

複合式文件隨選系統之同步管理機制設計與製作

Design and Practice of A Synchronization Management Mechanism in A Synthetic-Document-on-Demand System

童曉儒

國立屏東科技大學 資訊管理系
屏東縣內埔鄉學府路 1 號
srtong@mail.npust.edu.tw

賴文蘋

資策會網路及通訊實驗室
台北市南京東路二段 116 號 5 樓
plmicro@netrd.iii.org.tw

摘要

在複合式文件隨選系統(Synthetic-Document-On-Demand System 或 SDOD System)中,文件是由數種資料流(如影像、聲音與圖片等)所組成,因此在儲存與傳送時必須特別考慮到媒體間同步的關係,在先前的研究中[2],我們探討伺服器上支援媒體間同步的機制,期盼以最節省伺服器資源的方式,來實踐媒體間的同步關係。而本論文是該研究的延伸,加入用戶端的同步機制設計,以構成一套完整的 SDOD,可提供點對點或群體廣播,我們將針對實際開發完成的 SDOD 系統模組與運作內容做一詳細的說明。

關鍵詞:複合式文件、媒體間同步、時序關係、媒體共享

一、簡介

隨著資訊科技的發展,為了滿足人類視覺與聽覺的感官需求,愈來愈多的多媒體資訊透過網路傳送至使用者,讓使用者藉由電腦中的文字、聲音、影像、視訊、動畫等媒體來達到傳遞資訊的目的。圖 1 以流程方式說明多媒體資訊的產生、壓縮、儲存、傳送與播放的處理流程,多媒體資料通常儲存在一台或多台媒體伺服器內,提供作為隨選視訊或視訊會議等多媒體應用的資料儲存裝置,由於多媒體資料量大,以及通訊網路頻寬有限,因此多媒體資料在傳送到網路之前,應先完成壓縮處理的工作,再以串流(streaming)的方式傳送,以降低傳送資料的成本,在讀取播放之前,再作解壓縮與組合的處理。例如經過 MPEG 技術壓縮過的影音資料量約可以減少百倍[4],使得影音資料可以透過網路提供即時的播放功能。

然而多媒體應用日趨複雜,單一的影音資料已經無法滿足實際的需求,例如遠距教學需要同時播放老師的影像、聲音與授課相關內容的投影片等多種媒體資料,因此我們將不同類型的媒體(文字、圖片、聲音、視訊、投影片、

動畫等),依照某些事先定義的劇本(script)整合,編製成為一個多媒體節目播放,稱之為「複合式文件(Synthetic Document)」。而這些相關的媒體可能儲存於不同的媒體伺服器,當播放時,依照 script 從各媒體伺服器中擷取,然後在客戶端整合,再依照 script 既定的時序關係(Temporal Relationship)[10]來播放,如同交響樂演奏一般,稱為「媒體協奏(Media Orchestration)」。提供這樣複合式文件點選功能的系統,我們稱之為「複合式文件隨選系統(Synthetic Document-On-Demand System 或 SDOD System)」。

然而在網路的環境下要達成媒體間時序同步並非易事,主要是因為在資料傳輸的過程中,會遭遇各種不同的延遲[7],我們可以將其分成三個主要子系統來討論:「客戶端」、「網路」、「伺服器端」,如圖 2 所示,從伺服器端擷取媒體資料,透過網路傳輸,在客戶端的播放介面中呈現,都會遭遇許多的延遲,因此必須在各子系統中進行同步控制。

在「客戶端」子系統,就放映壓縮後的影片而言,如 MPEG-I 影片的畫面是以固定畫面張數速率(constant frame rate)被播放,由於畫面必須完整收到,且每張畫面資料量不同,所以在交送解壓縮之前,通常必須準備一暫存區來暫存資料[9]。因此自資料抵達客戶端到實際開始被使用或播放,會有些許的啟始延遲,此外客戶端作業系統的負荷量也可能影響媒體資料處理的進度,通常延遲的變異度約在數十個 millisecond。

就「網路」子系統提供的傳送服務而言,主要的任務是將伺服器暫存區內的資料傳送至客戶端暫存區內,由於影音媒體具有連續性,通道多是以資源預留為基礎的方式建立[12],提供具有延遲上限的 CBR/Constant Bit Rate 通道。亦或者在網路異質性、有限資源、網路隨機流量等前提下,提出使用智慧型的排程機制[11]來保證網路上 inter-stream 與 intra-stream 同步,通常延遲的變異度約在數十

到數百個 millisecond。

就「伺服器」子系統而言，先前有關的研究多以磁碟機為主要的硬體架構[8]，能否同時提供大量且無間斷的視訊流，往往是設計伺服器的重要目標。為了達到此目標，「資料輪置(data striping)」是一項相當普遍且實際用來提升磁碟機輸出效率的重要技術[5][6]，媒體伺服器為了要提升磁碟機的效能，常使用 data striping 的技術來存取資料，然而這也限制了隨意擷取(Random Access)的彈性，而導致相當長的啟始延遲，通常的延遲變異度約在數秒到數十秒之間。

要維護正確的時序關係，最主要的工作是消弭資料在傳遞時所累積的延遲變異度，如此我們就可以事先規劃資料的抵達時間。而完成時序關係，較有效的作法是在各子系統中對延遲變異度加以控制，也就是在伺服器端做較粗略的延遲控制，而在網路與客戶端再作較細微的延遲控制，如此處理負擔可有效的被分散。而就各子系統而言，伺服器的延遲變異度長達數十秒，影響最巨，因此它的控制更形重要。在先前的研究中[2]，我們專注在伺服器子系統方面，探討如何修改或強化以資料輪置為基礎的伺服器，用來支援媒體資料之間的時序關係。而本文延續該研究，將實際系統的模組與運作做進一步的闡述，特別是加入了「客戶端」子系統的同步控制設計，以達到端點至端點的同步控制，使得整個 SDOD 系統得以順利運作。本文於第二節回顧媒體伺服器的設計原理、影響時序同步的變因與所提出之時空映射機制模型；第三節中描述系統架構與設計流程；第四節描述系統細部模組架構與實作的步驟；第五節為結論。

二、伺服器同步控制機制回顧

(一) 啟始延遲變異度與暫存區

媒體伺服器大多採用資料輪置的技術來提升磁碟機效能，滿足同時大量讀取資料的需求。資料輪置的技術如圖 3 所示，假設媒體伺服器由 3 個相同容量與效能的磁碟機(disk)組成，經由一條匯流排聯結，每台磁碟機切割成 4 個磁區(zone)，所有磁區形成一個 3×4 的二維矩陣，影片資料事先切割成小資料塊 x ，然後以磁區為主的方式依序循環存放於矩陣之中。當要讀取資料時，每台磁碟機的讀寫頭再到各個磁碟的磁區中將資料塊讀取出來，稱為資料輪置。假如讀寫頭在每個磁區停留的時間(稱為「磁碟機週期」)為 1 秒，目前想要讀取 x_7 資料塊，但讀寫頭停在 x_8 的位置，則讀寫頭需要循環到 Disk 2 的 Zone 3 才能讀到的資

料，因此需要 12 秒，這 12 秒就是讀取資料需要等待的最長磁碟機週期，稱為啟始延遲變異度(Initial Delay Jitter)。

在資料輪置的伺服器中，磁碟機的效能與資料塊的大小有關，當資料塊愈大，讀寫頭實際讀取資料所需的時間變長，而花在移動讀寫頭的時間比率變小，所以效能愈佳，但啟始延遲變異度跟著變長。這種設計在一般以點播節目為主的應用環境中，觀賞者可以接受稍長的啟始等待；但在複合式文件的應用環境中，不同媒體間有著強烈的同步關係，如果在媒體儲存前未能妥善規畫儲存的位址，在讀取資料時過長的啟始延遲，可能會消耗掉可觀的資源，方能滿足同步的需求。

為了改善啟始延遲變異度所造成的影響，一般的做法是利用系統的暫存區來做預取(pre-fetch)的動作，如圖 4 所示，當媒體 A 播映到時間 t sec 時，啟動媒體 B 開始播映，如果最大的啟始延遲(即啟始延遲變異度)為 d sec，則我們必須在時間 $t-d$ 時即對媒體 B 發出播映需求，因為媒體 B 的啟動時間不確定(在時間 $(t-d)$ 與 t 之間)，所以在考慮最差狀況下，我們必須預留 d sec 長度的資料空間給媒體 B 做為緩衝之用，如資料流間存在許多同步點，則必須預留可觀的暫存區空間。

想要降低啟始延遲變異度對暫存區的依賴度，就是對啟動資料流的時間加以規範，直覺上可行的方法就是將所有媒體整合成一複合式文件儲存，如此媒體彼此的時序關係反應在它們的儲存位址上，使得當資料輸出時，讀寫頭可以省掉額外的搜尋負擔，立即完成資料流彼此間的同步動作。這種方法的優點為簡單、直接，因為一旦最先的資料流被啟動，接下來跟著被啟動的資料流就沒有啟始延遲所造成的困擾，不必預留任何額外的暫存區資源來解決同步問題；但最大缺點為缺乏資料共享的彈性，因為某一資料的放置是根據與另一特定的媒體資料流同步而決定，所以若該資料必須與一個以上的媒體資料流同步時，就無法兼顧多個同步需求。再者由於資料放置的位址有所限制，可能影響到伺服器空間的使用效率。最後由於複合式文件中資料的型態不同，導致各別的頻寬與容量需求不盡相同，使得伺服器的資源管理問題更形棘手，如未能提出有效管理策略，系統效能將大打折扣。

(二) 同步容錯度

另一項會影響暫存區空間的因素為「同步容錯度(Synchronization Tolerance, ST)」，並不見得所有媒體間的同步要求十分的精確，例如：演講者影音資料的啟始同步不能容許太大的誤差，否則會造成不對口(lip-sync)的現象，

又稱為 fine-grained sync。而配合演講者所呈現的投影片或影片片段則可容許有限度的誤差。相對的背景音樂或影片則純屬陪襯性質，故容許高容錯度，又稱為 coarse-grained sync。將同步容錯度納入考慮的最大優點，主要是為了節省暫存區資源，以前例而論，若媒體 A 與 B 間所容許的同步誤差度(ST)為 s sec，我們僅需在 $(t-d+s)$ sec 時對媒體 B 發出播映需求即可，而所需的暫存區減少為容納 $(d-s)$ sec 長度的資料量即可，如圖 5 所示。

(三) 相依度

另一項重要的設計考量是「相依度(Degree of Sharing, DS)」，它代表著其他會參考(啟動)該媒體的媒體數目，如果複合式文件中，媒體 B 只有被媒體 A 參考(啟動)，也就是媒體 B 的 DS 值為 1，則媒體 B 與媒體 A 可整合成一個複合物件(compound object)儲存，如此媒體彼此的時序關係反應在它們的儲存位址上，使得當資料輸出時，讀寫頭可以省掉額外的搜尋負擔，立即完成資料流彼此間的同步動作。例如，如圖 6 所示，當媒體 A 播放至區塊 a_3 時，媒體 B 開始播放，故將媒體 B 的啟始點(即區塊 b_0)放置於相同於區塊 a_3 的磁區中。這種方法的優點為簡單、直接，因為沒有啟始延遲所造成的困擾，不必預留任何額外的暫存區資源來解決同步問題；但此作法的先決條件，必須沒有其他媒體會參考到媒體 B (也就是媒體 B 的 $DS=1$)。另一個限制是整個複合物件(媒體 A、B)必須存放於同一媒體伺服器中。

(四) 時空映射機制

為了方便伺服器處理媒體同步，並有效利用資源，我們將複合式文件依既訂的時序關係(同步容錯度與相依度)，轉換成便於處理的同步格式，同時我們將 disk 組織成不同啟始延遲變異度的伺服器群，屬於某種同步格式的資料，將集中儲存於滿足該同步格式特性的一個伺服器上。當讀取複合物件時，我們依既訂的時序關係預留適當的暫存區，啟動讀取資料，以達到預期的同步效果，並且達成以下的設計目標：

- (1) 縮小啟始延遲變異度，以節省暫存區資源。
- (2) 提供資料放置位址的彈性，以提升儲存空間使用效率。

這整個映對轉換的過程我們統稱為「時空映射」[2]，其中包括三個步驟：(1)同步關係建立、(2)複合物件最佳化、(3)與媒體配置，如圖 7 所示。

為方便說明，我們以下述情境為例，假設有兩個複合式文件 SD_1 與 SD_2 ，其中為不同演

講者用不同的語言(英文與中文)，講解同一套投影片，因此所涉及的資料流與彼此的時序關係如圖 8 所示，其中包括兩道視訊流(V_1 、 V_2)及音訊流(A_1 、 A_2)，一道投影片流(S_1 、 S_2 、 S_3)。視訊與音訊間為 fine-grained 同步，投影片於不同的時間點與個別的視訊流同步，屬於 coarse-grained 同步。

(1). 「同步關係建立」：首先將圖 8 中媒體資料流彼此的時序關係轉換成圖 9 的同步關係圖，其中每一媒體對應一個節點(node)，如果媒體 A 必須啟動媒體 B，則建立一條由節點 A 到節點 B 的 direct link。節點的 incoming degree 代表該媒體的相依度(DS)；伴隨著每條 link 的同步容錯度(ST)係指所容許的同步時間誤差。某些媒體可能除了在這兩個複合式文件中被使用到之外，也會被其它的複合式文件使用，因此預設成可公開(public)分享，我們以「⊕」符號表示之。以本例而言，視訊(V_i)必須隨時調整與音訊(A_i)同步，故其間 link 的 $ST=1$ ；視訊(V_i)與投影片(S_i)間的同步較粗略，故其間 link 的 ST 較長(3 或 5)， ST 的值可依慣例或由外界定義。

(2). 「複合物件最佳化」：接著我們針對圖 9 的同步關係圖加以簡化成圖 10 的複合同步關係圖，主要的任務是盡可能的將媒體合併成複合物件，以利下一階段的媒體配置工作。其依據之法則有二：

A. node 必須不能公開的(public)，如果為公開的就必須與其它複合式文件中的媒體分享，因此不能與其它 node 合併為複合物件儲存。

B. $DS=1$ 的 node 因只與其 parent node 同步，而與其它 node 無關，故可與 parent node 合併成一複合物件。

合併的 node 必須符合上述兩個條件，並且合併後的複合物件繼承原先未合併 node 之 links，如果 links 重複指向同一物件，則僅保留 ST 為最小的那一條 link。合併的工作以 recursive 的方式進行，直到沒有任何 $DS=1$ 且為非公開的節點存在為止。就本例而言 node A_1 與 node V_1 合併；node A_2 與 node V_2 合併。

(3). 「媒體配置」：最後我們要決定配置多少資源給媒體。由於啟始延遲變異度，與磁碟機效能成正比，但也代表需要更多的暫存區，因此我們摒棄單一伺服器的作法，改採多重伺服器，各伺服器具有不同的啟始延遲變異度，而這些伺服器實際上可能是建構於同一或一個以上的實體伺服器之上。各節點依照下述法則配置於子伺服器

中：假設共有 x 個子伺服器，依序編號為 $0, 1, 2, \dots, x-1$ ，子伺服器 i 的伺服器週期為 c_i ，且如果 $i > j$ 則 $c_i > c_j$ 。某節點(媒體)的 ST 則定義為所有指向該節點 link 的 ST 中最小者。假設將該媒體放置於子伺服器 i 中，在播放時，於同步點前 c_i 的時間發出點播要求，如果 $c_i \leq ST$ ，則不需要額外的緩衝區，即可在容許的同步誤差內取得資料流；反之，如果 $c_i > ST$ ，則 ST 無法完全吸收啟始延遲變異度，故必須預留 $c_i - ST$ 時間長度的緩衝區。因為媒體放置於愈小的伺服器週期的伺服器中，代表付出愈昂貴的磁碟機效能代價，反之如果放置於愈大的伺服器週期的伺服器中，則可能必須付出暫存區的代價，所以為了達到這兩種狀況的平衡，採行的策略是，如果 ST 落於 c_i 與 c_{i+1} 間，則將媒體配置於子伺服器 $i+1$ 中，然後於點播時預留 $c_{i+1} - ST$ 資料流長度的緩衝區。就本例而言，假有三個子伺服器 $0, 1, 2$ ，且 $c_0=4, c_1=7, c_2=10$ ，則節點 S_1 與 S_3 之 ST 皆為 3 ，故配置於子伺服器 0 中；節點 S_2 之 ST 為 5 ，故配置於子伺服器 1 中；節點 (A_1, V_1) 與 (A_2, V_2) 無特定之同步需求，故配置於子伺服器 2 中即可，如圖 11 所示。當其中一個子伺服器的容量 overflow 時，可以分配給其它的子伺服器，選擇考慮的方式有兩種：第一是磁碟機效能，若要較好的磁碟機效能表示需要更多的暫存區；第二是選擇較少暫存區，但是磁碟機效能將會較差，因此必需在兩者之間衡量，選擇較適合的子伺服器。若編號 1 的伺服器容量 overflow，則必須選擇編號 0 或 2 的子伺服器，選擇伺服器 0 ，可以不需要額外的暫存區，但是付出磁碟機效能的代價；選擇伺服器 2 ，則需要額外的暫存區。

三、系統架構與流程設計

以「時空映射機制」為基礎，本章將描述我們所實際開發的 SDOD，整個系統分為三個部分，分別為「媒體伺服器(Media Server)」、「媒體節目管理子系統(Program Manager System)」與「媒體節目播放子系統(Program Presentation System)」，整個系統中各子系統之間的關係如圖 12 所示，媒體伺服器負責將複合式文件依既定的時序關係擺放到實體磁碟機中，讀取時讓被共享的媒體在適當的時間被重複啟動，使系統可以維持複合式文件內媒體間的同步，並達到媒體共享的理念，媒體伺服器部分請參閱 [1]，在此我們的重點在媒體節目管理子系統與媒體節目播放子系統。媒體節目管理子系統主要負責(1)媒體節目製作與(2)節目排程允入控

制管理，客戶端的媒體節目播放子系統主要負責複合式文件的播放，各子系統的功能如圖 13 所示。本系統設計了兩種傳輸的方式，分別為單點傳送(Unicast)與多點傳送(Multicast)，在單點傳送方面，每個客戶端可以隨時連線至媒體節目管理子系統查詢節目，選取想要收看的節目後，系統會即時將節目傳送到客戶端。在多點傳送方面，當客戶端想要收看節目時，先至媒體節目管理子系統查詢節目表單與節目播放時間，預訂想要收看的節目，並加入此節目的群組，節目排程允入控制機制會在預定時間廣播媒體節目，此時加入此群組的客戶端會收到媒體伺服器所廣播出來的節目，每一媒體節目佔用一個 multicast channel (multicast IP)，每一媒體資料流佔用一個 multicast session (multicast IP + port)，如果有任何一端資源預留不成功，則取消節目播放或調整節目的排程。

(一) 媒體節目管理子系統

媒體節目管理子系統內部模組架構(如圖 14)可以分為以下二個模組：「媒體節目製作(Program Maker)」與「節目排程允入控制機制(Admission Control Mechanism)」。

- (1) 媒體節目製作程序首先將媒體資料，依照節目劇本需求，以及媒體的特性、媒體內(intra-medium)的時序關係、啟始播放時間、磁碟機的傳輸速率等資訊，預先安排媒體播放排程，以時序語言(Temporal Script, TS)[3]描述完成，然後將時序語言交給時空映射機制處理媒體間(inter-media)的時序同步關係，而實際媒體資料則分類整理編號後儲存在媒體資料庫中。
- (2) 時空映射機制主要是依照媒體之間的同步關係，決定媒體應如何配置於伺服器中。時空映射機制接收到時序語言後，依據時序語言中媒體的資訊，進行媒體之間同步的判斷，並且將適合的媒體合併為複合物件，合併後的媒體資訊稱為空間語言(Location Script)[3]，最後將空間語言與使用到的實際媒體資料交給伺服器的 Location Oriented Mechanism 執行實際媒體資料的擺放。
- (3) 當客戶端想要收看節目，可以連線至節目排程允入控制機制，完成註冊流程與付費手續，等待系統確認用戶身份無誤後，會依照使用者的繳費情形給予不同的觀賞權限與密碼，使用者再依照擁有的權限，查詢目前欲觀看的節目或是預定時間播放的節目。
- (4) 節目排程允入控制機制開始安排此客戶端需要的資源，檢查客戶端、網路端與伺

伺服器端所需要的頻寬、緩衝區、群播通道...等資源，如果有任何一項資源無法預留成功，則通知此客戶端無法收看該節目。

- (5) 如果資源預留成功，則傳送”媒體播放時序資料”至客戶端，並且傳送一個訊息給伺服器的 Channel Oriented Mechanism，通知它依照複合物件之間的時序關係，開始依序將媒體資料從磁碟中讀取出來，或是在預定時間執行該節目的廣播，直到整個節目播完為止。

(二) 媒體節目播放子系統

媒體節目播放子系統分為下列幾個模組，我們以一使用者要求觀看一部影片為例，描述系統模組之間的互動關係如圖 15 所示：

- (1) 使用者連線至節目排程允入控制機制查詢節目選單，完成註冊流程與付費手續，等待系統確認用戶身份無誤後，會依照使用者的繳費情形給予不同的觀賞權限與密碼，使用者再依照擁有的權限，點選一部想要收看的節目。節目排程允入控制機制開始安排此客戶端需要的資源，如果有任何一項資源無法預留成功，則通知此客戶端無法收看該節目。
- (2) 節目排程允入控制機制將 Program Maker 產生的”媒體播放時序資料”傳送到播放排程機制(Actor)。”媒體播放時序資料”包括媒體的種類、媒體的傳送通道、資料流的編號、媒體的啟始播放時間、以及媒體的結束播放時間。
- (3) 媒體伺服器的 Channel Oriented Mechanism 在預定播放的時間開始將節目所需的媒體資料讀出並傳送至節目播放子系統，每一種媒體資料流透過適合的通道傳送。
- (4) 網路通訊協定接收媒體資料後，依照資料流的編號將媒體傳送到個別的緩衝區內，當媒體資料累積到適當的累積量後，在啟始播放時間播放。
- (5) 播放排程機制(Actor)負責將各個資料流中的媒體資料依時序關係排序，以便即時監督並啟動媒體資料在播映時間從緩衝區讀出，以及在結束時間通知播放程序停止播放，達到媒體同步的目的。
- (6) 由於各種媒體的特性不同，例如 MPEG 視訊資料需要經過解壓縮處理，因此每一種媒體需要不同的播放介面來播放，播放程序就是負責將媒體資料轉換為可以播映的格式，再透過各種播放瀏覽器將媒體資

料呈現在客戶端。

四、系統實作說明

(一) 系統開發環境

主要針對媒體節目管理子系統中的「時空映射機制」，以及客戶端的「媒體節目播放子系統」予以實作，系統發展平台為 Windows 2000 Advanced Sever，使用 Visual C++ 6.0 與 Microsoft DirectX Media 6.0 SDK 作為開發工具，完成這兩個子系統模組的建置，網路傳遞方式分為 Unicast 與 Multicast 兩個部分，希望能夠在理論之外，證明研究系統架構的可行性。

我們以一個遠距教學的例子來說明系統實作，這個複合式文件共有五道媒體資料流，其中有一道音訊資料流播放老師講解課程的聲音資料 Audio(A_{11})，兩道視訊資料流為老師講解的影片資料 Video1(V_{11})與相關課程的影片資料 Video2(V_{21} 、 V_{22} 、 V_{23})，一道投影片資料流播放課程相關的投影片資料(I_{11} 、 I_{12} 、 I_{13})，一道文字資料流(T_{11} 、 T_{12} 、 T_{13} 、 T_{14} 、 T_{15} 、 T_{16})是相關課程章節的說明，Audio 設定為共享媒體，影片資料 Video1 設定為主要媒體，聲音資料 Audio、影片資料 Video2、投影片資料與文字資料都必須在適當的時間點與老師講解的聲音資料同步，其中影片資料流 Video1、Video2 與聲音資料流 Audio 屬於 fine-grained 同步，其它資料流與聲音資料流屬於 coarse-grained 同步，各種媒體資料之間的同步關係如圖 16 所示。

(二) 媒體節目製作實作

媒體節目製作部分，包含五道媒體資料流的媒體資料的製作，以及媒體特性資訊的取得，並且將各種媒體依彼此的時序關係以「時序語言(TS)」描述完成，然後傳給時空映射機制，依同步容錯度與相依度，將媒體組合成複合物件，具有相類似同步特性的複合物件集中安置於滿足該物件同步需求特性的一個子伺服器上。另外還必須執行時空轉換的步驟，以 master 媒體(Video 1)為基礎來衡量，將時序關係的計算以畫面張數(frames)表示法轉換成以資料累計量(blocks)表示法，才能讓伺服器端正確的達到資料擺放的目的，在讀取影片時，也才能在準確的時間內讀取媒體資料。最後將這種複合物件對應於儲存位置的訊息，以「空間語言(LS)」描述完成，傳送至伺服器端進行實際的儲存，當伺服器端讀取複合物件時，伺服器端的「Channel Oriented Mechanism」會依既訂的時序關係預留適當的暫存區，啟動讀取資料，達到預期的同步效果，減輕在客戶端完成最終同步的負擔，如圖 17 所示。

傳送到媒體伺服器的 LS 包含三項資訊，分別介紹如下(如圖 18)：

- (1) SynDocInfo：記錄整個複合式文件中所有複合物件的資訊，包括每個複合物件中媒體的個數、最小同步容錯度(min ST)、複合物件的名稱等資訊。
- (2) ComObjInfo：依照複合物件的次序，記錄每個複合物件中所有媒體資料的資訊，包括媒體資料與 master media 的同步播放時間、資料流編號、傳輸速率與媒體名稱等資訊。
- (3) LocationStrip：記錄複合物件之間的驅動時間，包括複合物件與 master 複合物件之間的同步點、複合物件的編號、複合物件的名稱等資訊。

(三) 媒體節目播放子系統實作

在客戶端的媒體節目播放子系統中，使用多執行緒(Multi-thread)來執行接收節目媒體資料與播放媒體資料的工作，多執行緒可以在系統接收網路的資料時，同時執行其它程式，且程式之間不會相互影響。在本例中包含五道資料流，因此使用五個 thread 來控制從網路端接收媒體資料，另外使用五個 thread 來控制播放媒體資料，媒體節目播放子系統中各個模組的關係如圖 19 所示。

在本系統中，我們依照節目媒體資料需要的通道數目建立相對應個數的 data socket 與網路執行緒，當每個 data socket 接收到伺服器端傳過來的媒體資料時，即呼叫 PostThreadMessage() 函式，執行相對應的網路執行緒，首先設定每次接收 64KBytes 的網路資料至暫存區，然後呼叫緩衝區管理的 PutIn 函式將暫存區內的資料寫入緩衝區陣列中。

緩衝區管理負責建立緩衝區與處理資料的寫入與讀出，共建立 200 個單位的陣列 BufArray，每個陣列分為三個部份，NodeBody 主要放置媒體資料的部份，資料大小為 64KBytes，前端是 DataHead，負責記錄讀出資料時此陣列 BufArray 尚未使用的資料啟始點，後端是 DataTail，負責記錄寫入資料時此陣列 BufArray 尚未使用的資料結尾點。緩衝區管理中有兩個函數 PutIn 與 TakeOut，PutIn 函數負責將接收到的資料放入緩衝區陣列的 NodeBody 中；TakeOut 函數負責將資料從緩衝區陣列的 NodeBody 中讀出來播放。

當媒體節目播放子系統收到媒體節目管理子系統傳送過來的客戶編號時，媒體節目播放子系統的 data socket 同樣收到管理者端傳送過來的節目媒體同步資訊，內容包括：欲播放的節目中媒體類型、媒體發生的事件為媒體啟

始或媒體結束、發生事件的時間、媒體傳送使用的通道編號。當系統開始接收節目媒體資料後啟動播放排程機制(Actor)，開始計算播放節目媒體資料所需的啟始延遲，在啟始延遲時間到達時啟動播放執行緒開始播放節目。

播放的執行緒分為三個部分，第一部分 WM_DXINITIAL 主要負責將媒體資料從緩衝區中讀出，處理媒體資料的初始化設定，以及設定媒體播放狀態為 PLAYING，在視訊與音訊方面透過 OpenStream 函式呼叫 DirectX Media 的 DirectShow 播放，在圖片方面透過 VC++ 的 Cpalette 類別將圖片資料處理後呈現在使用者的螢幕畫面上，在文字方面直接從緩衝區將文字資料取出顯示在使用者螢幕畫面的文字框內。第二部分 WM_DXMONITOR 負責監督媒體資料的接收速度與播放速度，也就是緩衝區的流量控制機制，避免緩衝區內媒體資料 overflow 或 underflow。第三部分 WM_DXEND 負責結束媒體資料流的播放，例如視訊與音訊結束需要關閉 DirectShow 播放，並設定狀態為 CLOSE。播放執行緒由播放排程機制啟動計時器，在媒體節目資料劇本安排的播放時間到達時，呼叫 PostThreadMessage() 函式執行相關媒體的播放執行緒。

(四) 媒體節目播放子系統播放畫面

媒體節目播放的介面如圖 20 所示，右邊顯示目前管理者端預訂播放的節目表單，供使用者下次選擇參考，左邊播放兩道視訊資料流，分別是老師上課講解的影像(Video1)與課程相關的影片(Video2)，中間由上至下分別是課程內容的簡述(Text)、課程相關的投影片(Slide)與老師講解的聲音(Audio)，聲音部分可以選擇開啟【Open】或關閉【Close】，Video1、Video2 與 Audio 屬於動態媒體，它們右邊的黑色框框中出現的數字代表目前每個緩衝區內的資料量，每單位代表 64KBytes，43 個單位代表目前的資料量為 2752KBytes。最下面的狀態欄顯示目前媒體資料流的狀態資訊，右下角的【Connect】按鍵是連接到管理者端要求連線並接收節目表單，【Exit】按鍵是通知管理者端結束連線並結束媒體節目播放子系統。

五、結論

在設計媒體伺服器時，為了要提升磁碟機的效能，常使用資料輪置的技術來存取資料，而導致相當長的啟始延遲，為了達到媒體間的時序同步，節省暫存區資源，我們提出時空映射機制，以圖形化的方式來分析複合式文件中媒體之間的同步關係，將可以合併儲存的媒體合併為複合物件儲存，最後將分析完的資訊以

空間語言表示，傳送至媒體伺服器進行實際媒體資料的擺放，透過時空映射機制的分析結果，從伺服器中存取媒體資料時，共享的媒體可以被重複啟動來節省磁碟空間，同時在暫存區與磁碟機效率間達成平衡，並且達到媒體間同步的目標。

在系統實作部分，將本研究提出的時空映射機制運用在複合式文件的隨選視訊系統上，除了修改原有的媒體伺服器之外，另外設計了媒體節目管理子系統與媒體節目播放子系統，主要功能就是在播放複合式文件時達到媒體協奏的同步。我們將時空映射機制實作在媒體節目管理子系統中，預先根據節目劇本將媒體之間的時序關係依時序語言描述，然後進行時空映射機制，達到媒體間的同步，簡化媒體伺服器在讀取資料時處理同步問題的時間。媒體伺服器依照時空映射機制的分析結果擺放媒體資料，讀取時依照媒體之間的時序關係讀出，媒體節目播放子系統除了接收與播放媒體資料外，另外設計播放排程機制，再進行一次媒體間細微的同步(fine-grained sync.)，讓媒體在客戶端的播放更加順暢，播放品質更能讓使用者所接受。

六、誌謝

本論文為行政院國家科學委員會補助之專題研究計劃 NSC 89-2219-E-020-001 的部分成果，僅此誌謝。

七、參考文獻

Sheau-Ru Tong, Liu-Yih Lin, "Design of Media Servers: A Storage/Channel Management Mechanism for Supporting Media Orchestration in Synthetic Documents", Workshop on the 21st Century Digital Life and Internet Technologies, Taiwan, 2001.

- [1]. 童曉儒，林柳義，“媒體伺服器之設計—支援複合式文件中媒體協奏的儲存 / 頻道管理機制”，二十一世紀數位生活與網際網路科技研討會，2001.
- [2]. 童曉儒，賴文蘋，“媒體伺服器之設計—支援複合式文件中媒體協奏的同步管理機制”，二十一世紀數位生活與網際網路科技研討會，2001.
- [3]. 賴文蘋，“媒體伺服器之設計與製作—支援複合式文件中媒體協奏的同步管理機制”，國立屏東科技大學資訊管理系碩士班，碩士論文，2001.
- [4]. Gall D., “MPEG: A Video Compression Standard For Multimedia Applications”,

Communications of the ACM, Vol. 34, no. 4, pp. 46-58, April 1991.

- [5]. D. J. Gemell, H. M. Vin, D. D. Kandlur, P. V. Rangan, L. A. Rowe, “Multimedia Storage Servers: A Tutorial”, IEEE Computer, pp.40-50, May 1995.
- [6]. S. Ghandeharizadeh, S. H. Kim, and C. Shahabi, “Continuous Display of Video Objects Using Multi-Zone Disks”, Technical Report, UCLA, April 12, 1995.
- [7]. Chung-Ming Huang and Hsu-Yang Kung, “Mcast: A Multicast Multimedia Communication Software Development Platform”, Accepted by Lecture Notes in Computer Science, June, 2000.
- [8]. C.L.Liu, H.C. Du, M. Lin, “Performance of a Storage System for Supporting Different Video Types and Qualities”, IEEE JSAC, special issue of distributed multimedia systems and technology, 1996.
- [9]. McManus J and Ross K, “Video-on-Demand Over ATM: Constant-Rate Transmission and Transport”, IEEE Journal on Selection Areas in Communications, Vol. 14, no. 6, pp. 1087-1098, 1996.
- [10]. Minglu Li, Yongquang Sun, and Huanyue Sheng, “Temporal Relations In Multimedia Systems, Comput. & Graphics”, Vol. 21, no. 3, pp. 315-320, 1997.
- [11]. N.U. Qazi, M. Woo, and A. Ghafoor, “A Synchronization and Communication Model for Distributed Multimedia Objects”, Proc. First ACM Conf. On Multimedia, ACM Press, New York, pp. 147-156, Aug. 1993.
- [12]. Lixic Zhang, Stephen Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala, “RSVP: A New Resource ReSerVation Protocol”, IEEE Network, pp. 8-18, September 1993.

八、相關圖表

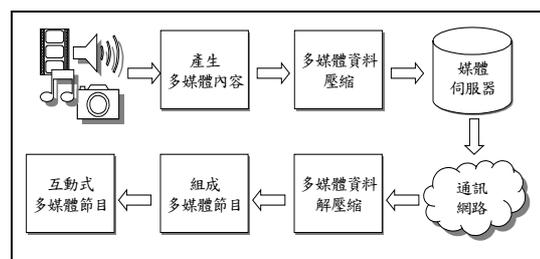


圖 1：多媒體的產生、壓縮、儲存、傳送與播放之流程

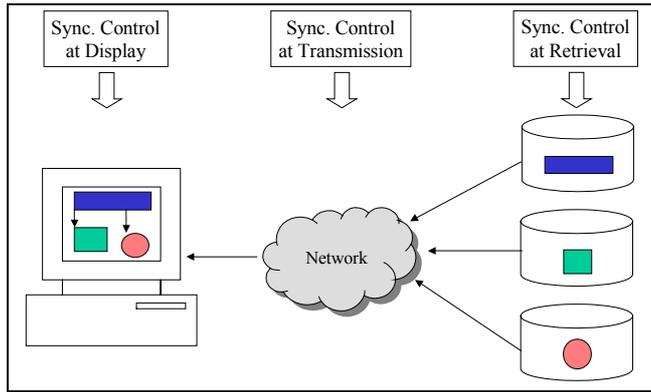


圖 2：分散式環境下各子系統的同步控制

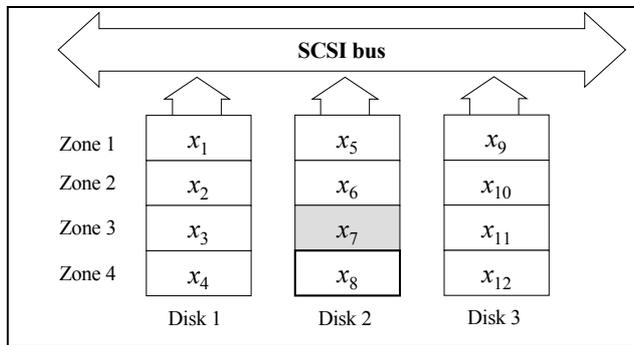


圖 3：資料輪置磁碟機組架構

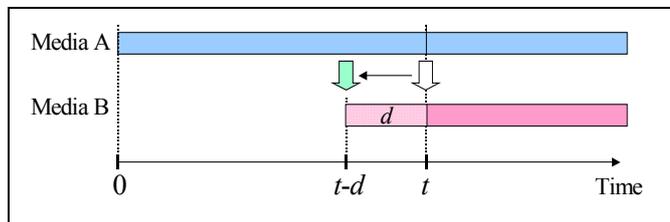


圖 4：啟始延遲與暫存區關係圖

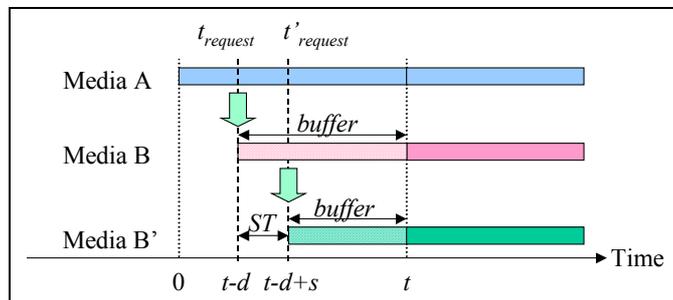


圖 5：暫存區與同步容錯度關係圖

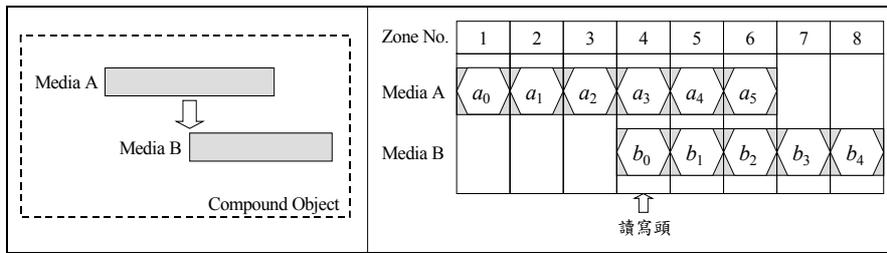


圖 6：複合物件儲存位址關係圖

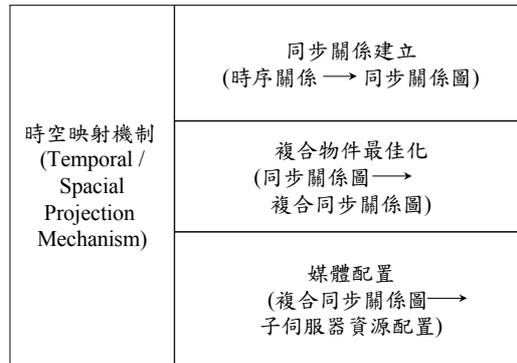


圖 7：時空映射機制的層次架構

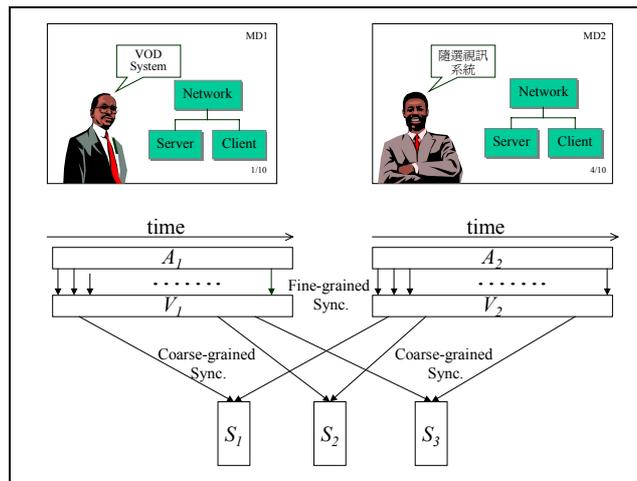


圖 8：兩個複合式文件的資料流與其間彼此的同步關係

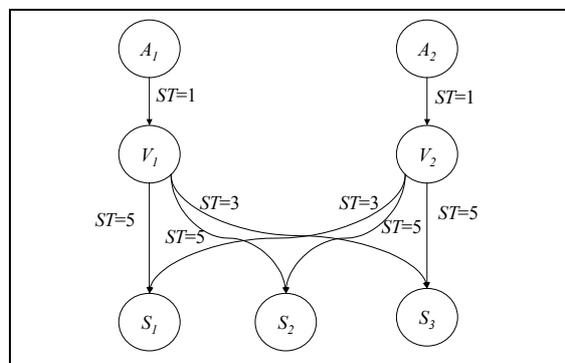


圖 9：對應圖 8 之同步關係圖

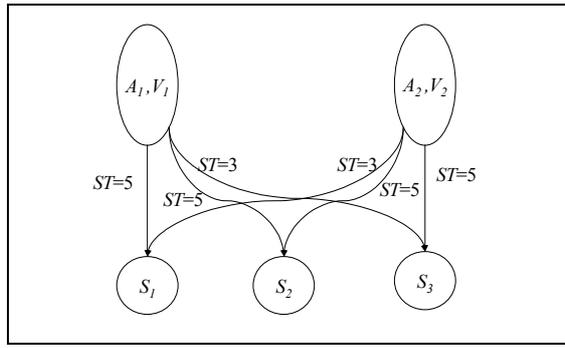


圖 10：自圖 9 所簡化之複合同步關係圖

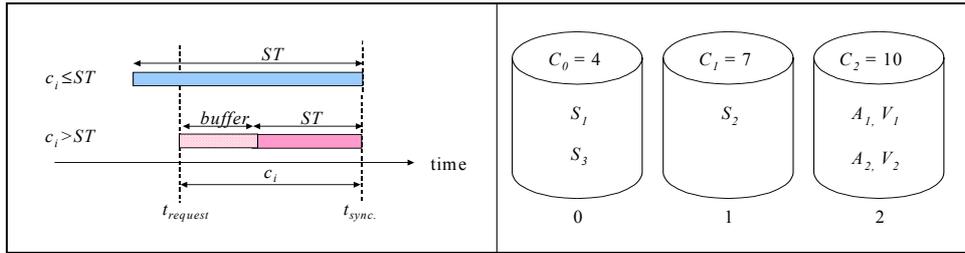


圖 11：子伺服器媒體配置圖

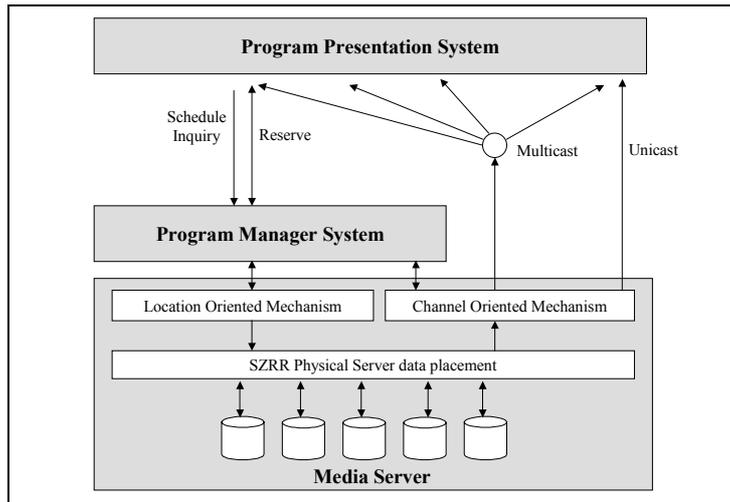


圖 12：複合式文件隨選系統子系統關係圖

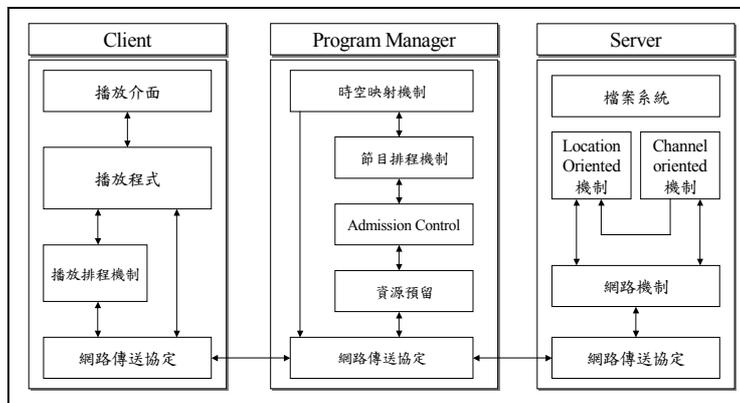


圖 13：複合式文件隨選系統子系統功能圖

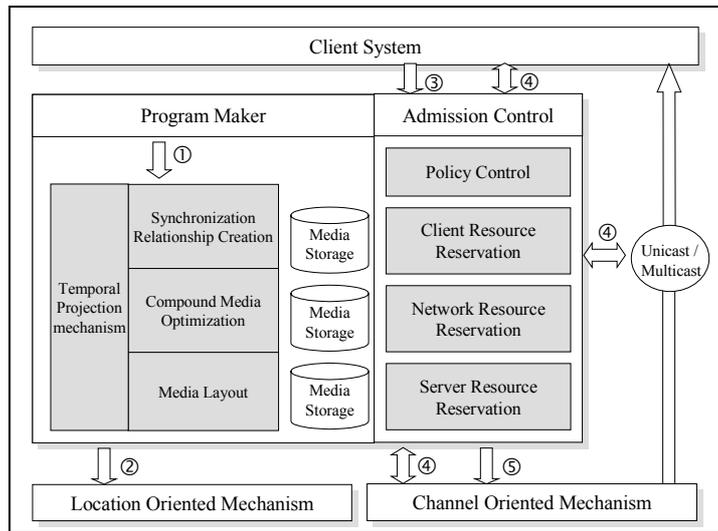


圖 14：媒體節目管理系統架構圖

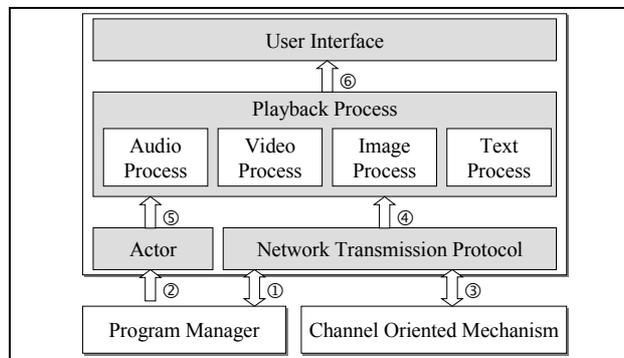


圖 15：媒體播放管理機制架構圖

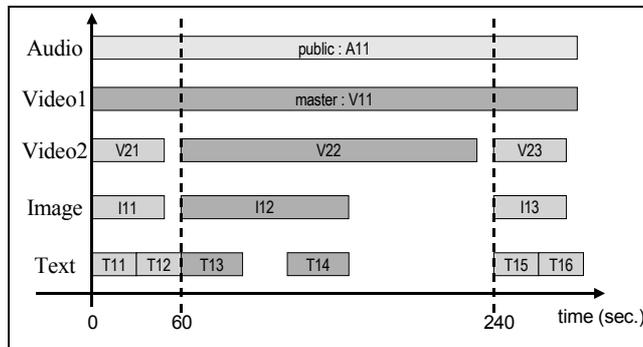


圖 16：媒體資料之間的同步關係圖

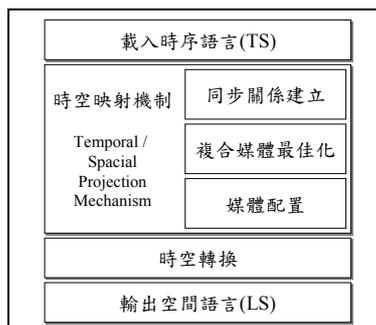


圖 17：媒體節目製作各模組關係圖

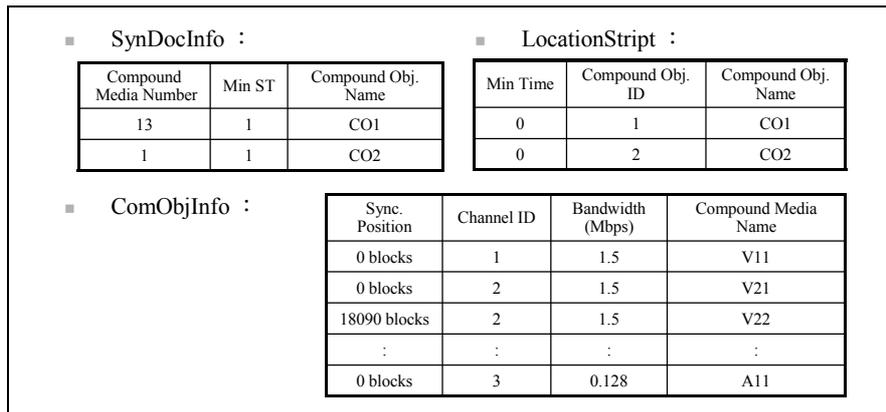


圖 18：空間語言資料結構

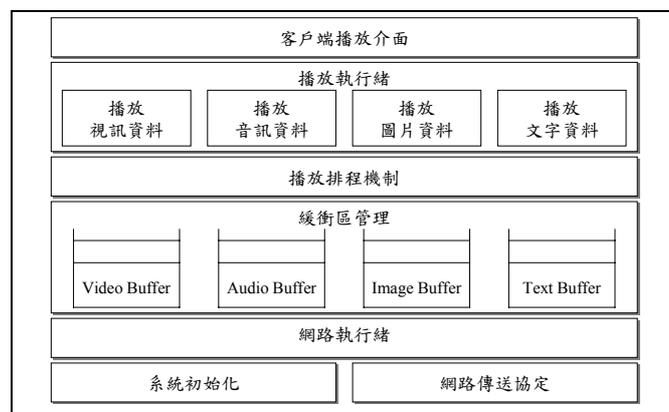


圖 19：媒體節目播放子系統各模組關係圖



圖 20：媒體節目播放子系統播放介面