

# 適應性的分散式文件存取系統

## An Adaptive Distributed Document Access System

羅文聰<sup>1</sup> 許瑞愷<sup>2</sup> 陳倫奇<sup>1,\*</sup>

<sup>1</sup>東海大學

資訊工程與科學系

台中市西屯區台中港路三段 181 號

<sup>2</sup>交通大學

資訊科學與工程研究所

新竹市大學路 1001 號

<sup>1</sup>{winston, casper}@thu.edu.tw      <sup>2</sup>rksheu@cis.nctu.edu.tw

<sup>1</sup>Win-Tsung Lo      <sup>2</sup>Ruey-Kai Sheu      <sup>1</sup>Lun-Chi Chen

<sup>1</sup>Department of Information Engineering and Science

Tunghai University

Taichung City, Taiwan

<sup>2</sup>Institute of Computer Science and Engineering

National Chiao Tung University

Hsinchu City, Taiwan

<sup>1</sup>{winston, casper}@thu.edu.tw      <sup>2</sup>rksheu@cis.nctu.edu.tw

*Received 15 May 2006; Revised 21 June 2006 ; Accepted 30 June 2006*

### 摘 要

隨著網路的發達及企業的全球經營方式，檔案資料大部分儲存在不同的區域網路，所以檔案管理系統將面臨檔案的存取效率以及確保取得的檔案內容一致的問題。針對檔案分散儲存與檔案存取的問題，一般解決方法是將檔案集中儲存管理或檔案集中管理後，透過備份或複製檔案系統與資料到其他區域網路。雖然檔案集中管理可以確保檔案一致性，卻造成外部網路的使用者須付出更多時間與網路頻寬來存取檔案資料。而採取異地備援或檔案複製的方法將造成不必要的資料儲存空間與頻寬浪費，且此方法須搭配複雜的資料同步機制。本論文提出一個透過階層式邏輯檔案索引與快取的設計建立於分散式文件管理系統中的檔案存取機制，在現有網路頻寬與架構下，提供更有效率的檔案傳輸與文件管理。

**關鍵詞：**分散式文件存取、階層式索引、適應性檔案管理、檔案快取、集中管理、分散儲存

---

\* 通訊作者

## ABSTRACT

More and more documents are stored dispersedly in geographically distributed branches. People often have to work over networks slower than LANs. How to keep the consistency of document states, and how to guarantee the performance of document accesses are the major problems for most enterprise, especially for the administrator of information management departments. To solve such problems, traditional mechanisms suggest a central server with a large storage containing all documents in it to provide the document access service. To satisfy enterprises' needs, this paper is proposed to provide an efficient and high performance distributed document access system for geographically separated users. With our proposed method, enterprises can upgrade the document management efficiency based on currently existent network topology and architecture. The high performance distributed document management system (DMS) is composed of two major modules, and they are the logic document index management module, and the file server module. Logic document index management module serves the adaptive distributed document access mechanism and meta-data management which is independent of physical document storages, and guarantees the consistency of document states. The file server module manages the mapping between hierarchical logic indices and physical storage directories. Besides, it supplies cache management functions with the same performance as local document access no matter where users are physically located.

**Keywords** : Distributed document access, Hierarchical index, Adaptive document management, Document cache, Control management, Distributed storage.

## 一、前言

現今文件管理系統的檔案資料大部分儲存在不同的區域網路，且文件檔案存放越來越分散，如何做到有效率的管理十分重要。對於此種情況，使用者取得檔案資料的存取效率以及確保使用者取得的檔案內容一致的問題變得更加複雜與繁瑣。

目前企業存取檔案都是透過使用比區域網路(LAN)還要慢的網路環境，即使透過寬頻的網際網路(Internet)，也只能達到 Mbit/sec 的頻寬，何況大部分企業分公司之間幾乎都只有 T1 的頻寬。所以如何讓所以檔案系統使用者能在現有的架構下，還能有效率的存取分散於各地的大量文件或大型檔案，是目前企業應該積極解決的課題[1][2]。

以上所面臨的問題可分為：(一)文件內容一致性問題：由於檔案內容可能不斷持續在更新，使用者無從得知所取得的文件是否為最新版次，及檔案內容是否正確。即使採用異地備援或資料複製機制，資料的同步仍須配合複雜的演算法才能達成。(二)單一文件樹的建置問題：由於文件分散各地，且資料如此龐大，使用者必須記住公司目前所有的文件種類，並且必須瞭解哪些文件分別存放於哪些地方，加上文件檔案不斷的更新。這是很沒有效率的工作方式。(三)文件存取效率問題：使用者必須從美國下載文件至台灣，如果有多位同仁同時下載文件，將會導致原本的網路頻寬不敷使用。在企業內部大量文件需要同時管理的情況下，沒有效率的文件存取機制將會用盡所有網路頻寬，甚至影響到正在運作的公司其他內部系統[2][3]。在原有頻寬早已不足的情況下，任由使用者在伺服器間傳輸大量文件資料，對企業來說，是一個無形的成本支出。

我們將重點放在如何快速有效率的取得遠端資料，尋求更好更佳的解決辦法，在不改變現在低效能的網路頻寬與架構下，系統能自動適應使用者位置，自動調整檔案存取位置的分散式文件管理系統，期望能滿足企業所需的文件資料集中管理、分散儲存的需求。

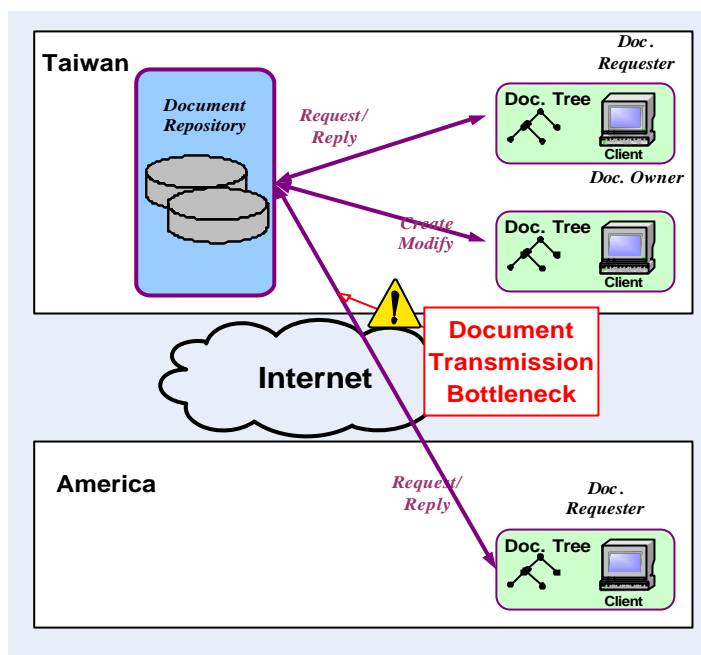
## 二、文獻探討

國內研發單位往往設計很新的系統，嘗試解決企業問題，但往往發生很多實際導入的問題。對於企業內部大量文件分散管理的問題，也是其中一例。為了解決文件內容一致性、單一文件樹的建置、文件存取效率的問題，大部分企業嘗試建立自己的文件管理系統[4]，目前企業使用的管理系統可分為四類：

### 1. 集中管理、集中儲存：

此類的舊型系統大致上是屬於功能比較簡陋系統，或充其量只能稱之為檔案系統，而非文件管理系統，如圖一所示，客戶端透過本身的文件樹(Doc. Tree)發送一個文件請求(Document Request)給文件存儲器(Document Repository)，而文件存儲器提供各個點的文件存取要求，且各個客戶端的文件系統皆相同。此類系統的優點在於能提供所有使用者一致的文件內容，但是缺點是外部網路的使用者都必須透過單一網路通道存取文件，因此，遠端使用者必須忍受較差的檔案存取效率，必須多花遠端存取所需的檔案傳輸時間。即使企業作了異地備援或資料複製的機制，系統還是存在資料同步的一致性

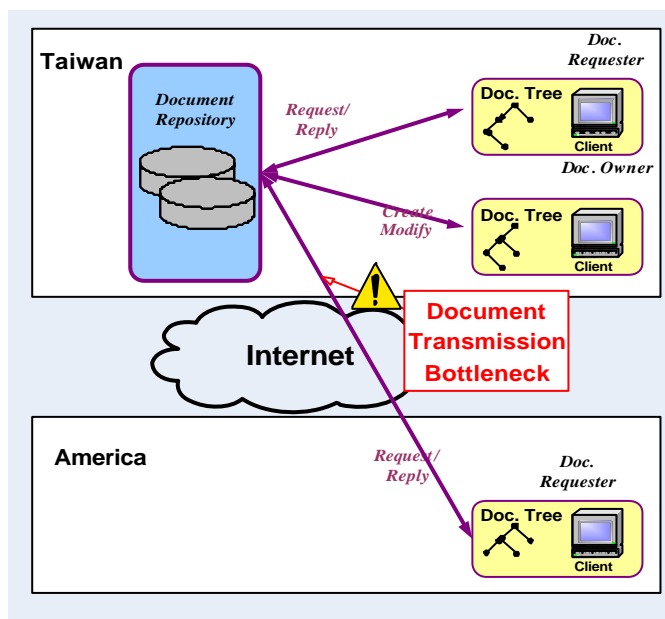
與大量資料傳輸所需的網路頻寬問題[4]。



圖一：集中管理、集中儲存的文件管理系統架構

2. 分散管理、集中儲存：

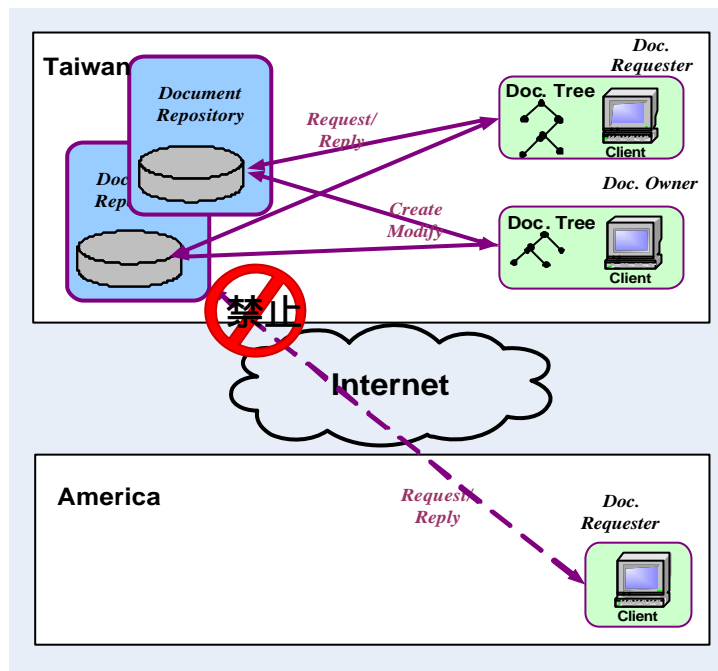
此類系統通常應用於企業的系统設計部門，存在的目的則是為了資料的安全與保護。此類系統的優點在於方便系統設計人員自行建置文件樹。同樣的，此系統的缺點在於文件集中儲存時，遠端使用者必須忍受檔案傳輸的低效率問題[5]，此架構如圖二所示，客戶端透過本身的文件樹發送一個文件請求給文件存儲器，而文件存儲器提供各個點的文件存取要求，但客戶端各自管理不同的文件系統。



圖二：分散管理、集中儲存的文件管理系統架構

### 3. 分散管理、分散儲存：

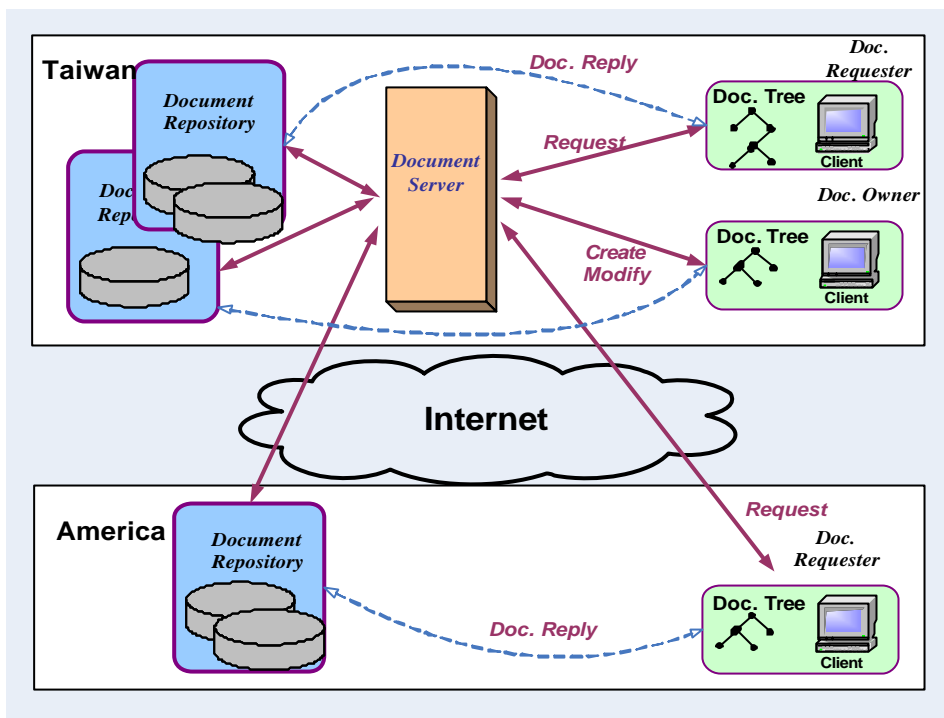
如圖三所示，客戶端透過本身的文件樹發送一個文件請求至特定的文件存儲器，而每一個文件存儲器只提供相同網域之客戶端的文件存取要求，且客戶端各自管理不同的文件系統。此類系統目前只有在分散式檔案系統有建置，例如 Unix NFS 可以提供使用者自行建立屬於自己的文件樹，也就是說並不提供統一的文件類別的管理介面。這類系統的優點在於提供很彈性的檔案組合管理機制，但是缺點是缺乏企業所需的統一的介面讓使用者瞭解完整的知識文件樹，以保證資料的一致性。另外一個嚴重的缺點則使用者必須與檔案儲存系統在相同網域才能使用[6]。



圖三：分散管理、分散儲存的文件管理系統架構

### 4. 集中管理、分散儲存：

為了提供企業一致性的知識地圖，企業需要有統一的文件樹，同時，為了文件傳輸的效率，也必須將文件分散儲存在各地。因此，文件集中管理、分散儲存是目前企業對於知識文件管理的首要目標[7]。本論文目的就是設計與實做一個有效率的分散式文件管理系統來滿足企業的需求，同時希望透過多層次的邏輯文件索引的設計以及快取的設計，來免除複雜的資料同步問題[8][9]，如圖四所示，客戶端透過本身的文件樹發送一個文件請求至文件伺服器(Document Server)，文件伺服器會找尋該客戶端要求的文件所儲存的文件存儲器，並傳送該客戶端資訊至此文件存儲器，而每一個文件存儲器提供文件給文件伺服器所請求的客戶端，且客戶端各自有不同的文件系統。

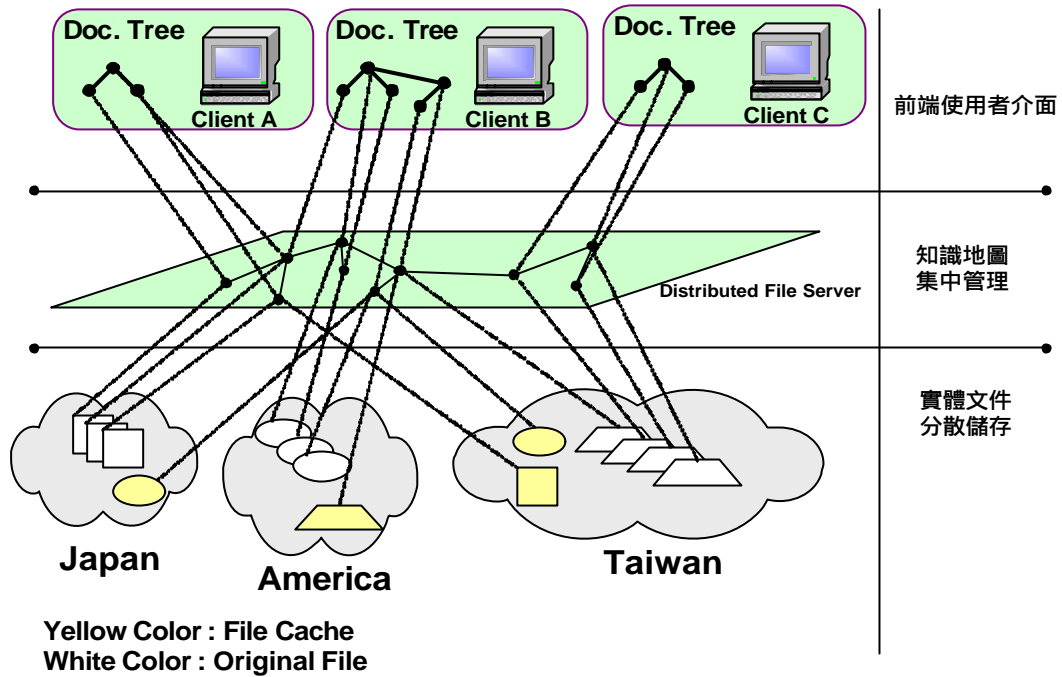


圖四：集中管理、分散儲存的文件管理系統架構

分散式儲存管理使用備份或複製檔案系統與資料以確保文件內容一致性，其中除了完整備份外，映射(Mirror)方法也是常用的[10]，映射的運作可分為同步跟非同步，映射同步為當發生在資料更新變動同時系統會做備份更新動作，而映射非同步為系統週期性的依照記錄檔來更新變動資料[8][11]。此方法固然可保持資料可用性，但確會造成大量的傳輸負荷，系統一再做更新資料，但不見得每筆資料都是重要且使用率高，所以常造成系統運作頻繁[12]。

### 三、適應性檔案存取機制

本論文主要在不改變企業現有的網路架構與頻寬的情況下，大量提升文件管理與存取效率，提供跨國企業完整的文件管理解決方案。此方法的重點在於設計透過集中管理的知識地圖，提供使用者一個一致性的存取介面，透過這個介面，使用者可以從任何地方進行文件存取的動作，而做到文件集中管理、分散儲存的管理機制。另一重點則是實體文件分散儲存，也就是說，使用者在存取文件之前不需要知道實體檔案的儲存位置。而為了整個檔案存取的效率，系統必須能自動適應使用者所在位置，並且動態的調整回復實體檔案的檔案伺服器。



圖五：具快取機制的分散式文件管理系統架構圖

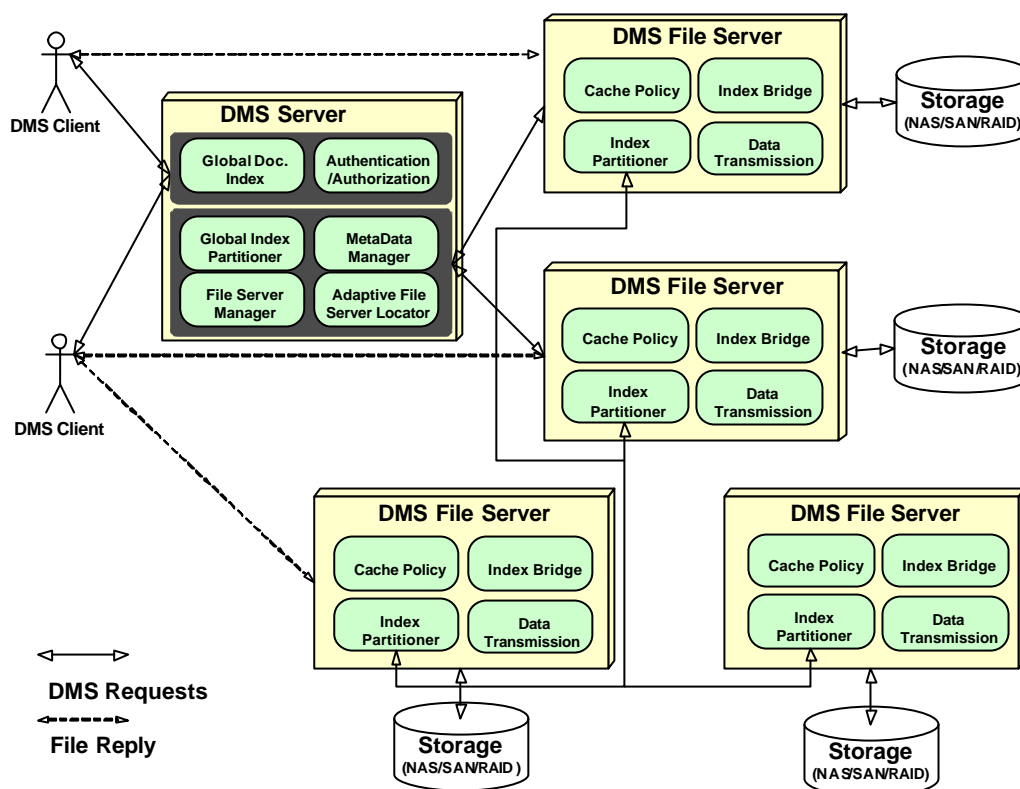
透過快取機制(Cache)的設計、多層邏輯檔案索引等使用者適應性機制的建立，所有的使用者將只透過本地端的檔案伺服器存取文件，如此可讓 99% 以上的文件存取效率與本地文件存取效率幾乎一樣快。系統如果固定使用者存取實體檔案的伺服器路徑，對使用者來說，將是非常沒有效率的設計，其方法的系統架構如圖五所示，各個客戶端的文件樹透過中間層的分散式檔案伺服器(Distributed File Server)，對應至各個不同地區儲存器的實體文件。黃色部分為檔案伺服器之間互相快取(Cache)的實體檔案[13]。

此系統的軟體架構如圖六所示，整個適應性分散式文件存取系統大致上可以分為兩個主要子系統，分別為 DMS 伺服器(Document Management System)以及 DMS 檔案伺服器。一個 DMS 可以連接多個 DMS 檔案伺服器，每 DMS 檔案伺服器也會藉由內部的 Cache Policy 模組連接到其他 DMS 檔案伺服器或透過索引分割樹(Index Partitioner)來與下一階層的 DMS 檔案伺服器互動。以下將敘述本架構的內部設計。

## ■ DMS 伺服器

DMS 伺服器主要負責接收所有的使用者的請求。對於 Metadata 的相關請求，則於 DMS 伺服器直接回應，如果是對於遠端文件的上傳或下載請求，則會透過 Adaptive File Server Locator 模組的判斷，將該請求送到對應的 DMS 檔案伺服器。

DMS 伺服器主要由 Global Document Index、Adaptive File Server Locator、Global Index Partitioner、Metadata Manager、File Server Manager 以及 Authentication/Authorization 模組組成。以下分別概述每一個模組的主要功能：



圖六：適應性分散式文件存取系統軟體架構圖

□ **Global Document Index**

此模組是主要提供企業內部單一文件樹的建置功能。透過此模組，可以提供企業使用者有一個集中管理的文件地圖。

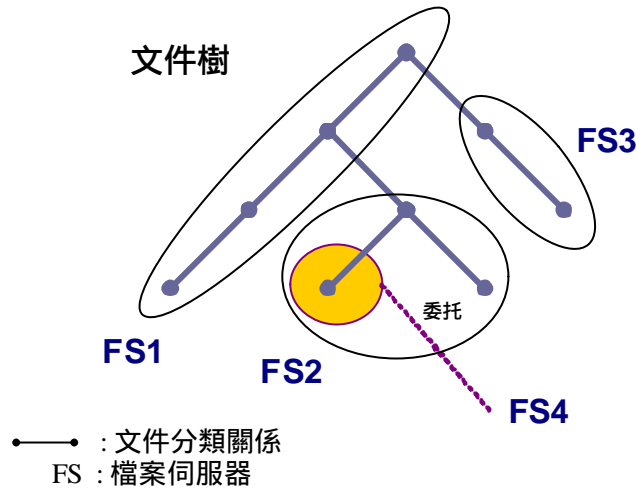
□ **Adaptive File Server Locator**

當 DMS 伺服器接收到使用者請求時，會立刻判斷使用者所屬位置決定該使用者對應的最有效率的 DMS 檔案伺服器，以便後續的檔案上傳或下載管理。由 Adaptive File Server Locator 透過 DNS 的查詢，可以假設該使用者最近的檔案伺服器為何者。因此，同一個使用者帳號可以由每個分公司登入系統。每次使用者登入，系統都會委派最近的 DMS 檔案伺服器來服務使用者請求。

□ **Global Index Partitioner**

此模組的主要功能是将企業內的單一知識地圖切成許多個子地圖，分別由一個 DMS 檔案伺服器來負責儲存。圖七可以清楚描述設計該模組的主要目的。樹狀結構即是企業內部完整的知識文件地圖。FS1~FS4 各代表不同文件關係樹，整個完整的文件樹可以被切分為三個主要 Partition，而每一個 Partition 分別為檔案伺服器 FS1, FS2 與 FS3 負責。其中 FS4 為 FS2 的下游檔案伺服器。此外，系統也可以再將 FS2 的其中一個檔案交由 FS4 管理。如此，透過 FS2 與 FS4 的委託關係，便可建構出階層式的檔案伺服器軟體架構。





圖七：文件索引分割樹示意圖

#### □ Metadata Manager

所有的文件都會有 Metadata，例如文件編號、文件名稱、最新版次等。為了維持資料的一致性，所有文件的 metadata 應該集中到 DMS 伺服器中負責統籌管理。每個文件版次都會有一個實際的檔案對應，因此，在版本管理部份，只要 Metadata 模組負責管理即可，並不需要在各個檔案伺服器之間保持版本內容的一致性。

#### □ File Server Manager

此模組主要是提供管理人員可以動態的增加新的檔案伺服器，以及讓 DMS 伺服器瞭解到目前總共有多少檔案伺服器存在。有新的檔案伺服器加入 DMS 伺服器時，檔案伺服器模組必須和所有的檔案伺服器更新資料。如此可以讓其他的檔案伺服器可以 Cache 最新的檔案伺服器資料。

#### □ Authentication/Authorization Module

此模組主要是讓使用者可以做權限管理以及帳號管理與系統登入。我們也預計將檔案服务器的設計信任此模組，避免有不肖份子直接透過檔案伺服器拿走文件資料。

### ■ DMS 檔案伺服器

DMS 檔案伺服器子系統是適應性分散式文件存取系統的關鍵模組。主要負責企業文件地圖與實體檔案存取位置關係的對應、檔案快取的設計，以及負責建置階層式檔案伺服器架構。DMS 檔案伺服器主要由 Index Partitioner、Cache Policy、Data Transmission 以及 Index Bridge 四個模組組成。以下分別描述每一個模組的功能。

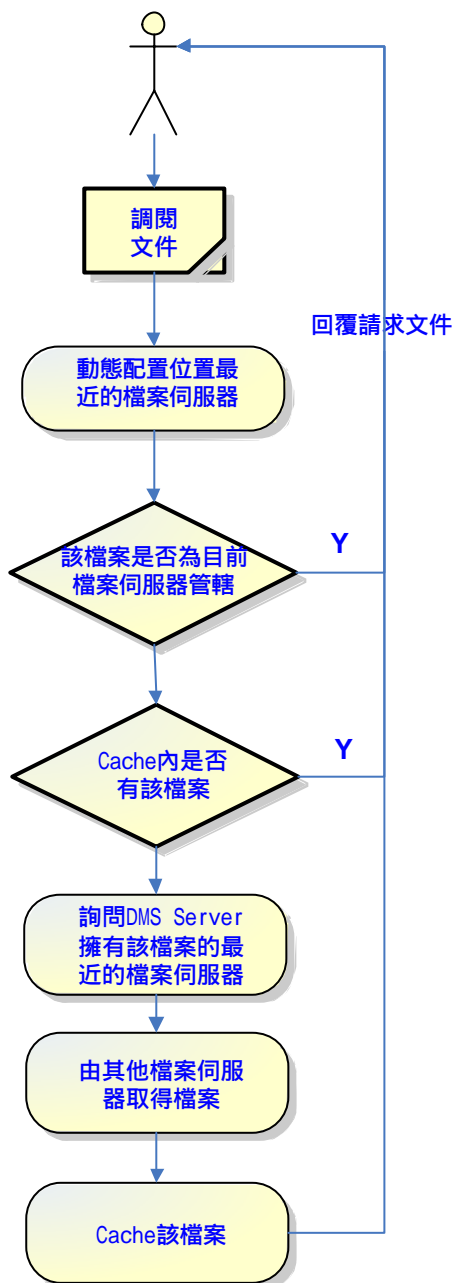
#### □ Index Partitioner 模組

此模組主要的目的是建構階層式 DMS 檔案服务器的管理模組。當有特殊或機密的資料需要特別保護或儲存、備援時，或當分公司所屬地區地理範圍比較廣泛時，可以透過這個模組將 DMS 檔案伺服器負責的文件，在委由另一個可用度(availability)以及可靠度(reliability)比較高的 DMS 檔案伺服器負責管理。如同文件索引分割樹一般，DMS 檔案服务器的索引分割樹將自己負責的部

分文件地圖再細分成幾個更小的子地圖，而將該子地圖對應的實體檔案委由其他 DMS 檔案伺服器儲存管理。

### □ Cache Policy 模組

Cache Policy 模組負責管理當地使用者存取文件時，發生 Cache miss 的文件實體檔案。目前我們規劃的 Cache Policy 如下：



圖八：Cache 存取管理演算法

#### 1. Cache in policy

我們預計將文件快取的模式分為以下三種：要求模式(On-Demand)、定期模式(Periodical)以及手動模式(Manual)。要求模式是指當使用者有文件下載，但發生 Cache miss 時，立刻啟動快取機制(Cache)，將檔案從遠端的

DMS 檔案伺服器快取到目前的 DMS 檔案伺服器。定期模式則是每天固定時間將其他 DMS 檔案伺服器的檔案快取到目前的 DMS 檔案伺服器。手動模式則是管理者隨時可以決定要快取那個檔案到 DMS 檔案伺服器中，可以提供臨時的檔案傳輸需求。至於系統將採用哪種模式的 Cache 策略則由管理者自行決定。

## 2. Cache 管理演算法

我們針對分散式文件存取的特性，設計更有效率的演算法。以下是我們設計 Cache 管理的概念如圖八，使用者若調閱文件，系統將配置一個最近的檔案伺服器，並查詢該文件是否儲存於此檔案伺服器，如果有直接下載文件，否則確認 Cache 是否有該檔案；如 Cache 有該檔案則下載文件，否則 DMS 伺服器會傳送該檔案被儲存最近的檔案伺服器資訊並複製目標檔案，且 Cache 該檔案。使用者調閱文件，透過此演算法決定出下載路徑與系統是否啟用快取機制。

## 3. Cache out policy

我們將依照管理者設定的 Cache Size，在 Cache Size 剩下不到管理者設定的 threshold 時，於半夜啟動 Cache out 機制。而 Cache out 的演算法則採用 LRU (Least Recently Used) 的方式[14]，將最少用到的快取檔案清除。

### □ Data Transmission

此模組主要負責檔案的傳輸，需要特別獨立此模組是因為部分產業的檔案 Size 非常大，需要特別管理上傳的檔案格式、以及傳輸時間。尤其當透過 Web 的方式上傳、下載檔案時，常會發生暫停(timeout)的問題，必須透過資料傳送 (Data transmission) 模組專責處理相關問題。

### □ Index Bridge

此模組負責將邏輯的檔案索引，對應到實際的檔案儲存位置。例如可以將 DMS 伺服器的 \root\SOP\請假標準程序，對應到 DMS 檔案伺服器 FS1 的 \SOP\請假標準程序，再對應到 FS1 的 d:\00000001\00000002.doc。或當 FS1 後端的檔案系統是 NFS 時，則必須對應到 /user/home/files\_server/001/003.doc。透過這個模組的建置，我們可以將實體檔案儲存的檔案系統以及儲存空間做很好的分離切割。如此才能連接到企業既有的所有大型的儲存空間。

## 四、方法分析

此章節我們透過一般網路傳輸速度來分析所提出的演算法。假設有兩個檔案伺服器分別架設於台灣與外地(非台灣)兩地，其網路架構是雙向 512K，本地端的內部網路速度為 100MByte/sec。以下分別分析有無使用適應性檔案存取機制的檔案存取狀況，其中 A 代表文件在網路上的傳輸時間、B 代表文件儲存時間、C 代表使用者端檔案開啟時間。表一為一般網路檔案下載情況，表二為有適應性檔案存取機制的檔案存取分析且 DMS 伺服器位於台灣，那麼檔案傳輸的時間如表 (以 10M 檔案大小計算)，精確的時間尚須

視使用者的機器設備能力而定。

表一：一般檔案下載情況

時間種類 傳輸情況	A 的時間	B 的時間	C 的時間	全部時間
1 人 (台灣到台灣)	1~2 秒	1~2 秒	1~2 秒	6 秒
1 人 (外地到台灣)	20 秒	1~2 秒	1~2 秒	24 秒
10 人 (外地到台灣)	200 秒	1~2 秒	1~2 秒	204 秒

表二：使用適應性檔案存取機制下的檔案下載情況

時間種類 傳輸情況	1 的時間	2 的時間	3 的時間	全部時間
1 人 (台灣到台灣)	1~2 秒	1~2 秒	1~2 秒	6 秒
1 人 (外地到台灣)	1~2 秒	1~2 秒	1~2 秒	6 秒
10 人 (外地到台灣)	約 6 秒	1~2 秒	1~2 秒	10 秒

以上計算方式均以理論值之最大極限計算，不考慮平時網路被其他應用系統或資料傳輸所佔據頻寬的情況。在使用適應性檔案存取機制的情況下，使用者如下載為非檔案伺服器所擁有的檔案或非 Cache 檔案則只須花費一次遠端下載時間，之後其他使用者只須花費本地端下載時間，由結果得知，此作法大大減少多端點與不同網域檔案下載時間。

## 五、結論

目前企業多屬跨國公司，企業最重要的智慧資產就是文件，往往會因為文件維護單位的設立地點不同而導致文件散落在各個地區，此外台灣資訊部門對於關連式資料庫的技術依賴性太高，導致有很多新的系統功能無法被快速開發，多層次的邏輯檔案索引，以及適應檔案伺服器的 Cache 即是最好的解決辦法。

本論文設計的適應性的分散式文件存取系統，提供自動化的適應能力，根據使用者來源，調整文件資料回復的檔案伺服器，以及設計邏輯檔案索引與 Cache 來加強文件存取的效能，期望能有助於建構適用於目前企業網路架構與頻寬的高效能分散式文件管理系統。

## 誌 謝

本研究承蒙國科會計畫 (編號：NSC 94-2213-E-029-010) 的部分贊助，特此感謝。

## 參考文獻

- [1] A. Muthitacharoen, B. Chen, D. Mazieres, "A Low-bandwidth Network File System," in Symposium on Operating Systems Principles, 2001, pp.174-187.
- [2] C.I. Lee, H.H. Lin, B.S. Luo, and Y.G. Wang, "System and Method for Synchronizing Documents between Multi-Nodes," Patents of Chinese Taipei, No. 2330713, Sept. 2004.
- [3] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K.J. Worrell, "A Hierarchical Internet Object Cache," in Proceedings of USENIX Annual Technical Conference, 1996, pp. 153-164.
- [4] M. B. Blake, P. Liguori, "An Autonomous Decentralized Architecture for Distributed Data Management and Dissemination," in IEICE Trans. on INF. & SYST. Vol. E84-D, No 10, Oct. 2001.
- [5] M. G. Baker, J. H. Hartman, M. D. Kupfer, K. W. Shirriff and J.K. Ousterhout, "Measurements of a Distributed File System," in Proc. of 13th Symp. On Operating Systems Principles, Oct. 1991, pp.198-212.
- [6] J. H. Howard, M.L. Kazar, S. G. Menees, D.A. Nichols, et al., "Scale and Performance in a Distributed File System," ACM Transactions on Computer Systems, Vol.6, No.1, Feb. 1988, pp.51-81.
- [7] J. L. Xu; J. C. Liu; Bo Li; X. H. Jia, "Caching and Prefetching for Web Content Distribution," Computing in Science & Engineering, Vol. 6, No. 4, Jul. 2004, pp. 54-59.
- [8] T.Y. Huang, T.Y. Chen, P.Y. Chuang, "Constructing a Scalable Universal File Cache in a Heterogeneous High-Speed Distributed System," in Proc. of Symp. on Operating Systems Design and Implementation(OSDI'04), Dec. 2004.
- [9] D. Zeng, F.-Y. Wang; M. K. Liu, "Efficient web content delivery using proxy caching techniques," Systems, Man and Cybernetics, Part C, IEEE Transactions, Vol.34, No.3, Aug. 2004, pp.270-280.
- [10] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso, "Understanding Replication in Databases and Distributed Systems," Proc. 20th Int'l Conf. Distributed Computing Systems (ICDCS '00), 2000.
- [11] T. C. Jepsen, "The Basics of Reliable Distributed Storage Networks," IT Pro, June 2004.
- [12] P. Rodriguez, E. W. Biersack, "Dynamic parallel access to replicated content in the Internet," Networking, IEEE/ACM Transactions, Vol.10, No.4, Aug. 2002, pp.455-465.
- [13] T.D.N. Francisco, "Cooperative Caching Middleware for Cluster-Based Servers," in 10th IEEE Int'l Symposium on High Performance Distributed Computing, IEEE Press, Aug. 2001, pp303-314.

- [14] S. J. Lee, and C. W. Chung, "VLRU: Buffer Management in Client-Server Systems," in IEICE Trans. on Information and Ssystems, Vol. 83-D, No. 6, June 2000, pp. 1245-1254.
- [15] P. Biswas, Nashua, "Issues in Cache Management Algorithms for Commercial Software Systems," in Proc. of 1st Workshop on Software and Performance, 1998, pp.76-77.
- [16] G. Chen, C. Wang and F. Lau, "Building a Scalable Web Server with Global Object Space Support On Heterogeneous Clusters," in Proc. of IEEE International Conf. on Cluster Computing, Oct.2001, pp.313-320.
- [17] E. Panagos, A. Delis, "Selective Replication for Content Management Environments," IEEE Internet Computing, Vol.9, No.3, June 2005, pp.45-51.