{i,x} (d_i + w_i(1,x) - w_i(j,x))$$

其中 d_i 代表的是將原始版本的多媒體文件從遠端伺服器下載到代理伺服器中，所需要的時間。

同樣的，我們亦可以定義出，將某一多媒體文件中其中幾個特定的版本，存放於快取磁碟中時，所獲得的「整體利益」如下：

$$PF(o_{i,j_1}, o_{i,j_2}, \dots, o_{i,j_k}) = \sum_{v \in [G_i']} \sum_{(v,x) \in E[G_i']} r_{i,x} (d_i + w_i(1,x) - w_i(v,x))$$

由個別利益和整體利益的關係，我們可定義出所謂「邊際利益」，即對

一份多媒體文件而言，在某些版本已經存放在快取磁碟機中時，新加入一個版本，可以增加的利益，邊際利益的定義如下：

$$PF(o_{i,j} | o_{i,j_1}, \dots, o_{i,j_2}) = PF(o_{i,j}, o_{i,j_1}, o_{i,j_2}, \dots, o_{i,j_k}) - PF(o_{i,j_1}, o_{i,j_2}, \dots, o_{i,j_k})$$

因此，對每一份多媒體文件而言，除了「被使用頻率」以及「體積」外，我們可以定義出第三個屬性：「利益」(p_{ij})。其定義如下，當有其他的版本存放在快取磁碟機中時，則 p_{ij} 代表的即為「邊際利益」。相反的，當沒有其他的版本存放在快取磁碟機中時，則 p_{ij} 代表的即為「個別利益」。

假設所有的資料皆存放在一個名為 D 的資料庫中，我們可以定義出「快取候選集合」(Caching Candidate Set) D_H ，來代表在 D 中，具有較高優先權去存放於快取磁碟機中的多媒體文件。同時，我們也定義出最大利益 P_H ，來代表將 D_H 全部存放在快取磁碟機中，所能獲得的利益。因此，我們便將問題轉化為：給定磁碟空間的限制，從資料庫 D 中決定 D_H ，使得 D_H

滿足磁碟空間的限制，且 P_H 可以達到最大值。

觀查此一問題的特性，我們可發現，此一問題可等效成「背包問題」(Knapsack Problem)：給定一些物品，每一物品有不同的價值和重量，而背包只能承受一定的重能，如何挑選，使得背包能裝的物品，具有最大的價值。而本問題中的「利益」，即相當於背包問題中物品的價值，同樣的，本問題的「體積」，則等效於背包問題的重量。而背包問題，可經由「動態規劃」[9]的方式來求得最佳解，於是，我們可以利用動態規劃的概念，求得 D_H ，其演算法如下面圖二所示：

```

Procedure DP
Input: the database  $D$ , and the cache size constraint  $Z_c$ .
Note that  $|D| = m \times n$ , where there are  $m$  kinds of data,
and each data object has  $n$  versions.
Output: the caching candidate set  $D_H$ 
begin
1. Create a two-dimensional  $(m \times n + 1) \times (Z_c + 1)$  array
 $c[][]$  and set each element  $c[i][j]$  to be 0.
2. for  $i:=1$  to  $m$  do
3.   for  $j:=1$  to  $n$  do
4.     for  $z:=1$  to  $Z_c$  do
5.       if  $z_1 \leq z$ 
6.         if  $p_{i,j} + c[(i-1) + n + j - 1][z - z_1]$ 
            $> c[(i-1) + n + j - 1][z]$ 
7.            $c[(i-1) + n + j][z]$ 
            $\leftarrow p_{i,j} + c[(i-1) + n + j - 1][z - z_1]$ 
8.         else
9.            $c[(i-1) + n + j][z]$ 
            $\leftarrow c[(i-1) + n + j - 1][z]$ 
10.        else
11.           $c[(i-1) + n + j][z]$ 
            $\leftarrow c[(i-1) + n + j - 1][z]$ 
12. Determine  $D_H$  by tracing the elements in  $c[][]$ 
end

```

圖二 動態規劃演算法

而根據所求得的 D_H ，我們可以設計出一套演算法，名為「最大利益快取置換演算法」，來進行快取磁碟機中的置換工作。此一演算法，可用下面圖三來表示。

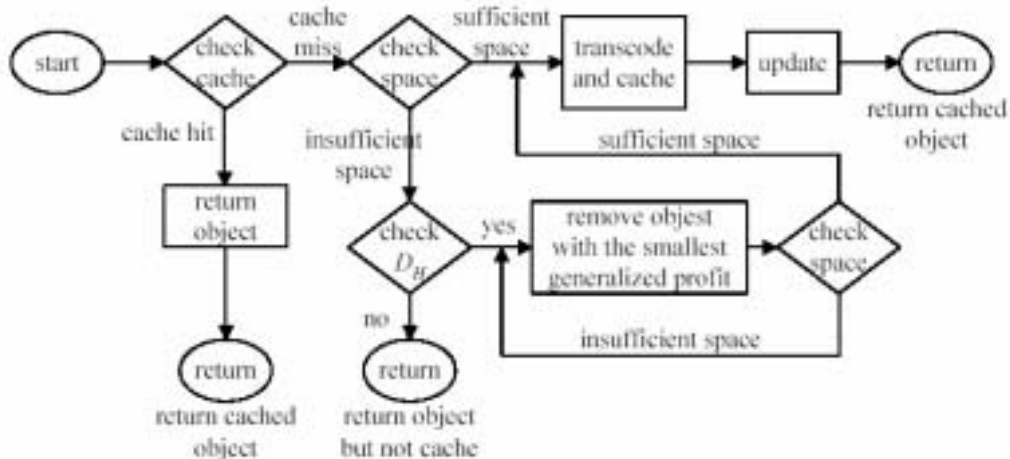
對每一個使用者的要求訊號而

言，轉碼伺服器處理多媒體文件，可分成下面三個方面來討論：

1. 如果轉碼伺服器存放有使用者要求的多媒體文件，且版本相同，則轉碼伺服器會直接將該物件回傳給使用者
2. 如果轉碼伺服器沒有存放有使用者要求的多媒體文件，而且快取磁碟機中仍有足夠的空間，則轉碼伺服器便會透過轉碼，或者是從遠方伺服器端下載該份多媒體文件，將其儲存在快取磁碟機中後，再回傳給使用者。
3. 如果轉碼伺服器沒有存放使用者要求的多媒體文件，而快取磁碟機中已無足夠的空間，則轉碼伺服器將會檢查該份多媒體文件是否屬於 D_H ，如果屬於 D_H ，則將快取磁碟機中的物件依照 $p_{i,j}/s_{i,j}$ 的值，由小到大移出快取磁碟機，直到空間足夠，然後重覆步驟 2；如果不屬於 D_H ，則回傳資料後，轉碼伺服器便不會將此一多媒體文件存入快取磁碟機中。

結果與討論：

在執行本子計畫的第一年中，我們成功地設計出能達成最佳效能之多媒體轉碼伺服器之架構，藉由「快取置換」的機制，使得使用者在等候多媒體文件之時間降至最低，此一系統架構，除了能有助於後兩年之研究進行外，也能使得後兩年的系統研究，能達到更高的效能。



圖三 最大利益快取置換演算法之流程圖

參考文獻：

- [1] V. Cardellini, P. S. Yu, and Y. W. Huang, "Collaborative Proxy System for Distributed Web Content Transcoding" In *Proceedings of ACM International Conference on Information and Knowledge Management*, 2000.
- [2] C. Y. Chang and M. S. Chen, "On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxy." *IEEE Transaction on Parallel and Distributed Systems*, Val. 14, No. 6, 2003.
- [3] S. Podlipnig and L. Boszormenyi, "A Survey of Web Cache Replacement Strategies" *ACM Computing Surveys*, Vol. 35, No. 4, December 2003.
- [4] M. F. Arlitt, L. Cherkasova, J. Dilley, R. J. Friedrich and T. Y. Jin, "Evaluating Content Management Techniques for Web Proxy Caches." *ACM SIGMETRICS Perform. Eval. Rev.* Vol. 27, No. 4, 2000
- [5] C. Y. Chang, T. McGregor and G. Holmes, "The LRU* WWW Proxy Cache Document Replacement Algorithm." In *Proceedings of the Asia Pacific Web Conference*, 1999.
- [6] K. Cheng and Y. Kambayshi, "A Size-Adjusted and Popularity-Aware LRU Replacement Algorithm for Web Caching." In *Proceedings of the 24th International Computer Software and Applications Conference (COMPSAC)*, 2000.
- [7] M. Abrams, C. Standridge, G. Abdulla, S. Williams and E. Fox, "Caching Proxies: Limitations and Potentials," In *Proceedings of Fourth International World Wide Web Conference*, 1995.
- [8] T. C. et al, "Introduction to Algorithms." McCraw Hill

- [9] M. Sniedovich. "Dynamic Programming." M. Dekker, 1992
- [10] C. Aggarwal, J. L. Wolf and P. S. Yu, "Caching on the World Wide Web," *IEEE Transaction on Knowledge and Data Engineering*, Vol. 11, No. 1, 1999
- [11] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Caching and Zipf-Like Distributions: Evidence and Implications," In *Proceedings of IEEE INFOCOM*, 1999
- [12] S. Chandra, C. Ellis, A. Vahdat, "Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding," *IEEE Journal on Selected Areas in Communication*, 2000
- [13] S. Chandra and C. S. Ellis, "JPEG Compression metric as a Quality-Aware Image Trnascoding." In *Proceedings of USENIX Second Symp. Internet Technology and Systems*, 1999