

國立交通大學

多媒體工程研究所

碩 士 論 文

雲端服務視訊互助平台之訊息傳遞伺服器研究

The study of message server for
video based cloud remote service platform

研 究 生：陳冠宏

指 導 教 授：傅心家 教授

中 華 民 國 一 百 年 八 月

雲端服務視訊互助平台之訊息傳遞伺服器研究
The study of message server for video base cloud remote service platform

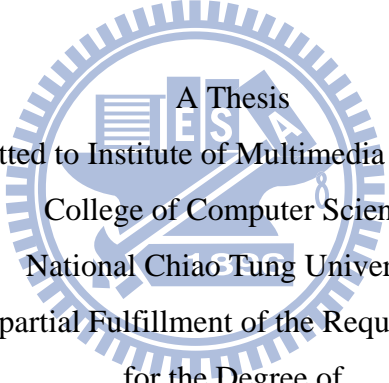
研究生：陳冠宏

Student : Guan-Hung Chen

指導教授：傅心家

Advisor : Prof. Hsin-Chia Fu

國立交通大學
多媒體工程研究所
碩士論文



A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science

August 2011

Hsinchu, Taiwan, Republic of China

中華民國一〇一年八月

雲端服務視訊互助平台之訊息傳遞伺服器研究

研究生：陳冠宏

指導教授：傅心家 教授

國立交通大學多媒體工程研究所

摘要

雲端服務視訊互助平台是一個以互助教學為目的之視訊連線平台，在此平台中使用者可透過各式多媒體資訊，與其他使用者進行與溝通。且該連線歷程將會被記錄下來供往後觀看。

本論文提出一個提供平台文字溝通功能之訊息傳遞系統，實作一訊息傳遞伺服器，將所收到之訊息轉送至其目的地，並將該訊息紀錄供後續處理使用。

同時研究如何讓提昇訊息傳遞伺服器的負載量與可靠度。藉由改進訊息傳遞伺服器之訊息處理流程、實施寫入策略來提升單台伺服器的負載量。實施配置策略與超載機制提升伺服器的可靠度。最後，將單台訊息傳遞伺服器架構擴充為多台訊息傳遞伺服器之分散式架構，更進一步提升雲端服務視訊互助平台之負載量。

The study of message server for video based cloud remote service platform

Student : Guan-Hung Chen

Advisor : Prof. Hsin-Chia Fu

Institute of Multimedia Engineering
National Chiao Tung University

Abstract

Video based cloud remote service platform is a video connection platform for mutual teaching. In this platform, the user can communicate with other users through connection that including of multimedia information like video, sounds, text, and painting marks. And the connection process was recorded for follow-up views.

This paper present a Message transferring system that providing communicating functions for Video based cloud remote service platform. Implementing a Message Server that forwarding the received messages to their destination, and recording those messages for follow-up treatment.

I also studied how to enhance the loading ability and reliability. Enhancing the loading ability of single Message Server by improving the flow of message processing and the writing policy. Enhancing the reliability by allocating policy and overloading mechanism. Finally, expending the single Message Server architecture to distributed multi- Message Server architecture for further enhancing loading ability.

誌謝

感謝傅老師的指導和照顧，承蒙老師給予學生的鼓勵與支持並幫助我找到研究方向，使得本論文能順利完成，在此謹向老師致上最誠摯之謝忱與敬意。同時，感謝實驗室博士班學長政龍，同學永彰、金昇，還有學弟祐豪、宜叡、醇豐平常在生活上以及學業上的建議與指教。最後，感謝我的家人對我的支持，讓我不必擔憂生活上的問題，專注在學業上得以順利完成學業

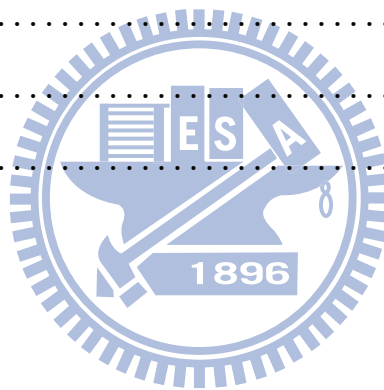


目錄

中文摘要.....	i
英文摘要.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vii
圖目錄.....	viii
第一章 緒論.....	1
1.1 研究背景.....	2
1.2 研究動機.....	2
1.3 研究目標.....	2
第二章 相關研究.....	3
2.1 現有視訊連線服務平台其文字溝通功能之介紹與探討.....	3
2.1.1 通訊軟體 Windows Live Messenger.....	3
2.1.2 通訊軟體 Skype.....	3
2.1.3 影音串流播放平台 own3D.tv.....	4
2.1.4 小結.....	4
2.2 萬人伺服器負載問題.....	5
2.2.1 萬人伺服器負載問題介紹.....	5
2.2.2 解決萬人伺服器負載問題之策略探討.....	5
第三章 系統架構與設計.....	11
3.1 雲端服務視訊互助平台設計理念.....	11
3.1.1 雲端服務視訊互助平台介紹.....	11

3.1.2	訊息傳遞伺服器介紹.....	12
3.1.2.1	訊息傳遞伺服器設計理念介紹.....	12
3.1.2.2	訊息傳遞伺服器之基本功能介紹.....	13
3.2	訊息傳遞伺服器設計.....	15
3.2.1	訊息介紹.....	15
3.1.2.1	訊息格式介紹.....	15
3.1.2.2	對話、標誌訊息介紹.....	16
3.1.2.3	特殊指令訊息介紹.....	17
3.2.2	訊息傳遞伺服器架構與訊息處理流程設計.....	18
3.2.2.1	訊息傳遞伺服器架構.....	18
3.2.2.2	訊息處理流程.....	21
3.3	訊息傳遞伺服器處理機制介紹.....	25
3.3.1	寫入策略.....	25
3.3.2	配置策略與超載機制.....	26
3.3.2.1	配置策略.....	26
3.3.2.2	超載機制.....	27
3.4	分散式訊息傳遞伺服器設計.....	28
3.4.1	分散架構.....	28
3.4.3	分散策略.....	31
3.4.3	故障處理.....	31
 第四章 系統測試		
4.1	雲端服務視訊平台整體測試.....	34
4.1.1	測試方法.....	35
4.1.2	測試環境.....	37
4.1.3	網頁伺服器傳送訊息至客戶端 Flash 測試.....	38
4.1.4	客戶端發送訊息要求其他客戶端接收影音串流測試.....	40

4.2	訊息傳遞伺服器測試軟體設計.....	42
4.2.1	雲端服務視訊互助平台中的使用者發送行為統計.....	42
4.2.2	訊息傳遞伺服器之測試軟體設計.....	43
4.3	訊息傳遞伺服器測試.....	44
4.3.1	伺服器負載能力測試.....	44
4.3.2	寫入策略對於訊息傳遞伺服器之訊息處理能力影響.....	55
4.3.3	分散架構對訊息傳遞伺服器服務人數之提升測試.....	61
4.3.4	分散架構下訊息傳遞伺服器之故障處理成功率.....	62
第五章	結論與未來工作.....	63
5.1	結論.....	63
5.2	未來展望.....	63
	參考文獻.....	64
	附錄.....	65



表目錄

表 3-1 : 訊息傳遞伺服器之基本功能介紹表.....	13
表 3-2 : 訊息傳遞伺服器之組件功能介紹表.....	19
表 3-3 : 何種條件下, 將會開始執行硬碟寫入相關動作之對應表.....	26
表 4-1 : Web、Media、Msg Server 之硬體配備列表.....	37
表 4-2 : web、media、msg server 的軟體版本.....	37
表 4-3 : 使用者發送訊息行為之記錄及統計結果.....	42
表 4-4 : 討論室內人數對於訊息傳遞伺服器處理能力影響之參數設定.....	45
表 4-5 : 討論室數量對於訊息傳遞伺服器處理能力影響之參數設定.....	45
表 4-6 : 使用者所送訊息長度對訊息傳遞伺服器處理能力影響之參數設定...	46
表 4-7 : 加權討論室之佔有率與推導出的伺服器處理能力對應表.....	52
表 4-8 : 寫入暫存佇列長度對於訊息傳遞伺服器處理能力影響之參數設定...	56
表 4-9 : 在同樣寫入暫存佇列長度時, 改變所要寫入的文字檔案數量時對於訊息傳遞伺服器處理能力影響之參數設定.....	57
表 4-10 : 寫入暫存佇列長度超過 100 後對訊息傳遞伺服器處理能力關係表..	59
表 4-11 : 故障處理測試之結果.....	62

圖目錄

圖 2-1 : 阻塞式 I/O 模型示意圖.....	6
圖 2-2 : 非阻塞式 I/O 模型示意圖.....	7
圖 2-3 : 多路復用式 I/O 模型示意圖.....	7
圖 2-4 : 信號驅動式 I/O 模型示意圖.....	8
圖 2-5 : 異步式 I/O 模型示意圖.....	9
圖 2-6 : 五種 I/O 模型流程之比較圖.....	9
圖 3-1 : 訊息傳遞伺服器為大量討論室之集合體示意圖.....	13
圖 3-2 : 影音連線頁面之聊天室與白板介面.....	17
圖 3-3 : 訊息傳遞伺服器架構與訊息流向圖.....	18
圖 3-4 : 各執行緒之狀態轉換示意圖.....	22
圖 3-5 : 接受連線.....	23
圖 3-6 : 訊息讀取.....	23
圖 3-7 : 網路傳輸.....	24
圖 3-8 : 硬碟寫入.....	24
圖 3-9 : 分散式 Msg server 架構示意圖.....	28
圖 3-10 : 開啟新的視訊連線活動時, 其過程中與 Msg server 相關之部分... ..	29
圖 3-11 : 圖 3-10 過程之分散式版本.....	30
圖 3-12 : 故障處理流程圖.....	32
圖 4-1 : 網頁伺服器傳遞訊息至客戶端 Flash 之流程圖.....	34
圖 4-2 : 客戶端發送訊息要求其他客戶端向影音伺服器接收影音串流.....	35
圖 4-3 : 使用者(測試人員)人數與網頁伺服器傳遞訊息至客戶端 Flash 耗費時間之關係圖.....	39
圖 4-4 : 使用者(模擬軟體)人數與網頁伺服器傳遞訊息至客戶端 Flash 耗費時間之關係圖.....	39
圖 4-5 : 使用者(測試人員)人數與客戶端發送訊息要求其他客戶端向影音伺服器接收影音串流耗費時間之關係圖.....	41
圖 4-6 : 使用者(模擬軟體)人數與客戶端發送訊息要求其他客戶端向影音伺服器接收影音串流耗費時間之關係圖.....	41

圖 4-7：在開啟 100 個討論室下，討論室內使用者人數對伺服器處理能力影響之關係曲線圖..... 47

圖 4-8：討論室數量對於訊息傳遞伺服器處理能力影響之關係曲線圖..... 49

圖 4-9：伺服器承載人數非均勻分布在討論室對於訊息傳遞伺服器處理能力影響之關係曲線圖..... 51

圖 4-10：在伺服器承載人數非均勻分布在討論室時，推導出的伺服器處理能力數據與實際數據之比較圖..... 53

圖 4-11：測試訊息長度對伺服器處理能力影響之關係曲線圖..... 54

圖 4-12：寫入暫存佇列長度對於訊息傳遞伺服器處理能力之關係曲線圖.... 58

圖 4-13：寫入的目標數量時對於訊息傳遞伺服器處理能力之關係曲線圖.... 60

圖 4-14：伺服器台數與分散式系統每秒最多所能處理之數量之關係圖..... 61



第一章 緒論

1.1 研究背景

日常生活中各種器具發生故障是常有的事，其中多數故障的排除經常是十分簡易且直覺的。但人們往往因為不熟悉修理流程而不敢動手，只能將故障的物品送回原廠由工程師進行處理，亦或請工程師到現場進行維修動作。不但維修過程曠日廢時，若是此物品已超出保固期，那麼尋求原廠的維修服務對使用者來說便是一筆額外的開銷。此時若是能在網路上尋求其他人所分享之維修教學文件來自行維修物品，既可省錢又十分方便。

但以維修教學文件來說，目前網路上之此類文件多是以圖片、文字類型來呈現，這種方式的問題在於資訊的交流通常是單方向，也就是僅有教學者（撰寫文件者方）的資訊會給學習者（尋求教學者方）看到，而學習者的狀況卻很難、甚至是無法讓教學者得知。

若是能讓學習者與教學者進行視訊連線，將學習者處的問題與狀況透過影像與聲音傳遞給教學者，教學者依收到的資訊對學習者發出指示。而除了影像和聲音之外，再額外搭配文字溝通與畫筆標誌系統讓兩者的資訊交流隔閡進一步降低，如此必可更完善的解決學習者的問題。此外，將整個教學的過程記錄下來，供往後有相同問題的使用者觀看，避免同一個問題被重複解答的情況發生。

因此，我們希望能建立一個以維修教學為出發點的雲端服務視訊互助平台，提供兩大功能：一、協助學習者找尋適當的協助對象，所有想尋求協助或想幫忙他人的使用者皆可透過此平台進行配對。二、提供使用者能透過影像、聲音、文字與他人進行連線，並能在連線時使用畫筆標誌系統協助教學引導。每次視訊連線後將其影像、聲音、文字溝通、畫筆標誌等紀錄進行保存。

1.2 研究動機

在互助服務的過程中，各使用者間明確的溝通是最為重要的。以雲端服務視訊互助平台上的影音連線活動為例，影像與聲音的傳輸是透過影音伺服器(將在章節3.1.1中作介紹)，但如果只有影像及聲音的互動，在溝通時還是會有語意不清或是重點模糊的狀況。因此我們決定除了影像和聲音之外，再增加文字溝通的功能，文字溝通的好處是：有慎重的語意表達、可重複詳細觀看訊息、可多國語文互譯等。

訊息傳遞伺服器便是為了替平台中各個伺服器與使用者傳遞訊息而誕生，除了文字溝通訊息以外，畫筆標誌系統的各式資訊、網頁伺服器對使用者所發出的各式指令訊息也是透過訊息傳遞伺服器來傳輸。

1.3 研究目標

本研究主要著重在雲端服務視訊互助平台中訊息傳遞伺服器 (Message Server)，設計出符合平台各式需求且高效能之訊息傳遞伺服器，此伺服器欲達到以下目標：

- (1) 單台伺服器能同時服務數以千計，甚至萬計之使用者。
- (2) 迅速、流暢的訊息傳遞服務。
- (3) 儲存影音連線活動中有意義之各式訊息，例如文字溝通、影像標註等訊息，以供往後使用者重播觀看。
- (4) 可擴充的分散式伺服器架構，可藉由擴充伺服器個數來因應平台使用者人數的成長。

第二章 相關研究

2.1 現有視訊連線服務平台其文字溝通功能之介紹與探討

本節將介紹數種視訊連線服務相關的軟體及網站，觀察並探討其文字溝通功能之實作。

2.1.1 通訊軟體 Windows Live Messenger (MSN Messenger)

Windows Live Messenger 是最多人使用的即時通訊軟體之一，現時的版本除了有基本的即時文字通訊之外還支援視訊連線(雙向)、檔案傳輸、即時語音交談、多人會議、連線遊戲等等延伸功能。而一般使用者最常使用的為及時文字通訊功能。

Windows Live Messenger 的文字溝通功能為 Client-Server 之架構，使用者們在進行文字溝通前必須先互相加為好友，透過伺服器找到對方並進行文字訊息交換，而其文字溝通紀錄僅儲存在客戶端，並不會在伺服器端保留。



2.1.2 通訊軟體 Skype

Skype 是支援語音通訊的即時通訊軟體，提供了視訊連線(雙向)、即時語音交談、即時文字通訊等多樣功能。絕大多數的Skype 使用者都是使用其即時語音交談與視訊連線功能，即時文字通訊功能則極少使用。

Skype 整體為P2P 架構，其中並沒有中央伺服器的概念，而是由許多超節點與客戶端組成的全球索引伺服器擔當(超節點為特殊的客戶端)，每一個Skype使用者皆連向鄰近的超節點，而超節點彼此之間會交換資訊。當使用者A 想要與使用者B 溝通時，需透過掌管A 之超節點向其他的超節點詢問使用者B 的位置，再由掌管使用者B 之超節點通知使用者A 之超節點，使用者A 與使用者B 方得以進行溝通。而其文字溝通紀錄僅儲存在客戶端，並不會在伺服器端保留。

2.1.3 影音串流播放平台 own3D.tv

own3D.tv 為一個通過網路進行視訊廣播的網站平台，以網頁的形式提供使用者實況轉播(單向)、錄影重播、與即時文字通訊等功能。使用者透過own3D.tv 之首頁挑選欲觀看之實況轉播頻道，進入該頻道開始觀看並加入頻道附屬之討論室與其他在同一頻道內之使用者進行文溝通。

其即時文字通訊功能之實作遵循IRC protocol，使用者透過own3D.tv 網頁內嵌之IRC client 連上特定之IRC server，並進入欲使用的頻道開始與頻道內的使用者進行文字溝通，而每個實況廣播皆擁有自己專屬之頻道，使用者們可以在觀賞同一份實況時彼此進行文字溝通。IRC 基本上不會儲存任何文字溝通紀錄，但若使用者需要的話可在IRC client 端進行儲存。

2.1.4 小結

以上三者雖皆為可靠且具擴展性之商用系統，但其功能取向卻和雲端服務視訊互助平台所需有所出入。首先互助平台需要能夠儲存每次活動的所有對話以及畫筆標註記錄，以上三者在這方面都只能由參與的客戶端而非伺服器進行紀錄，不僅影響到紀錄的可信度之外，同時也容易因為網路延遲等因素引發記錄不一致的問題，IRC 在此問題上特別嚴重。再者就算嘗試於此三現存商用系統之伺服器上增添訊息紀錄功能，在MSN 與Skype 上會遇到系統閉源而無法直接對伺服器進行修改的問題。其三，互助平台由於具有畫筆標誌功能，每次畫筆標誌將會造成該討論室具有突發性的大量資料流動，其負載的特徵和上面所提的三種系統有不少差別。綜合以上原因使得互助平台之訊息傳遞伺服器需自行研發而無法使用現有系統。

2.2 萬人伺服器負載問題

2.2.1 萬人伺服器負載問題介紹

萬人伺服器負載問題(C10K problem)，指的是當伺服器在處理數以萬計的客戶端連結時，往往出現效率低下甚至完全癱瘓之情形。其特點為，設計不夠良好的伺服器程式，其吞吐量與硬體性能的關係往往是非線性成長。這是由於在處理客戶端連結之策略不當時，處理單個客戶端連結的資源消耗與當前連接數的關係為 $O(n)$ 而非常數關係，因此在同時服務的客戶端數目達到數以萬計時會造成相當可觀之資源消耗，導致伺服器之吞吐量無法與其硬體性能匹配。

2.2.2 解決萬人伺服器負載問題之策略探討

此種問題之應對需改變伺服器對客戶端服務的策略，主要分成兩個重點，重點一、伺服器程式以何種方式和作業系統合作，獲取I/O 事件並調度各傳輸端口(socket)上的I/O 操作。重點二、伺服器程式以何種方式處理任務和程序(process)/線程(thread)的關係。

以重點一來說，可分成阻塞式I/O、非阻塞式I/O、多路復用式I/O、信號驅動式I/O、異步式I/O 五種模型，以下將略作說明。圖2-1 到2-5 為重點一中使用上述五種模型之伺服器在接收由客戶端而來的訊息時，其調用作業系統的示意圖。圖左側為伺服器程序端，右側為伺服器作業系統之內核(kernel)端。

1. 阻塞式I/O 模型

如圖2-1所示，伺服器程序呼叫recvfrom函式向內核發送system call，內核進入wait for data狀態，等待客戶端發送訊息。當內核收到客戶端發送之訊息後進入copy data階段，開始將收到的訊息回傳給伺服器程序，完成後通知伺服器程序，整個過程到此結束。

在整個過程中，伺服器程序都會阻塞住，無法處理其他的工作。當要處理多數客戶端時，就必須使用大量的線程來服務，是五種模型中效率最低的。

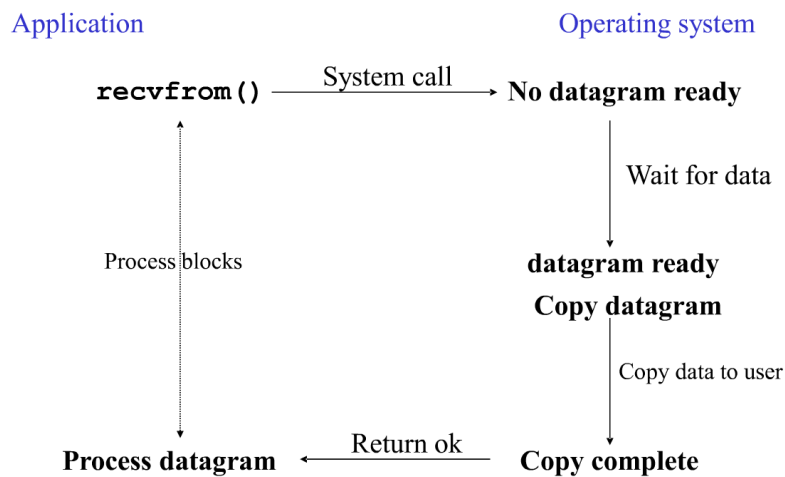


圖2-1: 阻塞式I/O模型示意圖

2. 非阻塞式I/O模型

如圖2-2所示，伺服器程序呼叫recvfrom函式向內核發送system call，若內核尚未收到客戶端發送之訊息便立刻回傳EWOULDBLOCK訊息。這個過程會不斷重複直到當kernel收到客戶端發送之訊息，進入copy data階段開始將收到的訊息回傳給伺服器程序，最後通知伺服器程序。在進入copy data階段前不斷重複詢問內核之行為我們稱作輪詢(Polling)。

在整個過程中，伺服器程序只有在copy data的階段會阻塞住，直到其收到內核的通知為止。在效能方面，雖說少了wait for data階段的阻塞，但須耗費大部分資源在輪詢的過程上，因此效能依舊不佳。

Application

Operating system

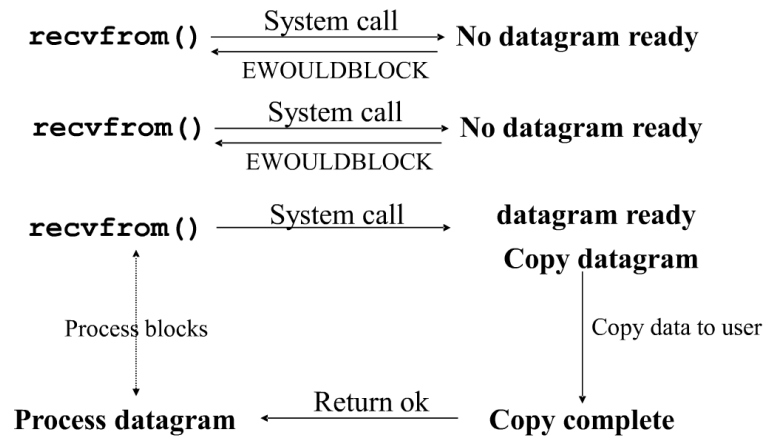


圖2-2: 非阻塞式I/O模型示意圖

3. 多路復用式I/O模型

如圖2-3所示，伺服器在wait for data階段使用select函式監控內核端是否有事件(代表收到訊息)發生。若有，則呼叫recvfrom函式進入copy data階段，完成後通知伺服器程序。

在整個過程中，如果內核中沒有訊息進入的話，select函式會是處於阻塞狀態，如此一來減少了在非阻塞式I/O模型中進行輪詢所耗費的資源，提升伺服器效能。

Application

Operating system

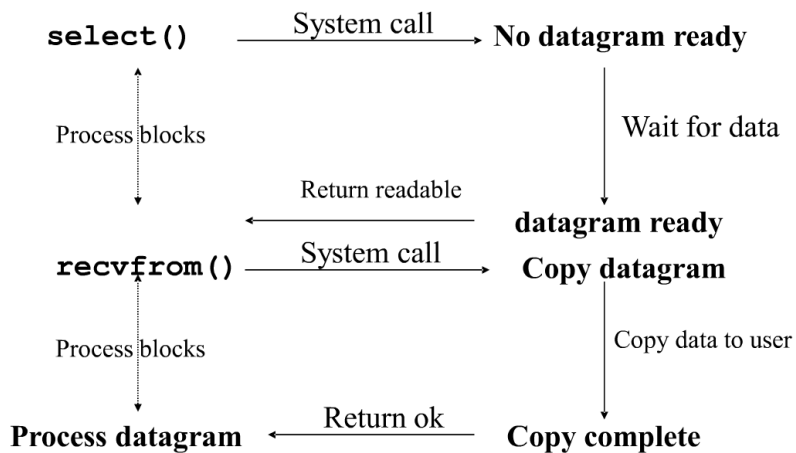


圖2-3: 多路復用式I/O模型示意圖

4. 信號驅動式I/O模型

如圖2-4所示，與多路復用式I/O相比，差異在於使用信號驅動式I/O(SIGIO) 信號處理器取代select函式。此處理器會向內核註冊事件，當socket收到訊息後，內核會主動通知伺服器程序調用recvfrom函式，進入copy data階段，完成後通知伺服器程序。

在整個過程中，wait for data階段時由於是由內核主動通知信號處理器，免去了多路復用式I/O模型中的阻塞與輪詢部分，大大的提升了伺服器效能。

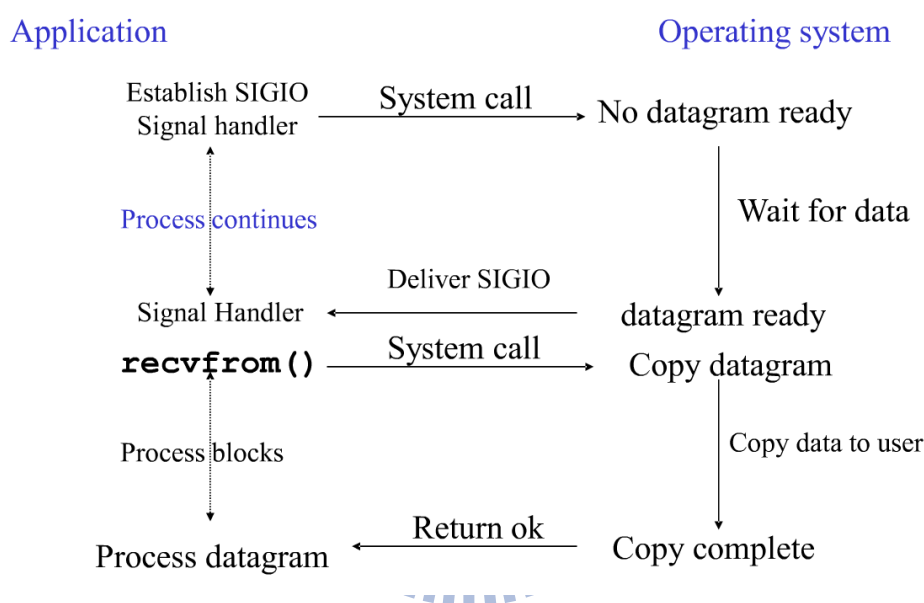


圖2-4: 信號驅動式I/O模型示意圖

5. 異步式I/O模型

如圖2-5所示，伺服器程序調用aio_read函式向內核發送system call後即返回處理其他事項。內核進入Wait for data與copy data狀態，處理完成後通知伺服器程序。

在整個過程中，伺服器程序均不會阻塞或是輪詢，因此效能是五種模型中最好的，但目前絕大部分linux系統並不支援此模型。

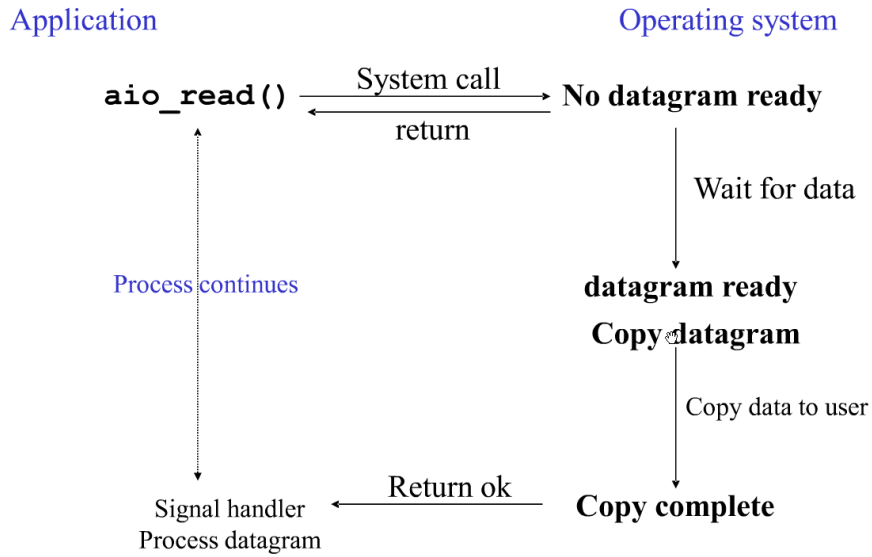


圖2-5: 異步式I/O模型示意圖

圖2-6為五種I/O模型流程之比較圖，可以看出越往後其阻塞越少，效能也就越高。

阻塞式I/O	非阻塞式I/O	多路復用式I/O	信號驅動式I/O	異步式I/O	
開始	確認 確認 確認 確認 確認	確認	Kernel通知	開始	} wait for data
阻塞	阻塞	阻塞 ↓ 收到訊息	開始	開始	
完成	完成	開始 阻塞 ↓ 完成	阻塞 ↓ 完成	通知完成	} Copy data from kernel

圖2-6: 五種I/O模型流程之比較圖

重點二的部分主要有每任務/程序、每任務/線程、多任務共享線程池等作法。而重點一與重點二的選項互相配對可以得出許多處理策略，常見的經典策略如下：

1. 一個線程服務一個客戶端，使用阻塞式 I/O

由於使用阻塞式 I/O，導致每個客戶端皆須要配置一個完整的線程持續等待其訊息進入，而作業系統處理數百個線程時存在一定的問題，例如：每個線程需要2MB stack，系統共有1GB的虛擬記憶體可供使用，而 $1\text{GB}/2\text{MB} = 512$ 個線程便佔用了所有可用的記憶體，導致效能低落。

2. 一個線程服務多個客戶端，使用多路復用式 I/O

伺服器程序持續監視kernel中有無注目的I/O事件其狀態發生改變，而在處理該事件I/O的過程中也會持續的監視並回報，因此此策略的效能與監視的事件數相關，也就是說在客戶端連結數上升時，伺服器之負擔也會隨之上升。

3. 一個線程服務多個客戶端，使用信號驅動式 I/O

伺服器程序向kernel註冊其所關心的事件，事件發生時kernel會自行通知伺服器程序。並且在該事件處理I/O的過程不會發出通知，僅在處理結束時發出，因此效能較2.好。

4. 一個線程服務多個客戶端，使用異步式I/O

程序對kernel發出I/O請求後便返回做其他工作，當kernel完成I/O操作後才對程序發出信號表示操作已完成。效能相當的優秀，但絕大部分linux的作業系統並不支援異步式I/O。

最後，在考量過雲端服務視訊互助平台之特性、需求、以及所開發之伺服器程式對各作業系統之移植可能後，決定以一個線程服務多個客戶端，使用信號驅動式 I/O 之策略模型來建構訊息傳遞伺服器。

第三章 系統架構與設計

3.1 雲端服務視訊互助平台設計理念

雲端服務視訊平台以互助教學為設計出發點，提供使用者們一個網路連線平台進行視訊互助教學。在雲端服務視訊平台中，每位使用者都可以是解決他人疑問的協助者或是發問請求者，透過此互助平台使用者將可以多元的形式尋求合適的引導教學。

當使用者有任何想維修的物品或想解決的難題時，可至本平台中搜尋過往的影音紀錄尋求解決範本，也可在平台中張貼文章請求其他使用者的協助解答。協助解答可以使用回覆文章的方式，或是直接邀請發問者進行視訊連線利用即時影音傳送與標誌輔助系統進行引導協助。而視訊連線的過程會由平台自動保存成影像紀錄，持有者可對其進行後製編輯供其他使用者觀摩學習。

為了降低使用者的操作門檻，雲端服務視訊平台採用網頁形式做為使用者操作介面，從平台的入口至視訊連線頁面均以網頁呈現，使用者只需有瀏覽器及 Flash 影音插件即可順利進行視訊連線等操作。

3.1.1 雲端服務視訊互助平台介紹

本平台之架構主要是由三種伺服器所組成，包含網頁伺服器、訊息傳遞伺服器及影音串流伺服器，以 Client-Server 的形式服務使用者，每種伺服器皆有其功能與目的，以下將詳細說明：

網頁伺服器 (Web server)

作為入口平台使用，採用 Nginx 伺服器軟體與 php、Javascript 程式語言做為開發程式並搭配 Mysql 資料庫管理使用者資訊、連線紀錄等相關資訊，在視訊連線中扮演的是前導者的角色，當使用者欲進行連線時，負責溝通聯繫訊息傳遞伺服器、影音串流伺服器進行運作。

影音伺服器 (Media server)


在視訊連線方面平台採用 Client-Server 架構，由影音伺服器負責視訊連線時的影音傳遞，當使用者進行影音連線時主要由處在客戶端之 Flash 程式，對影音伺服器進行影音收發動作。影音伺服器除了影音接收傳遞外，也提供使用者播放影片記錄的服務。

訊息傳遞伺服器 (Message server)

訊息傳遞伺服器在整個互助平台中負責扮演訊息傳遞中心的角色。訊息傳遞伺服器主要是以自行開發之 Java 程式做為伺服器端的處理程式，透過網路接受來自使用者、網頁伺服器與影音串流伺服器的訊息並轉送到適當的目的地，並且在伺服器端記錄其文字溝通、影音標註歷程等資訊，以供後續觀看使用。

3.1.2 訊息傳遞伺服器介紹

3.1.2.1 訊息傳遞伺服器設計理念介紹



訊息傳遞伺服器為雲端服務視訊互助平台提供訊息傳遞的功能，負責視訊影音活動時的各式功能的訊息傳遞，例如文字溝通、畫面標誌系統、攝影機切換功能等等，伺服器在接受各式訊息的同時會將訊息分析並執行該訊息所要求的動作，使視訊影音活動能順利進行。由於上述之各功能對使用者來說皆有強烈的即時性需求，因此訊息傳遞伺服器必須要能夠快速、有效率的處理大量來自於使用者的訊息。

訊息傳遞伺服器可以想像為大量討論室的集合體，每個討論室內有許多使用者，訊息的傳遞以討論室為目的地，傳送給在此討論室的所有成員，不同討論室的訊息不會互相干擾。而訊息傳遞伺服器服務的對象包括雲端服務室訊互助平台的使用者以及平台內的各個伺服器。圖 3-1 為訊息傳遞伺服器為大量討論室之集合體示意圖。圖中的人像代表平台的使用者，各自加入不同的討論室進行訊息傳遞，而電腦圖像代表平台內之其他伺服器，透過訊息傳遞伺服器傳送訊息給討論

室內的成員。

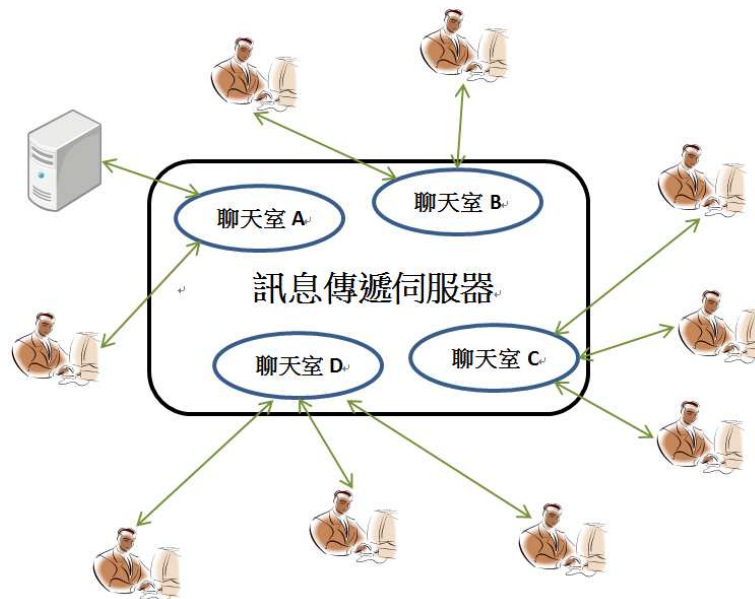


圖 3-1: 訊息傳遞伺服器為大量討論室之集合體示意圖

3.1.2.2 訊息傳遞伺服器之基本功能介紹

訊息傳遞伺服器所擁有之基本功能請見表 3-1，左側為基本功能名稱，右側為該基本功能之介紹。

表 3-1: 訊息傳遞伺服器之基本功能介紹表

功能名稱	介紹
開啟討論室	當平台中有新的影音連線活動產生時，開啟對應的討論室供活動中的使用者們使用。
關閉討論室	當平台中之影音連線活動結束時，關閉該活動之討論室，回收其空間與資源供往後的討論室使用。

<p>使用者 進入討論室</p>	<p>依據使用者所指定的討論室號碼將使用者加入至該討論室，該使用者即可透過討論室收發訊息。</p>
<p>使用者 離開討論室</p>	<p>使用者主動離開影音連線活動、或是與訊息傳遞伺服器斷線時，伺服器便將此使用者由討論室名單中移除，降低伺服器之負擔。</p>
<p>廣播訊息</p>	<p>使用者所發送的訊息，訊息傳遞伺服器依據其目的討論室，傳送給所有在此討論室之使用者。</p>
<p>儲存訊息</p>	<p>所有透過訊息傳遞伺服器傳遞之訊息，若其有儲存的必要，伺服器便將該訊息儲存至與其目的討論室對應的文件紀錄檔。而此文件紀錄檔可與影音伺服器所記錄之影音紀錄檔配合在平台上播放，完整重現此影音連線活動之歷程。</p>
<p>統計人數</p>	<p>統計在討論室內的使用者人數，此人數可在使用者之客戶端軟體顯示，亦可在平台的首頁上的影音連線活動告示板顯示，讓網頁的使用者了解該影音連線活動的熱門情形。</p>

3.2 訊息傳遞伺服器設計

訊息傳遞伺服器之設計目的為雲端服務視訊互助平台的伺服器群以及使用者提供快速、穩定的訊息傳遞功能，紀錄必要的文字溝通、標誌系統等訊息供往後播放使用，並且盡可能的提升同時能服務的最大使用者數。

訊息傳遞伺服器採用章節2-2所提到之一個線程服務多個客戶端，使用信號驅動I/O之模型，網路協定採用TCP/IP通訊協定。而為了讓訊息的轉傳過程能夠迅速完成，將處理一個訊息的流程拆分成數個部分，由不同的程序來處理，避免硬碟寫入等速度較慢的操作將整個處理時間拉長。此外，考慮到往後雲端服務視訊互助平台的使用者人數成長可能，訊息傳遞伺服器也包括了可擴充之分散式設計。



3.2.1 訊息介紹

訊息為訊息傳遞伺服器處理的基本單位，以下將介紹訊息的格式、種類。

3.2.1.1 訊息格式介紹

所有傳送至訊息傳遞伺服器的訊息，不論是來自使用者端或是平台內伺服器，均會包裝成固定的格式，用以讓訊息傳遞伺服器端辨別該訊息的目的地、欲執行動作、發送者等等資訊。格式範例如下：

標籤；目的地；發送者 ID；訊息內容

標籤代表此訊息想要執行的動作，例如開啟/關閉討論室、廣播訊息給此討論室的所有使用者等等。

目的地即為討論室的號碼，隨標籤的不同可能代表要關閉討論室或是將此訊息廣播給此討論室內的所使用者。

發送者 ID 為發送此訊息之使用者 ID 或是平台上其他伺服器代號，以供訊息傳遞伺服器保存使用者之狀態，可在後續處理使用。

訊息內容依照標籤而改變，可能為使用者所輸入之對話訊息或是標誌訊息、或是切換攝影機指令、切換討論室管理者的通知等等。訊息傳遞伺服器基本上不會對這部分進行分析，僅執行轉送。

而訊息依照其用途可分成兩種，分別為對話畫筆訊息以及背景指令訊息。

3.2.1.2 對話、標誌訊息介紹

在影音連線時，為了滿足使用者與他人的溝通需求，影音連線頁面提供了文字溝通介面以及畫面標誌介面，分別對應至對話訊息以及標誌訊息，如圖 3-2 所示。使用者在文字溝通介面以及畫面標誌介面中所輸入的對話、各式圖案等，透過影音連線頁面的編碼程序轉換成符合訊息傳遞伺服器格式之訊息，而後傳送至訊息傳遞伺服器再轉傳給另一端的使用者影音連線頁面，由解碼程序解析出原本的對話及圖案後顯示在介面上。

所有的對話、標誌訊息均會由訊息傳遞伺服器依照時間順序記錄，往後在播放此筆影音記錄時，其對話畫筆資訊也會隨時間出現配合影像，讓使用者能觀看到最完整之影音連線歷程。

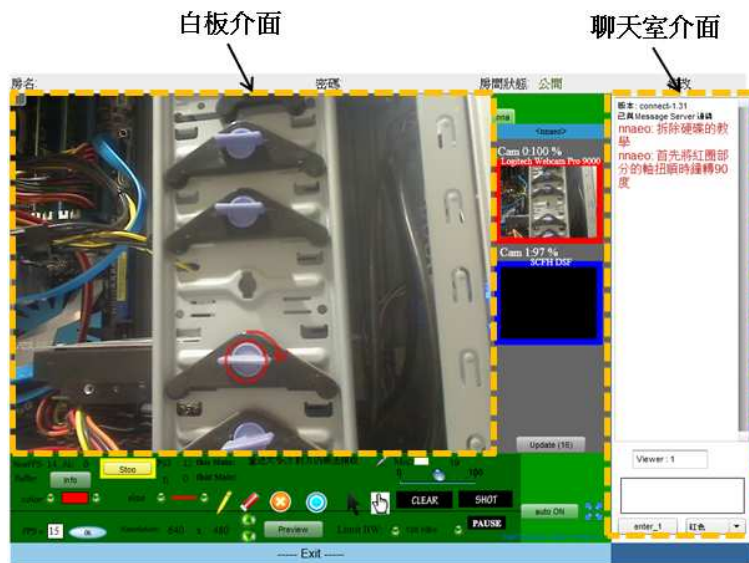


圖 3-2: 影音連線頁面之聊天室與白板介面

3.2.1.3 特殊指令訊息介紹

在影音連線活動進行時，除了會在介面中顯示對話、標誌訊息之外，同時也有許多使用者看不到的指令訊息透過訊息傳遞伺服器進行傳遞。由於顯示特性的不同，對話、標誌訊息以外的訊息我們稱為特殊指令訊息，特殊指定訊息的發送者與發送的對象可為一般的使用者也可為伺服器端，特殊指令訊息代表了一個特殊動作的發起，要求另一端的發送對象執行這個動作。而整個影音連線的運作便是由這些特殊動作組成。

例如，影音連線的開創者邀請了另一位使用者進行影音連線，該使用者成為升權旁觀者進入影音連線的通知便是由網頁伺服器發起，透過訊息傳遞伺服器廣播給影音連線內的所有使用者，使用者介面即新增升權旁觀者的攝影機切換介面供切換使用，為伺服器對使用者的模式。開啟討論室請求為伺服器對伺服器，攝影機的切換為使用者對使用者等等。

訊息傳遞伺服器通常不會儲存特殊指令訊息，除非該特殊指令訊息與往後的播放有關，在訊息格式中特別要求伺服器將其記錄下來。

3.2.2 訊息傳遞伺服器架構與訊息處理流程設計

3.2.1.1 訊息傳遞伺服器架構

訊息傳遞伺服器由許多不同功能的組件組成，圖 3-3 為訊息傳遞伺服器架構與訊息流向圖，每個組件的功能與訊息的流動將在表 3-2: 訊息傳遞伺服器之組件功能介紹表與訊息流向說明中介紹。

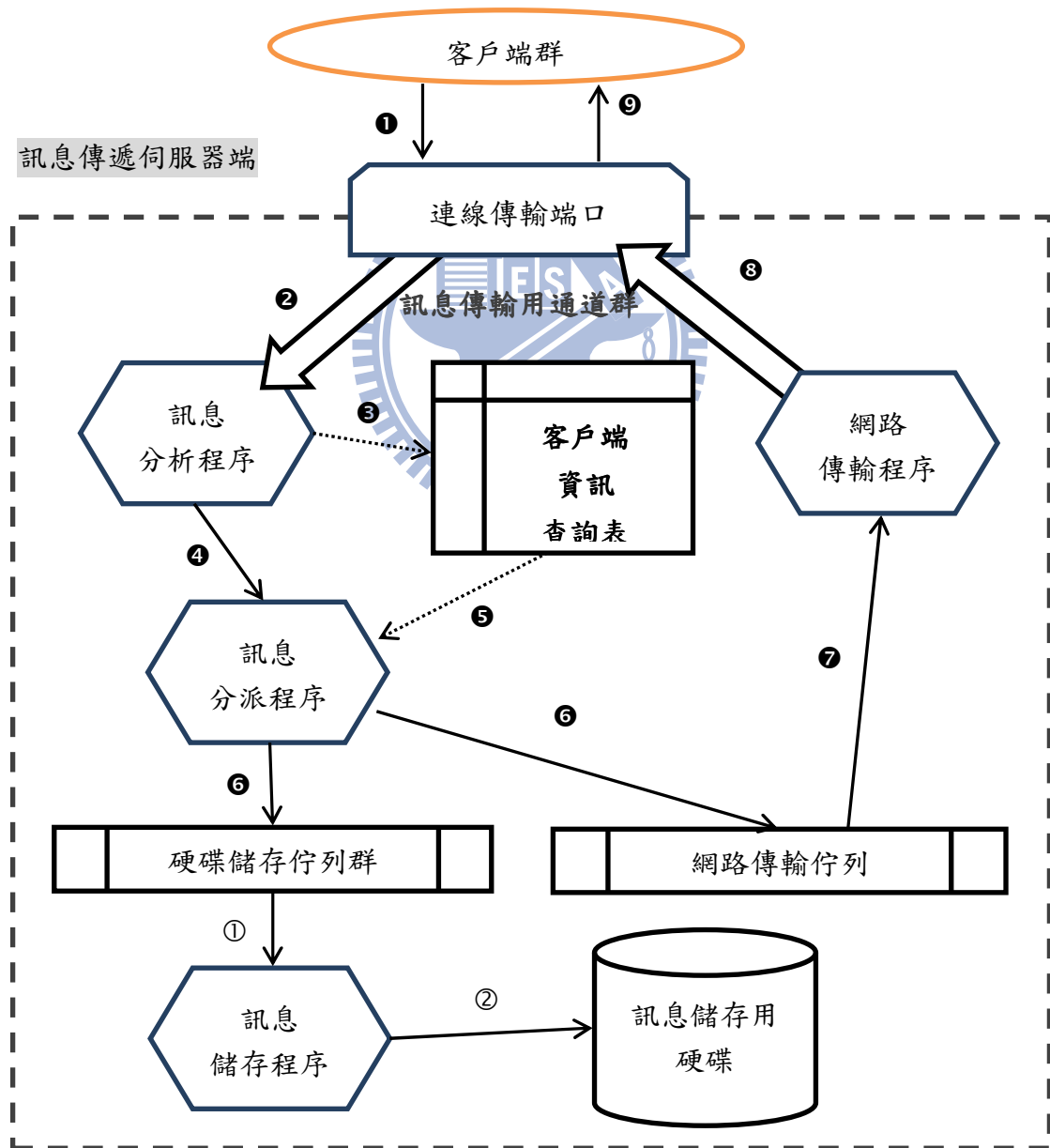


圖 3-3: 訊息傳遞伺服器架構與訊息流向圖

表 3-2 為訊息傳遞伺服器中各組件之功能介紹。

表 3-2: 訊息傳遞伺服器之組件功能介紹表

組件名稱	組件功能
連線傳輸端口	為伺服器的客戶端服務連接口，所有使用者之客戶端程式皆透過此端口與伺服器交換、傳輸訊息。
訊息傳輸用通道群	包含多條傳輸用通道，通道與客戶端為一對一對應，只要知道通道號碼便可以傳訊息給特定之客戶端。 (註:圖 3-2 為了理解方便將通道依方向分開，通道事實上無方向之分，收與送都是透過同一條。)
客戶端資訊查詢表	儲存 1. 討論室資訊(哪些客戶端在該討論室、人數等等)。 2. 通道資訊(每個客戶端對應至的通道號碼)。
訊息分析程序	1. 透過通道由連線傳輸端口讀取訊息。 2. 分析出該訊息包含之資訊並寫入客戶端資訊查詢表。
訊息分派程序	依照討論室號碼查看客戶端資訊查詢表，得知此訊息應當傳給哪些客戶端(也就是與客戶端對應的通道號碼)，將訊息與其通道號碼放入網路傳輸用佇列，以及依據討論室號碼將訊息放入該討論室專屬之硬碟儲存用佇列。
網路傳輸佇列	欲傳送給使用者之訊息暫存於此，等待網路傳輸程序處理。
硬碟寫入佇列群	每個佇列與討論室為一一對應，欲儲存至硬碟的訊息會暫存於此，等滿足一定條件時(寫入策略)才寫入硬碟。
網路傳輸程序	檢視網路傳輸用佇列，將其中的訊息依照對應的通道號碼填入寫入緩衝區預備傳送
訊息儲存程序	檢視硬碟儲存用佇列群，依照寫入策略決定哪個硬碟儲存用佇列內的訊息可以寫入硬碟並處理之。
訊息儲存用硬碟	將訊息依照房間號碼以及建立時間進行分類儲存之處。

註：寫入策略將在章節3.3.1 寫入策略中說明

接著是訊息流向說明，這邊說明單筆訊息在伺服器各組件中的流動過程。

訊息流向說明

- ❶ 使用者所輸入之對話、畫筆圖案或是切換攝影機等特殊動作，由客戶端程式包裝過後傳送至訊息傳遞伺服器之連線交換端口。
- ❷ 訊息分析程序透過訊息傳輸用通道，由連接傳輸端口讀入每一筆原始訊息，進行分析以取得該訊息欲進行之動作、目的地、對應的通道號碼等資訊，同時執行該訊息欲進行之動作。
- ❸ 將❷中分析出之資訊存入客戶端資訊查詢表。
- ❹ 訊息分析程序過後緊接著為訊息分派程序。
- ❺ 訊息分派程序查詢客戶端資訊查詢表，來決定此筆訊息該送給那些客戶端。
- ❻ 訊息分派程序將分析過後之訊息加上其目的地的訊息傳輸用通道編號，填入網路傳輸佇列。同時將分系過後之訊息依據討論室號碼填入對應之硬碟儲存佇列群。
- ❼ 網路傳輸程序不斷檢查網路傳輸佇列內是否有訊息，若有則將訊息自佇列內取出並開始處理
- ❽ 網路傳輸程序將訊息依照訊息傳輸用通道編號轉送至連接傳輸端口準備傳送。
- ❾ 訊息由連線交換端口透過網路傳送給指定之客戶端。
- ①② 儲存處理程序檢查硬碟寫入佇列群內的訊息狀態，當達到一定標準時便將其寫入訊息儲存用硬碟。

3.2.1.2 訊息處理流程

訊息傳遞伺服器在處理來自同一使用者之訊息的過程可以切割為四個狀態，分別為接受連線狀態、訊息讀取狀態、網路傳輸狀態、以及硬碟寫入狀態。其中接受連線狀態、訊息讀取狀態由訊息讀取執行緒，網路傳輸狀態由網路傳輸執行緒、硬碟寫入狀態由硬碟寫入執行緒進行處理，整個訊息傳遞伺服器的運作便是由這三個執行緒完成。

將處理訊息的過程切割成四個狀態且由數個執行緒來處理的用意在於，當整個過程交由同一個執行緒處理時，勢必會在速度最慢的硬碟寫入部分耗費大量時間，導致伺服器的整體處理效率下降。因此依照功能不同將整體過程切割交由不同的執行緒平行處理即可避免因硬碟寫入所造成的效能瓶頸。

因此上述各執行緒的關係並非依序執行，而是類似平行處理的概念。例如，訊息讀取執行緒處理完一筆訊息後將資料分別放至網路傳輸佇列與硬碟寫入佇列，然後繼續處理下一筆訊息。而網路傳輸執行緒與硬碟寫入執行緒則是不斷的檢查各自的佇列，一有訊息就開始進行處理，完畢後繼續等待下一筆訊息進入佇列。

圖3-4為各執行緒之狀態轉換示意圖，首先為訊息讀取執行緒，該執行緒一開始會處於接受連線狀態，若有新的客戶端連接上訊息傳遞伺服器，便會由接受連線狀態轉換至訊息讀取狀態，代表伺服器已經準備好接受來自此客戶端之訊息。而在訊息讀取狀態時若接收到來自使用者之訊息，訊息讀取執行緒在處理完相關動作(細節在後面介紹)後，將該訊息推送至網路傳輸佇列與硬碟儲存佇列，至此完成一次循環，而後再度回到訊息讀取狀態等待下一筆訊息。而由訊息讀取狀態回到接受連線狀態的情形會發生在此使用者離開影音連線活動或是關閉客戶端程式時。

網路傳輸執行緒則是處與網路傳輸狀態，不斷檢查網路傳輸佇列中是否有訊息，若有則將訊息依照其通道號碼傳送給對應的使用者，至此完成一次循環。而後繼續回到網路傳輸狀態繼續檢查網路傳輸佇列。

硬碟寫入執行緒與網路傳輸執行緒的狀態轉換相似，在硬碟寫入狀態中不斷檢查硬碟寫入佇列群中的訊息是否滿足寫入策略(將在章節3.3.1中做詳細說明。)所設定之條件，若滿足則將其寫入至硬碟中，完成一次循環。回到硬碟寫入狀態繼續檢查硬碟寫入佇列群。

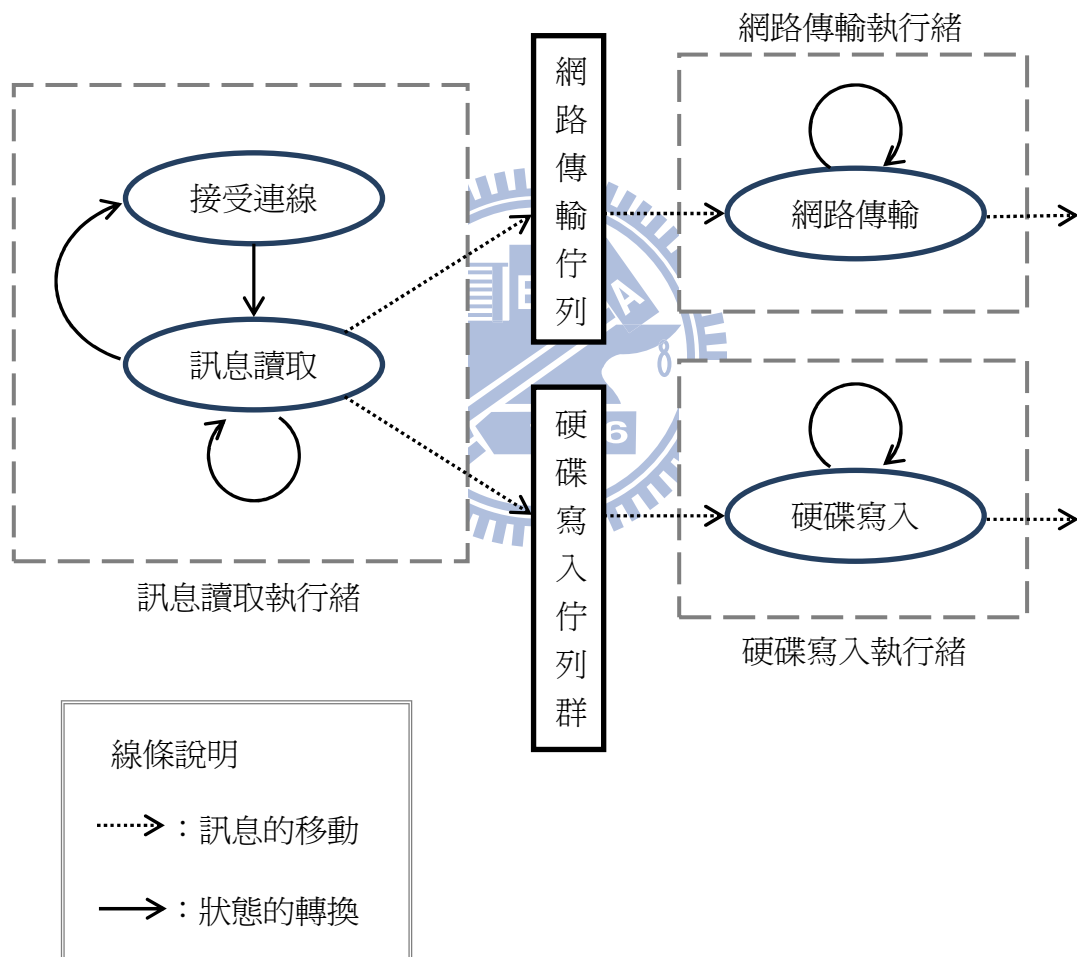


圖 3-4: 各執行緒之狀態轉換示意圖

以下為各狀態的詳細介紹，首先是接受連線，如圖3-5所示。訊息讀取執行緒監視著連線傳輸端口，一旦有客戶端初次連接至訊息傳遞伺服器時便會開始執行。連接傳輸端口在客戶端連接上時會產生一與此客戶端對應的訊息傳輸用通道，訊息傳輸用通道為程序與實體(連線傳輸端口)的連接，所有由該客戶端傳送而來或是欲傳送至該客戶端的訊息皆是由此通道傳輸。

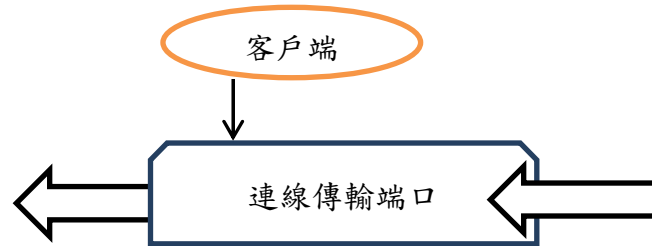


圖 3-5: 接受連線

客戶端在接受連線完成後進入訊息讀取，如圖3-6所示。訊息讀取執行緒發現連接傳輸端口收到訊息，便透過訊息傳輸用通道讀入訊息，進行訊息分析程序將資訊取出，執行該訊息指定的動作並將必要資訊寫進客戶端資訊查詢表，而後訊息分派程序透過查詢表找出對應客戶端之通道號碼，與訊息配對填入至網路傳輸佇列。同時依據訊息的目的討論室號碼，填入對應的硬碟儲存佇列群。

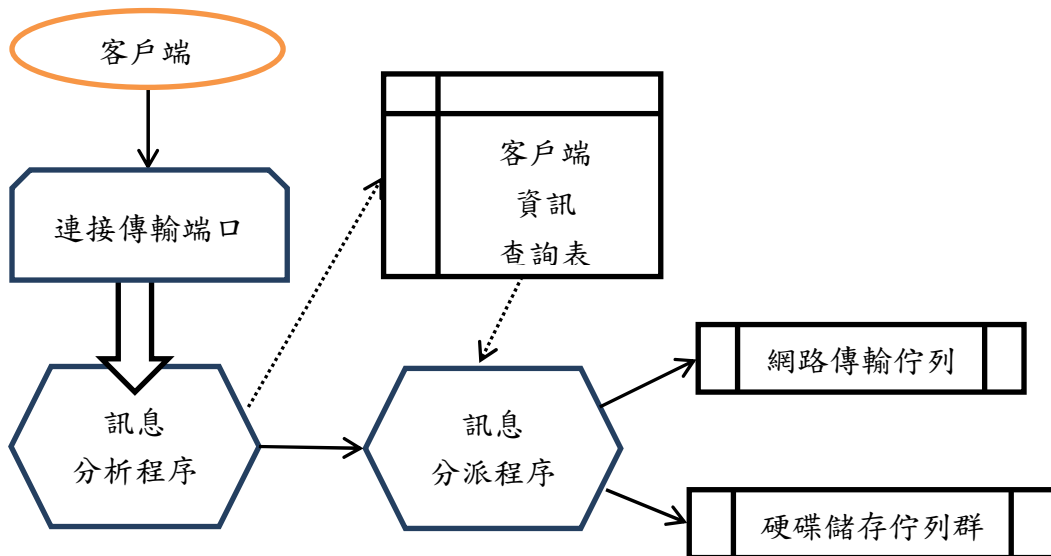


圖 3-6: 訊息讀取

如圖 3-7，網路傳輸執行緒監視著網路傳輸佇列，佇列內有訊息時即開始網路傳輸程序，取出訊息並依據其配對的訊息傳輸通道號碼，透過該通道交由連接傳輸端口準備傳送給對應的客戶端。

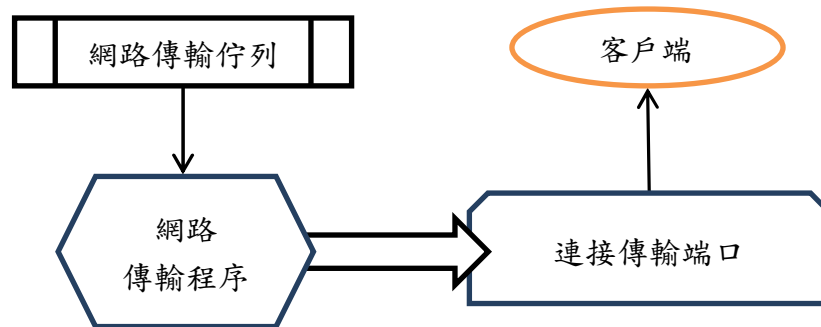


圖 3-7: 網路傳輸

如圖 3-8，硬碟寫入部分與網路傳輸類似，硬碟寫入執行緒監視著硬碟寫入佇列群，差異在於硬碟寫入佇列群要達到一定條件時才會開始硬碟寫入程序，這部分稱為寫入策略，將在章節 3.3.1 中做詳細說明。

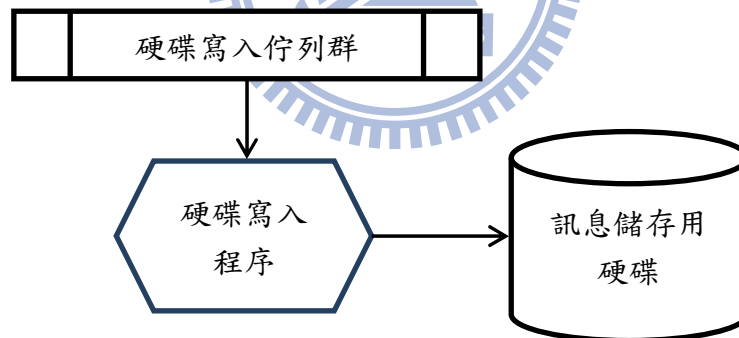


圖 3-8: 硬碟寫入

3.3 訊息傳遞伺服器處理機制介紹

此章節介紹訊息傳遞伺服器的功能與對各種狀況之處理機制，在先前章節 3.1.2 中提到了訊息傳遞伺服器可以想像為大量討論室的集合體，訊息的傳遞以討論室為單位，而伺服器之處理機制亦是環繞討論室來設計。

3.3.1 寫入策略

在章節 3.2.1.2 中硬碟寫入執行緒的部分提到了將寫入訊息的動作與網路傳輸訊息分開，藉此雖可不需等待上筆資料寫入硬碟即可處理下筆資料，但對於整體伺服器的吞吐量並沒有太大幫助，因為伺服器的吞吐量受制於硬碟寫入的速率，而無論是否將寫入訊息的動作與網路傳輸訊息分開，每次對硬碟寫入的資料量皆為單筆訊息，由於每筆訊息寫入的目標文件不同，將導致同時有多份文件檔案交錯的被寫入，硬碟的磁頭將被迫頻繁移動於多份檔案間，使得硬碟寫入的效率低下。此即為寫入策略欲改善之問題。

寫入策略的作法為，每個討論室在創立時均會附帶一個專用的訊息暫存佇列，該討論室中欲儲存的訊息皆暫存於此。這些暫存佇列的集合稱為硬碟寫入佇列群，配置了專屬之記憶體空間供其使用。而佇列內的訊息要滿足所設定之條件後才寫入硬碟，藉由將訊息集合後再進行寫入，盡可能地降低硬碟磁頭移動的次數，使硬碟儘可能的工作於其所擅長的循序存取狀況下，藉此提升硬碟寫入效率並提高伺服器的吞吐量。表 3-3 為何種條件下，將會開始執行硬碟寫入相關動作之對應表。

表 3-3: 何種條件下，將會開始執行硬碟寫入相關動作之對應表

條件	執行動作
該佇列內訊息數達到設定個數	將佇列內的所有訊息寫入硬碟。
該討論室所屬之影音活動完成	將佇列內的所有訊息寫入硬碟，完成後將此佇列刪除。

3.3.2 配置策略與超載機制

訊息傳遞伺服器會依照雲端視訊服務互助平台之要求來開啟討論室並讓使用者進入使用。但若開啟討論室的數量、以及每個討論室內之使用者數量可以無上限增長的話，勢必會造成訊息傳遞伺服器負載過重，更甚者可能導致伺服器故障。配置策略與超載機制即是為了解決上述情形。



3.3.2.1 配置策略

配置策略會在下列兩種狀況下開始作用：

一、平台向訊息傳遞伺服器要求開啟新的討論室。

訊息傳遞伺服器會依據其本身硬體負載狀況是否達到所設定之開啟討論室上限值來決定答應/拒絕開啟要求。而開啟討論室上限值以 CPU 使用率為基準，當訊息傳遞伺服器之 CPU 使用率達到 70%時即為達到開啟討論室上限值，無法在伺服器上開啟新的討論室。

而訊息傳遞伺服器在開啟一個新的討論室後，會將自身所測得之 CPU 使用率暫時增加一固定值(目前定為 2%)，隔一段時間後(目前定為 10 分鐘)再扣除。這麼做的原因為在討論室剛開啟時，由於尚未有其他使用者或是只有極少數使用者加入此討論室進行活動，因此這個新開啟的討論室對於伺服器之 CPU 使用率幾乎沒有提升的效果。若是在短時間內有大量開啟討論室之要求，伺服器極可能會全數接受並開啟大量新的討論室，造成在使用者陸續進入這些新的討論室進行活動後，伺服器之負擔大大提升至無法承受的地步。因此在開啟新的討論室後，透過將伺服器 CPU 使用率暫時增加而後扣除的作法來避免上述情形。

二、使用者要求進入某討論室參與活動。

訊息傳遞伺服器會去查詢開討論室之使用者名單，依照其內之人數是否達到討論室內人數上限來決定答應/拒絕進入請求。討論室內人數上限依照平台之使用型態與需求而定，目前均統一為 100 人。而設置討論室人數上限原因請見章節

4.3.1 訊息傳遞伺服器處理能力測試中之子測試 C 部分。

3.3.2.2 超載機制

當使用者之發送訊息行為踴躍造成訊息傳遞伺服器其 CPU 使用率即將達到極限時，為了保障已存在之討論室傳送訊息品質，必須要有一對應方式來處理此情況，保證正在進行之影音活動能順利進行。此對應方式即為超載機制。

當訊息傳遞伺服器其 CPU 使用率超過 0%時超載機制即會啟動，限制各討論室內旁觀者之發言權。詳細作法為，若是伺服器所收到之訊息為旁觀者所發送，便不將其進行訊息的分析與轉送，藉此讓伺服器之實際處理訊息量數減少、降低伺服器負擔。而各討論室之開創者其訊息傳送則不受影響，保證影音連線活動

能順利完成。

3.4 分散式訊息傳遞伺服器群設計

雲端視訊服務互助平台目前的硬體架構為網頁伺服器、訊息傳遞伺服器、影音伺服器各一台，僅能服務少數的使用者。為了因應未來的平台之使用者成長，將單台的訊息傳遞伺服器擴充為分散式訊息傳遞伺服器，在不影響平台整體架構的前提下進行訊息傳遞伺服器的修改。

3.4.1 分散架構

分散式訊息傳遞伺服器之設計想法為，採用階層式架構設計，由一中央分派伺服器來統整旗下各個訊息傳遞伺服器。在往後章節之圖與其相關解釋時，網頁伺服器將稱為 Web server，訊息傳遞伺服器將稱為 Msg Server，影音伺服器將稱為 Media Server。分散式架構如圖 3-9 所示。

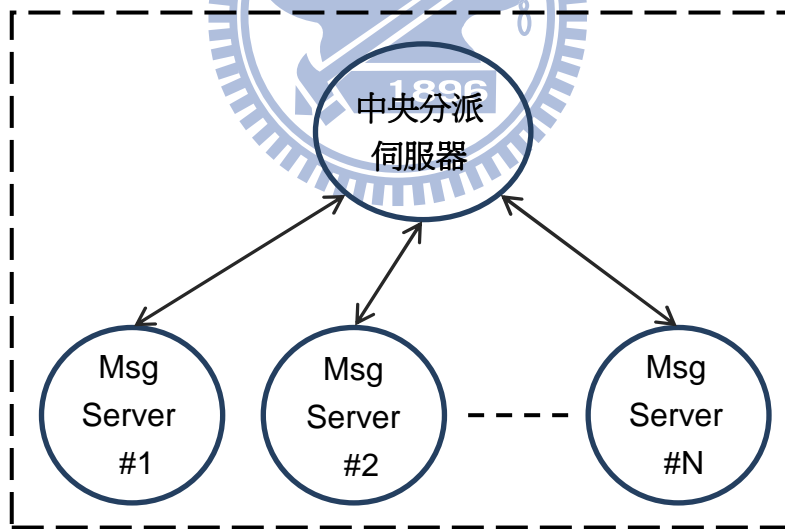


圖 3-9: 分散式 Msg server 架構示意圖

中央分派伺服器為分散前後差異最大之處，負責掌管該分散架構內所有 Msg Server，例如：監控各 Msg Server 負載狀況、決定新的影音連線活動之討論室要配置在哪個 Msg Server 上、以及當某台 Msg Server 故障時該如何處理在其上之討論室與使用者(將在 3.4.2、3.4.3 中說明)。而 Msg Server #1 至 #N 每台伺

服务器的功用相同，擁有 3.1.2.2 所提到之基本功能。

接著介紹若使用分散式訊息傳遞伺服器設計，將對雲端視訊服務互助平台現有之架構造成何影響。圖 3-10 為原先架構下開啟新的視訊連線活動時，與訊息傳遞伺服器相關之部分。

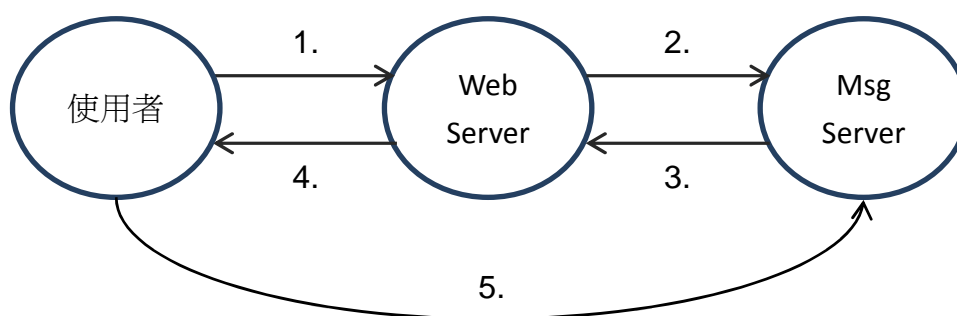


圖 3-10: 開啟新的視訊連線活動，其過程中與訊息傳遞伺服器相關之部分

以下為圖 3-X 之詳細說明：

1. 使用者向 Web Server 要求開啟新的視訊連線活動。
2. Web Server 向 Msg Server 詢問是否允許開啟新的討論室。
3. Msg Server 允許，開啟新的討論室並回覆 IP 與討論室號碼。
4. Web Server 將 Msg Server 之 IP 與討論室號碼給與使用者。
5. 使用者依據 IP 連線至該 Msg Server，進入與討論室號碼相符之討論室，開始進行訊息傳遞。

而修改為分散式訊息傳遞伺服器設計後，圖 3-10 所示流程將轉變為圖 3-11。

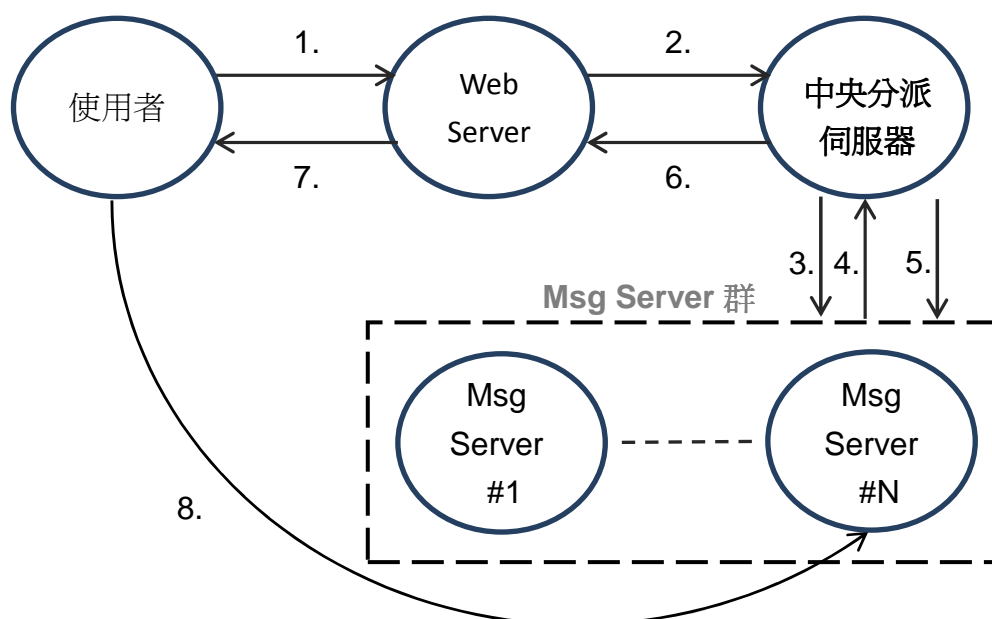


圖 3-11：圖 3-10 流程之分散式版本

以下為圖 3-11 之詳細說明：

1. 使用者向 Web Server 要求開啟新的視訊連線活動。
2. Web Server 向中央分派伺服器詢問是否允許開啟新的討論室。
3. 中央分派伺服器詢問其下各個 Msg Server 負載情況。
4. 各 Msg Server 回報負載情況。
5. 中央分派伺服器依據回報結果決定該討論室要分派給哪個 Msg Server，對該 Msg Server 發送開啟討論室命令，該 Msg Server 開啟討論室。
6. 中央分派伺服器回覆該訊息傳遞伺服器 IP 與討論室號碼給網頁伺服器。
7. Web Server 將 IP 與討論室號碼給與使用者
8. 使用者依據 IP 連線至該 Msg Server，進入與討論室號碼相符之討論室，開始進行訊息傳遞。

3.4.2 分散策略

此處將介紹如何將平台所要求開啟之討論室分散在各個訊息傳遞伺服器上。首先中央分派伺服器依據其下各 Msg Server 回報之負載狀況來決定新的視訊連線活動之討論室要開啟在何處。而此處描述的負載狀況，也就是各伺服器其開啟討論室上限值，其基準與章節 3.3.2.1 配置策略一樣使用 CPU 使用率。

中央分派伺服器會指定 CPU 使用率最低之訊息傳遞伺服器來開啟新的討論室，被指定之訊息傳遞伺服器其開啟新的討論室做法與章節 3.3.2.1 配置策略相同，在開啟討論室後會將自身所測得之 CPU 使用率暫時增加一固定值，隔一段時間後再扣除。避免短時間內開啟的討論室皆集中在同一台訊息傳遞伺服器上。

3.4.3 故障處理

當訊息傳遞伺服器群中，突然有一台故障了，要如何重新配置在故障訊息傳遞伺服器上的討論室與使用者至其他正常運作的訊息傳遞伺服器上便是此章節要處理之議題。

在進行故障處理前，有一項設定必須先告知，視訊連線活動中每個使用者皆有其身分與對應的權限(詳細請看附錄)，當中存在唯一一個最高權限者稱為管理者，這個管理者將成為故障處理的發起者，以下將詳細說明。圖 3-12 為當訊息傳遞伺服器故障時，其故障處理之流程圖



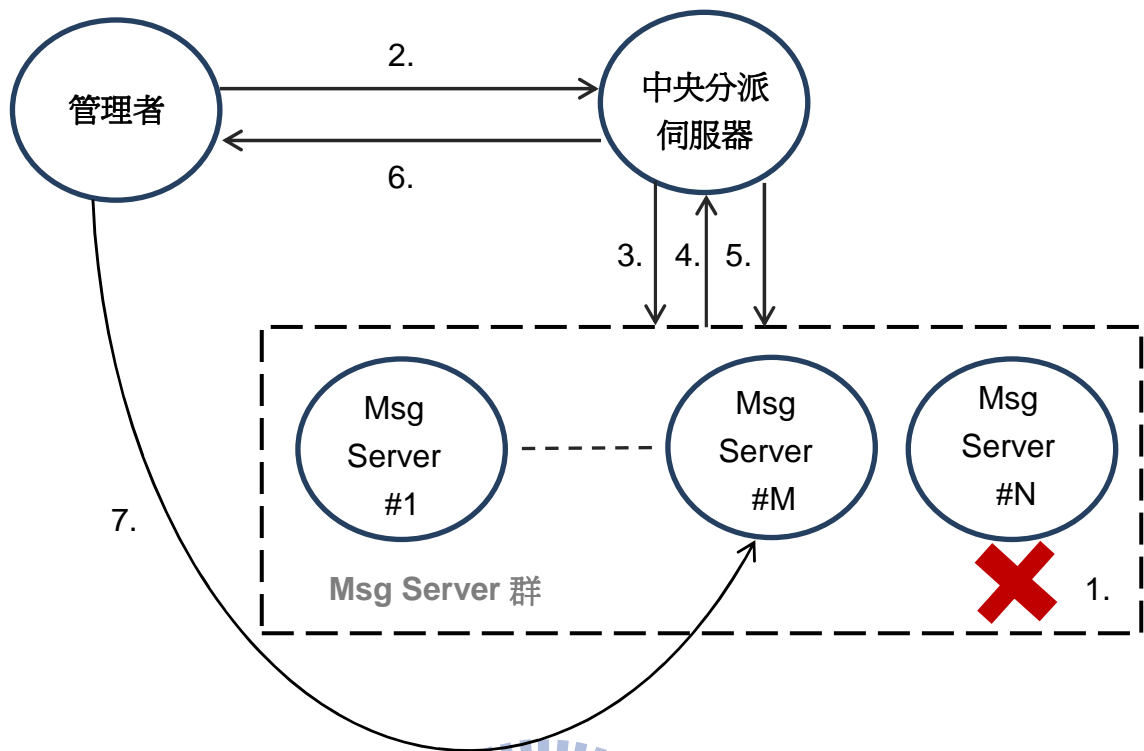


圖 3-12: 故障處理流程圖

圖 3-12 之流程說明：

1. Msg Server #N 發生故障。
2. 管理者的客戶端程式發現故障，自動向中央分配伺服器回報該討論室內存在多少使用者以及要求一個新的 Msg Server 來繼續視訊連線活動。
3. 中央分派伺服器詢問其下各個 Msg Server 負載情況。
4. 各 Msg Server 回報其負載狀況。
5. 中央分派伺服器挑選負載狀況最低之 Msg Server，此處為 Msg Server #M，對其發送開啟討論室命令，Msg Server #M 開啟討論室。
6. 中央分派伺服器回覆 Msg Server #M 的 IP 與討論室號碼給管理者。
7. 管理者連向 Msg Server #M 繼續原本之視訊連線活動。

註：步驟 5 之作法與章節 3.4.2 分散策略做法相同。

僅有管理者的客戶端程式會在故障時自動連線至中央分派伺服器，其他身分之使用者則是在視訊連線活動頁面中彈出一確認重連的提示，使用者須自行手動按下確認後才會連向中央分派伺服器，詢問該連向何處後再轉向新分派的訊息傳遞伺服器。如此設計的理由如下：

1. 確保每個討論室僅有一人發送重啟討論室要求，避免重複開啟討論室。
2. 錯開各使用者連向中央分派伺服器的時間，避免在短時間內大量使用者湧入中央分派伺服器造成伺服器癱瘓。



第四章 系統測試

4.1 雲端服務視訊互助平台整體測試

雲端服務視訊互助平台整體測試欲得知在視訊連線過程中的使用者人數，對訊息傳遞伺服器與其他伺服器之間的訊息傳遞與處理工作所需時間的影響。分別測試視訊連線過程中兩種與訊息傳遞伺服器相關的訊息傳遞路徑，記錄在不同的使用者人數下，傳遞訊息所需的時間。

兩種主要的訊息傳遞路徑介紹如下：

一、網頁伺服器傳送訊息至客戶端 Flash

訊息傳遞路徑如圖 4-1。Web server 透過 Msg Server 將訊息轉發至客戶端的 Flash 程式，此訊息傳遞路徑主要用於當使用者拒絕或允許連線請求時，通知已經進入連線活動的其他使用者，

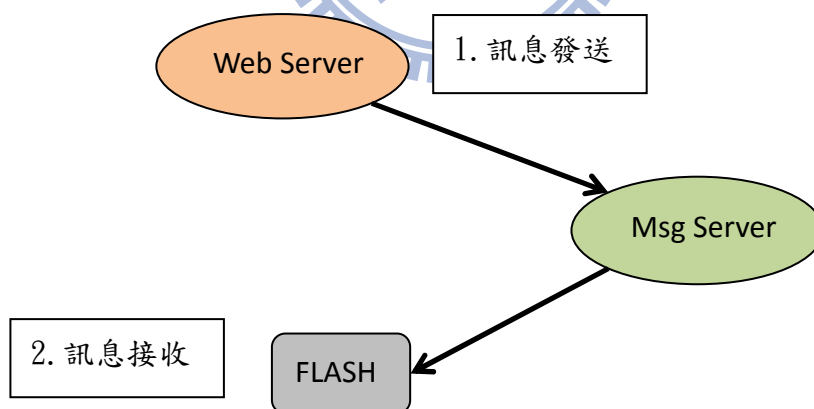


圖 4-1: 網頁伺服器傳送訊息至客戶端 Flash 流程圖

二、客戶端發送訊息要求其他客戶端接收影音串流

訊息傳遞路徑如圖 4-2，發佈端(Flash A)向 Msg Server 發送訊息，訊息內容為"要求客戶端向 Media Server 接收串流"，Msg Server 轉送訊息給接收端(Flash B)，接收端收到訊息後向 Media server 送出接收串流的要求，取得影音串流。

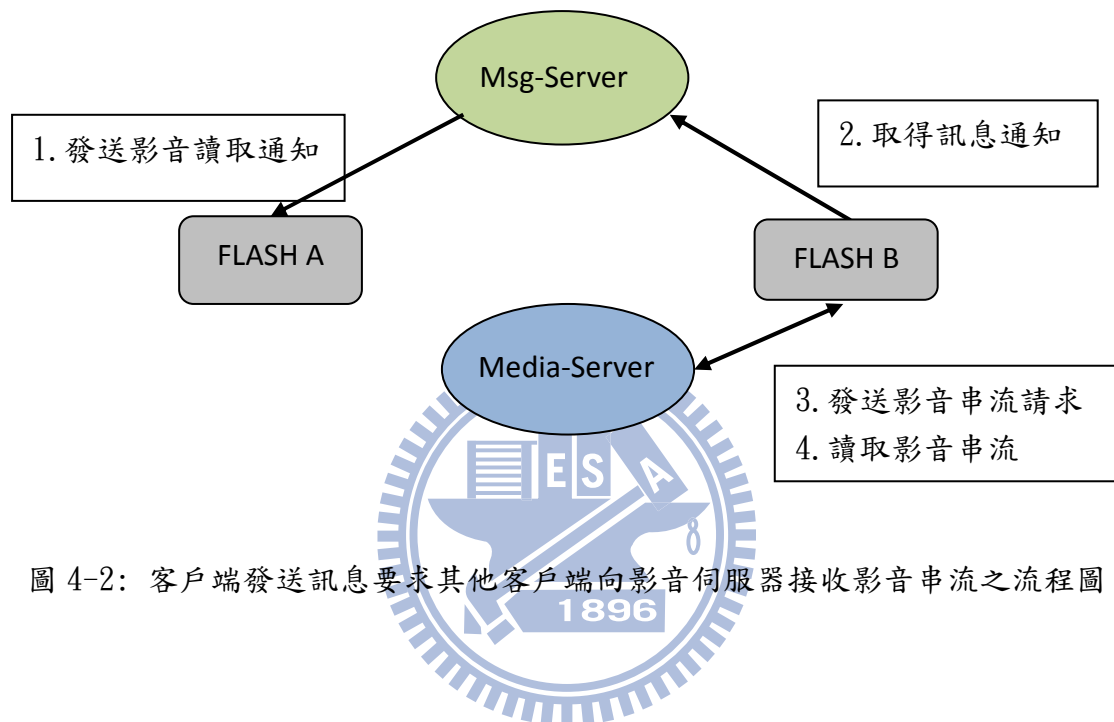


圖 4-2: 客戶端發送訊息要求其他客戶端向影音伺服器接收影音串流之流程圖

4.1.1 測試方法

在上述兩種訊息傳遞路徑之測試中，每次測試會逐次增加測試人數。由測試人數 1 人開始，做數次測試取得其平均完成時間，完成一次測試；接著增加至 10 人做第二次測試，而後每次增加 10 人直至做到 90 人為止。

超過 90 人之後，因測試人員及設備有限，所以利用測試軟體模擬多於 90 位使用者加入視訊連線活動的情形，逐次增加 100 人直至三種伺服器其中一者達到瓶頸。

以下介紹各伺服器之測試軟體如何模擬多於 90 位使用者之情形：

一、Web Server：

當使用者登入本網站平台後，即使不做任何之操作，網頁程式仍會定期向網頁伺服器送出 request，以更新客戶端資料。因此假設每位使用者登入本網站平台後，固定停留在首頁並且開啟一彈跳視窗旁觀視訊連線活動，之後便不做其他操作行為，則從網頁伺服器的 log 檔分析得出，在此使用者行為定義下，每位使用者平均每秒會對網頁伺服器送出 0.412 則 request 以更新資料。

依據此對應比率，使用 http_load 此測試軟體，以固定頻率對網頁伺服器送出 request，即可模擬多位使用者進入網站平台後對網頁伺服器之影響，例如：每秒對網頁伺服器送出 41 則 request，則可模擬 100 位使用者之負載。

二、Msg Server：

視訊連線活動頁面中的影音畫框與聊天室程式會依使用者之操作對 msg server 分別發送標記、對話文字及用戶端程式控制指令訊息，發送的頻率與訊息長度會隨使用者行為而改變。因此當使用測試軟體模擬時，此測試軟體設計需要盡量貼近上述之使用者行為。在此依據平台現有之視訊連線紀錄所統計出來之使用者發送訊息行為模型，設定每個使用者平均每秒對 Msg Server 發送 0.048 則訊息，訊息長度為 52.7 個字元。因此，藉由每秒發送模擬人數乘上 0.048 則長度為 53 個字元之訊息，可以模擬 Msg Server 處理 100 人以上之大量使用者操作時的情況。（註：Msg Server 詳細的測試程式設計將於章節 4.2 中說明。）

三、Media Server

採用 Smaxe 公司的 JUV RTMP LoadTester (Lite) 0.9 作為模擬大量使用者的測試軟體，先對 Media Server 進行預先設定數量的 RTMP 連線，可用來模擬相同數量的使用者對 Media Server 造成的 CPU、RAM、硬碟效能與網路等之負載情形。

4.1.2 測試環境

表 4-1 為各 Server 之硬體設備，各 Server 間以區域網路互相串連。在各項測試中，各 Server 與客戶端彼此皆以區域網路互相連線，使用 ping 測量後，任二者彼此訊息交換所需傳送時間皆在 0.07ms 以內，對於測試數據的影響 0.001% 以下，因此忽略由網路造成的延遲。

表 4-1: Web、Media、Msg Server 之硬體配備列表

Hardware Configuration	
Processors	Intel® Pentium® Processor E5300 @ 2.60 GHz (LGA 775)
Motherboards	ASUS P5KPL-CM (G31/ICH7) LGA 775
Memory	4 GB (2GBx2) Kingston KVR800D2N6/2G DDR2-800 6-6-6-18 @ 1.8 V
Graphics	Integrated Intel Graphics Media Accelerator (Intel® GMA 3100)
Network	Atheros AR8121/AR8113/AR8114 PCI-E Ethernet Controller (L1e)
Storage	Seagate Barracuda 7200.10 160 GB

表 4-2 為 Web Server 與 Media Server 所使用的軟體版本。Msg Server 是自行以 Java 程式開發，因此僅列出使用的 Java Virtual Machine(JVM)版本。

表 4-2: Web、Media、Msg Server 的軟體版本

Server	軟體版本	作業系統
Web Server	PHP Version: 5.2.17 nginx version: nginx/1.0.0	CentOS 5.5
Media Server	Red5 1.0 RC1	Scientific Linux 6.0

Msg Server	JRE 6.0	Scientific Linux 6.0
------------	---------	----------------------

而測試時客戶端使用的 Flash player 版本為 Flash player 10.3。

4.1.3 網頁伺服器傳送訊息至客戶端 Flash 測試

測試流程：

使用一 shell script 程式對 Web Server 中之 PHP 程式發出訊息傳遞要求，模擬使用者在網頁上之要求訊息傳遞之行為。

當 shell script 程式對 PHP 程式送出訊息傳遞要求時記錄下時間 T11。此 PHP 程式被觸發後會發送訊息至 Msg Server，發送的同時記下時間 T12。最後當客戶端的 Flash 程式從 Msg Server 收到此訊息後，由 flash 程式記下時間 T13。

紀錄上述時間的目的如下：

T12-T11 之值為 Web Server 被觸發傳遞訊息到訊息發送完畢所耗費的時間。此值僅與 Web Server 相關。

T13-T12 之值為 Web Server 實際送出訊息至 Msg Server，客戶端 Flash 程式在由 Msg Server 處收到訊息所耗費的時間。此值僅與 Msg Server 相關。

T13-T11 之值為由觸發 Web Server 傳遞訊息至客戶端所耗費的總時間。

量測數據：

圖 4-3 為測試人員實際進入平台進行測試所得數據之圖。圖 4-4 為使用模擬軟體進行測試所得數據之圖。而使用模擬軟體進行之測試至 1200 人即停止的原因為，此時 Web Server 之訊息傳遞或處理工作之錯誤率已經超過 70%，錯誤率之定義為使用者欲透過網頁伺服器傳遞 10 則訊息至客戶端 Flash，共對網頁伺服器提出 20 次的傳遞要求，若客戶端 Flash 程式只接到 10 則訊息，則錯誤率即為 50%。

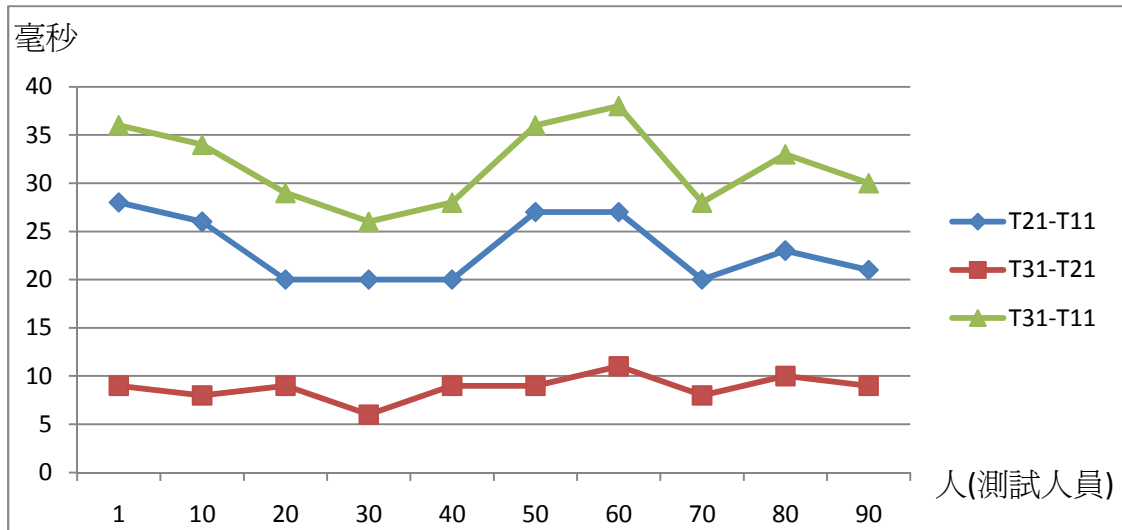


圖 4-3: 使用者(測試人員)人數與網頁伺服器傳遞訊息至客戶端 Flash 耗費時間之關係圖

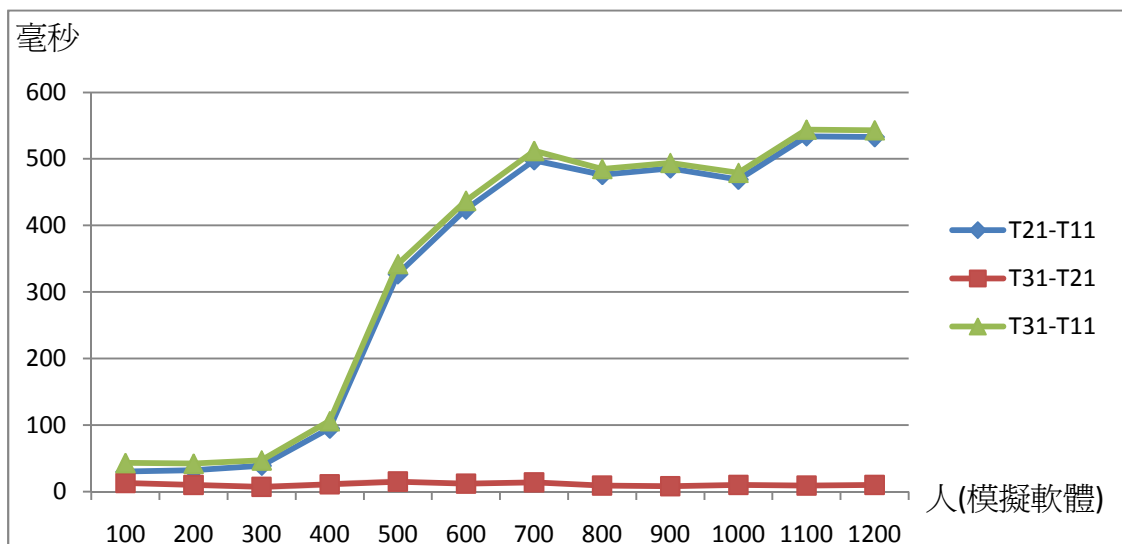


圖 4-4: 使用者(模擬軟體)人數與網頁伺服器傳遞訊息至客戶端 Flash 耗費時間之關係圖

數據分析：

此處僅分析與 Msg Server 相關的 T13-T12 之值。可以發現 T13-T12 之值不論是使用測試人員或是模擬軟體測試其值在人數提高時皆穩定在 10ms 左右，若是想要服務更多的使用者，則需從 Web Server 處進行改善。

4.1.4 客戶端發送訊息要求其他客戶端接收影音串流之測試

測試流程：

使用一客戶端程式(Flash A)向另一客戶端程式(Flash B)發送接受串流訊息，模擬真實情形下如切換攝影機等動作。

當 Flash A 發送測試訊息給 Msg Server 時由 Flash A 記錄下時間 T21。Flash B 在收到測試訊息後由 Flash B 記錄下時間 T22。Flash B 接著向 Media Server 要求串流時由 Flash B 記錄下時間 T23。最後 Flash B 由 Media Server 收到串流(串流流量為皆 400kbps)時由 Flash B 記錄下時間 T24。

紀錄上述時間的目的如下：

T22-T21 之值為由 Flash A 發出接受串流訊息，到 Flash B 收到此訊息所耗費的時間。此值僅與 Msg Server 的處理時間相關

T23-T22 之值為由 Flash B 收到接受串流訊息後，經過處理後再向 Media Server 要求串流所耗費的時間。此值僅與客戶端之 Flash 處理時間相關。

T24-T23 之值為 Flash B 向 Media Server 要求串流，到 Flash B 接收到串流所耗費的時間。

T24-T21 之值為由 Flash A 發送接受串流訊息，到 Flash B 真正接受到該串流所耗費的總時間

量測數據：

圖 4-5 為測試人員實際進入平台進行測試所得數據之圖。圖 4-6 為使用模擬軟體進行測試所得數據之圖。而使用模擬軟體進行之測試至 500 人即停止的原因為，此時 Media Server 已經無法傳送更多串流。

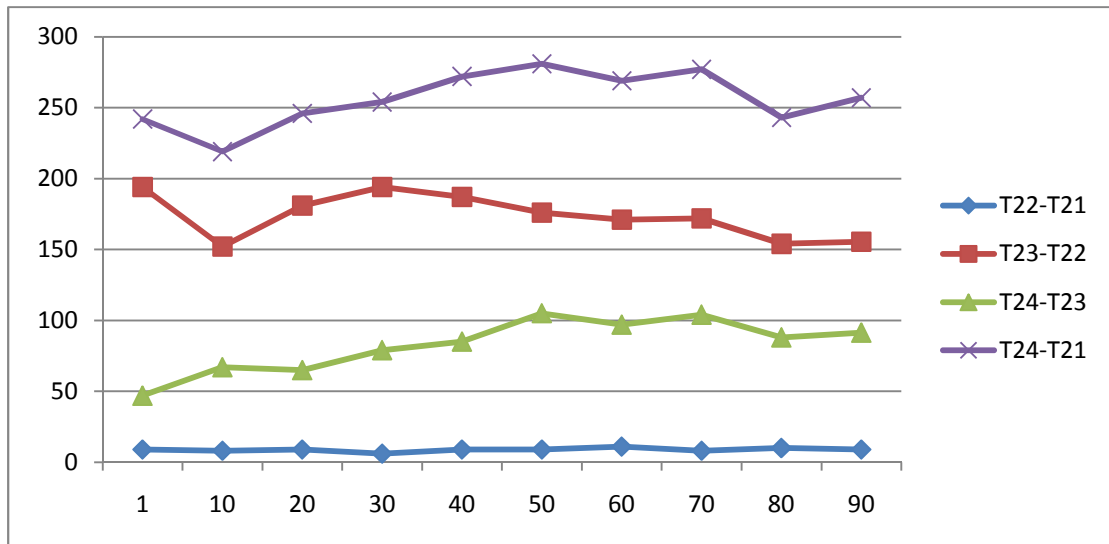


圖 4-5: 使用者(測試人員)人數與客戶端發送訊息要求其他客戶端向影音伺服器接收影音串流耗費時間之關係圖

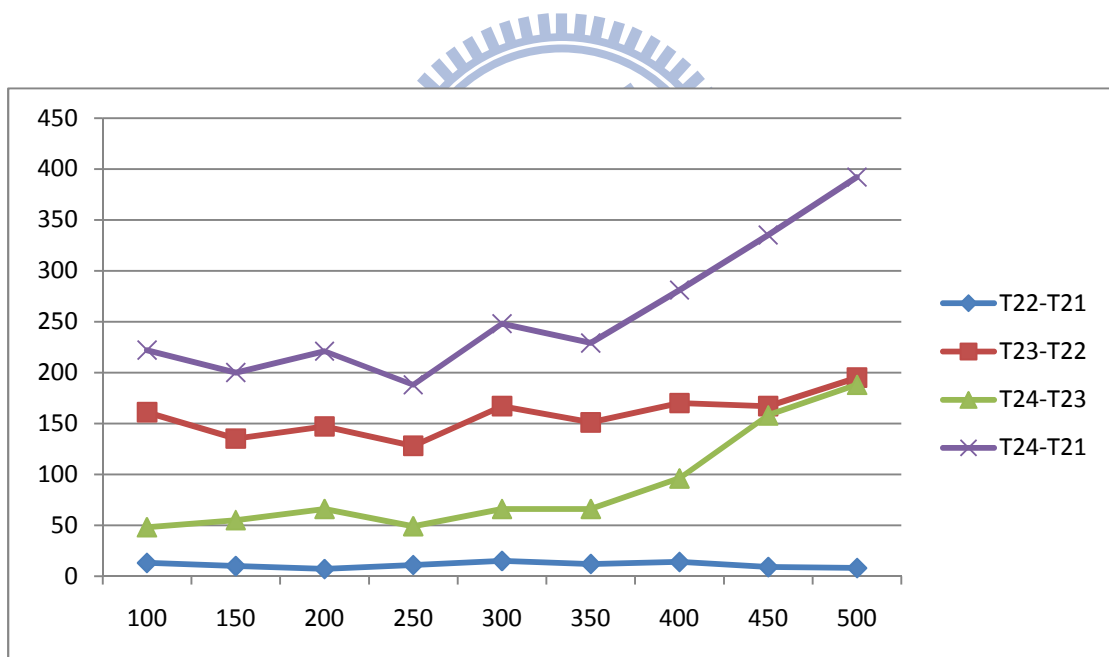


圖 4-6: 使用者(模擬軟體)人數與客戶端發送訊息要求其他客戶端向影音伺服器接收影音串流耗費時間之關係圖

數據分析：

此處僅分析與 Msg Server 相關的 T22-T21 之值。可以發現 T22-T21 之值不論是使用測試人員或是模擬軟體測試其值在人數提高時皆穩定在 10ms 左右，若是想要服務更多的使用者，則需從 Media Server 處進行改善。

4.2 訊息傳遞伺服器測試軟體設計

4.2.1 雲端服務視訊互助平台中的使用者發送訊息行為統計

在設計訊息傳遞伺服器測試軟體之前，必須要先了解雲端服務視訊互助平台的使用者在影音連線活動進行時其訊息發送行為，也就是訊息平均發送頻率與訊息平均長度。藉此作為設計訊息傳遞伺服器測試軟體的依據。

觀察平台中在正常使用下之視訊連線活動，分別統計每個使用者會對訊息傳遞伺服器送出多少則訊息以及其訊息長度總和，而後除以視訊連線活動時間，即可得到訊息平均發送頻率以及訊息平均長度。

而依照使用者身分的不同(請參照 4.4 附註:使用者在視訊連線活動中之身分說明)，其所發送的訊息量與長度也會有所差異。例如視訊連線活動的開創者因為可以使用影音標註及攝影機切換功能，所以其訊息平均發送頻率會高於其他無法使用此兩項功能之旁觀者。表 4-3 為伺服器上所儲存 7 月 1 號至 7 月 26 號之 41 個視訊連線活動其紀錄及統計結果。

表 4-3: 使用者發送訊息行為之記錄及統計結果

身分	樣本數	所佔總人數之比例	發送訊息次數總和	發送長度總和	訊息平均發送頻率	訊息平均長度
開創者	64	15%	10249	876319	0.126	85.5
人			訊息	字元	則/秒	字元
旁觀者	371	85%	16032	509830	0.034	31.8
人			訊息	字元	則/秒	字元

由表 4-3 得知，由統計結果可見開創者與旁觀者的比例約為 15:85。每個使用者當其身分為開創者時，訊息平均發送頻率約為 0.126 則/秒，訊息平均長度約為 85.5 個字元。而使用者身分為旁觀者時，訊息平均發送頻率約為 0.034 則/

秒，訊息平均長度約為 31.8 個字元。此數據視為訊息傳遞伺服器之使用者行為模型，用以將伺服器每秒所能處理之訊息量換算為伺服器能服務多少使用者。

例如使用此使用者模型模擬 100 位使用者時，表示其中約有 15 位開創者與 85 位旁觀者，等同於訊息傳遞伺服器每秒需處理開創者部分的(15*0.126)則長度為 86 字元之訊息，再加上旁觀者部分的(85*0.034)則長度為 32 字元之訊息。平均計算下來伺服器在 100 位使用者時，平均每秒約處理 4.8 則訊息，平均訊息長度為 52.7 個字元。

4.2.1 雲端服務視訊互助平台中的使用者發送訊息行為統計

測試軟體會對伺服器發送固定數量的訊息，紀錄下開始發送的時間與全部訊息發送完畢的時間(時間以毫秒為單位)，將後者減去前者即為伺服器處理該數量訊息的耗費時間。耗費時間除以所發送的訊息總量即可得伺服器平均處理一個訊息要花費多少時間(毫秒)。最後將 1000 除以伺服器平均處理一個訊息要花費多少時間即為伺服器一秒鐘能夠處理多少筆訊息。

$$\text{耗費時間} = \text{開始發送時間} - \text{全部訊息發送完成時間 (ms)}$$

$$\text{伺服器平均處理一個訊息所花費時間} = \frac{\text{耗費時間}}{\text{伺服器訊息發送總量}} \quad (\text{ms})$$

$$\text{伺服器一秒鐘能夠處理多少筆訊息} = \frac{1000}{\text{伺服器平均處理一個訊息所花費時間}}$$

而每個討論室所分派到的訊息數是依照每個討論室中人數對總人數的比例來分配。例如伺服器內有 1000 人，其中有 200 人在 A 討論室內，則測試軟體發送至 A 討論室的訊息量即為總發送訊息量的 20%。

4.3 訊息傳遞伺服器測試

本節測試欲得知在各種狀況下的訊息傳遞伺服器處理能力，也就是訊息傳遞伺服器每秒最多能處理多少訊息數。接著測試訊息傳遞伺服器上之寫入策略對於處理能力之影響，以及訊息傳遞伺服器之分散設計對於提升服務人數的效果。

測試環境與 4.1.2 節同，測試訊息長度皆設為 50 個字元。寫入策略之佇列大小皆為 1000 個訊息。

註：此章節中所提之伺服器皆為訊息傳遞伺服器。

4.3.1 訊息傳遞伺服器處理能力測試

使用自行撰寫之測試軟體模擬大量使用者之行為，觀察在各項伺服器參數（使用者人數、討論室數量、是否均勻分布）設定下，觀察伺服器每秒最多能處理多少訊息數。訊息傳遞伺服器處理能力測試又分為下列各子測試。

子測試 A.

討論室內使用者人數對於訊息傳遞伺服器處理能力的影響：
不同的伺服器承載人數均勻分布在固定數量的討論室中時，觀察訊息傳遞伺服器每秒最多能處理多少訊息數。

子測試 B.

討論室數量對於訊息傳遞伺服器處理能力的影響：
固定討論室中的使用者人數，在討論室數量增加時，觀察訊息傳遞伺服器每秒最多能處理多少訊息數。

子測試 C.

伺服器承載人數非均勻分布在討論室中對於訊息傳遞伺服器處理能力的影響：
固定的使用者人數非均勻地分布在固定數量的討論室中，增加某一討論室（往後此討論室皆稱為**加權討論室**）之使用者人數比例，觀察訊息傳遞伺服器每秒最多能處理多少訊息數。

子測試 D.

使用者所送訊息長度對於訊息傳遞伺服器處理能力的影響：

固定的使用者人數均勻分布在固定數量的討論室中時，改變所發送之測試訊息長度，觀察訊息傳遞伺服器每秒最多能處理多少訊息數。

測試方法：

上述四項測試中伺服器的參數設定如下，表 4-4 為子測試 A 之參數設定。表 4-5 為子測試 B 之參數設定。子測試 C 之參數設定將使用文字說明。表 4-6 為子測試 B 之參數設定。

表 4-4: 討論室內人數對於訊息傳遞伺服器處理能力影響之參數設定表

討論室內人數	討論室數量	使用者人數在討論室中之分布狀況
10	100	均勻分布
20	100	均勻分布
.	.	.
.	.	.
.	.	.
100	100	均勻分布

表 4-5: 討論室數量對於訊息傳遞伺服器處理能力影響之參數設定表

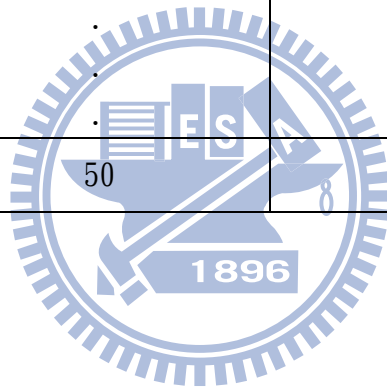
討論室數量	伺服器承載人數	使用者人數在討論室中之分布狀況
10	1000	均勻分布
20	2000	均勻分布
.	.	.
.	.	.
.	.	.
100	10000	均勻分布

子測試 C 之參數設定說明：

將伺服器的總使用者人數訂為 1000 人，討論室數量訂為 10 間。使用者分布情形為加權討論室佔有總使用者人數乘上一係數 X 之人數， X 稱為加權討論室之佔有率，由 0.1 開始，每次遞增 0.1 直至 X 等於 0.9)，剩餘的使用者人數則是均勻散布在其餘 9 間討論室內。

表 4-6：使用者所送訊息長度對於訊息傳遞伺服器處理能力影響之參數設定表

測試訊息長度	討論室數量	討論室內使用者人數
20	50	100
30	50	100
.	.	.
.	.	.
.	.	.
100	50	100



測試數據與分析討論：

子測試 A

量測數據：

此測試在各種人數條件下，當伺服器每秒處理的訊息數量達到其最多能處理訊息數量時，伺服器之 CPU 使用率均達到 100%，記憶體與網路流量皆尚未到達極限，因此在此測試時 CPU 為伺服器之瓶頸所在。

圖 4-7 為討論室內人數與訊息傳遞伺服器每秒所能處理訊息數量之關係曲線。橫軸代表有多少使用者平均分散在伺服器內的 100 個討論室中。縱軸為伺服器每秒最多能處理的訊息數。

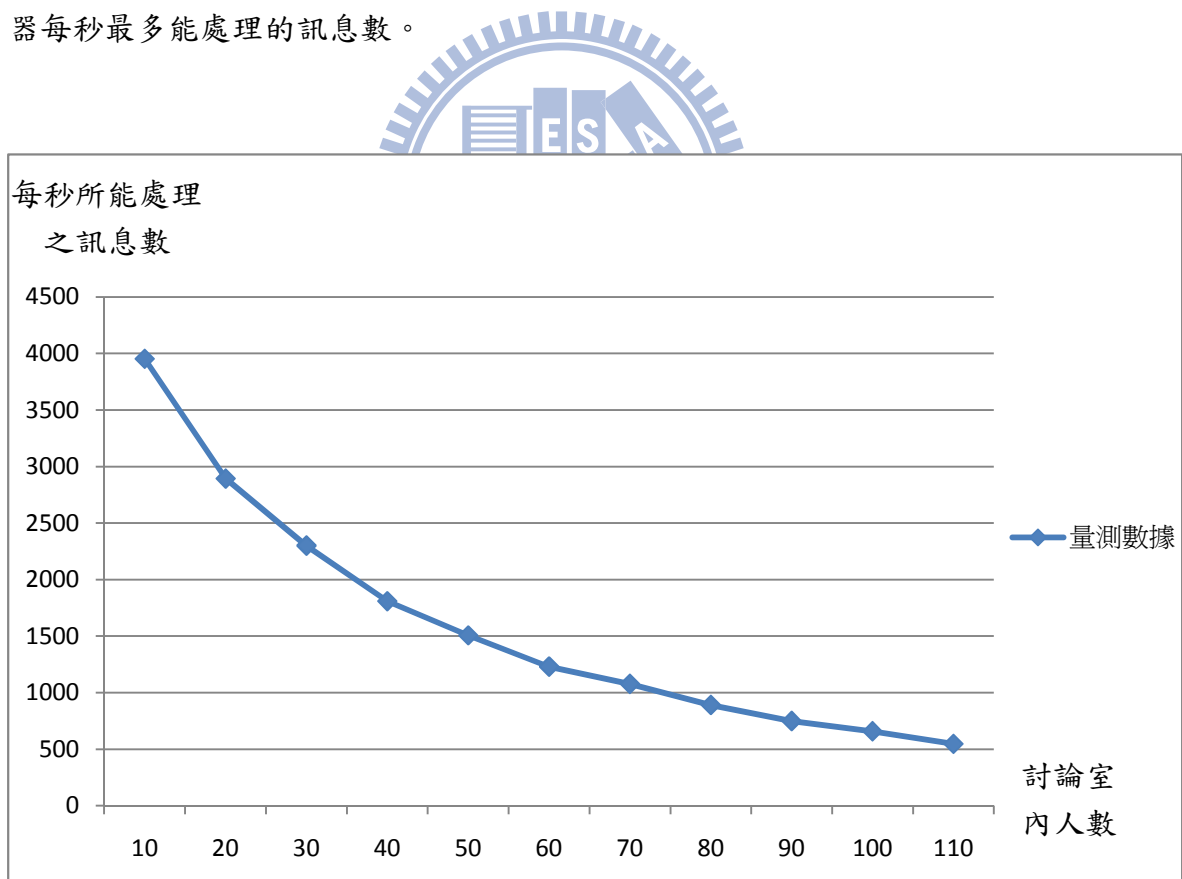


圖 4-7: 在開啟 100 個討論室下，討論室內使用者人數對伺服器處理能力影響之關係曲線圖。

數據分析：

由圖 4-7 可以發現，討論室內使用者人數與訊息傳遞伺服器每秒所能處理訊息數量之關係曲線是呈現一個負的下降趨勢。

以下探討為何此曲線呈現負的下降趨勢，由於討論室之廣播機制會將收到的訊息傳送給在此討論室內的所有使用者，伺服器程式會不斷的去查詢討論室的使用者名單並傳送訊息。在討論室內的人數越來越多，處理每筆訊息時的查詢名單成本與傳送訊息成本便會越來越大。若討論室人數為 n 的話，伺服器處理每筆訊息的複雜度即為 $O(n)$ 。而伺服器每秒最多處理的訊息數與處理每筆訊息的成本為反比關係。因此隨著 n 值的增加，伺服器每秒最多處理的訊息數將會呈現 $1/n$ 之走向。所量測之數據亦符合以上推論。

此外在量測數據中，討論室內人數達到 110 人，也就是伺服器總人數為 11000 人時訊息傳遞伺服器每秒最多能處理訊息數量為 546 則。而在章節 4.2.1中所提到之訊息傳遞伺服器之使用者行為模型若以 11000 人代入的話，可得其平均每秒所發送之訊息數約為 528 則，此數值十分接近在 11000 人之量測數據之 546 則。因此若以此訊息傳遞伺服器之使用者行為模型作為換算標準，即可認為訊息傳遞伺服器在 110 間房間內皆有 100 位使用者時，其最大使用者人數為 11000 人。

子測試 B

量測數據：

此測試在各種討論室數量下，當伺服器每秒處理的訊息數量達到其最多能處理訊息數量時，伺服器之 CPU 使用率均達到 100%，記憶體與網路流量皆尚未到達極限，因此在此測試時 CPU 為伺服器之瓶頸所在。

圖 4-8 為討論室數量與訊息傳遞伺服器每秒所能處理訊息數量之關係曲線。橫軸代表有多少個討論室，每個討論室內皆有 100 位使用者。縱軸為伺服器每秒最多能處理的訊息數。

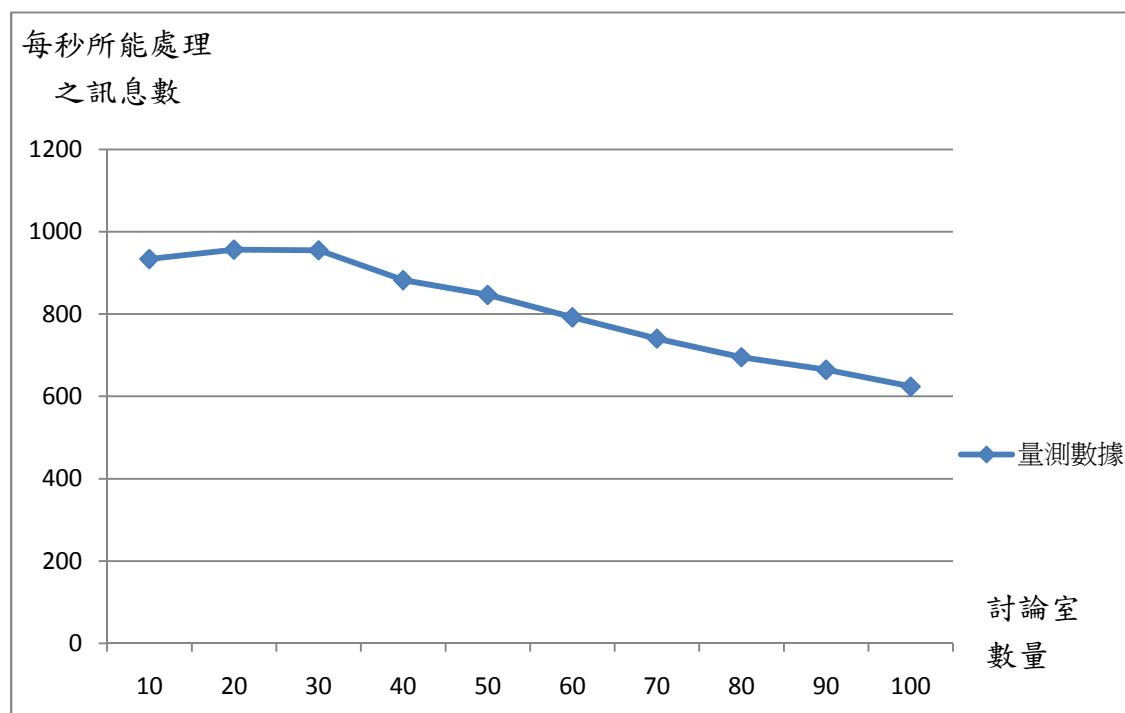


圖 4-8：討論室數量對於訊息傳遞伺服器處理能力影響之關係曲線圖

數據分析：

由圖 4-8 可以發現，討論室數量與訊息傳遞伺服器每秒所能處理訊息數量之關係曲線的趨勢約是呈現一個負的線性下降。

以下探討造成此線性曲線的原因，由於討論室與其內使用者列表的對應由 hash map 完成，每筆訊息在接收後廣播給該討論室之使用者時，查詢動作的複雜度應為 $O(1)$ 而與討論室個數無關，只要每個討論室內使用者數量保持恆定，則總討論室個數的多寡不應影響伺服器每秒所能處理的訊息總數。

但實際情況未必如此簡單，隨著討論室數量增加代表儲存著討論室號碼與其內使用者列表的對應的 hash map 相應的增大。此時資料的存取動作在空間區域性上有所下降，也就意味著快取命中率的下降以及存取時間的上升。同時，討論室變多意味著伺服器上的總使用者個數增加，也代表著系統必須儲存更多的連線狀態，同時網路卡的負擔也會相應的加重，這些應該都是使測試結果不如推論狀況的原因。



子測試 C

量測數據：

此測試在加權討論室的各種佔有率下，當伺服器每秒處理的訊息數量達到其最多能處理訊息數量時，伺服器之 CPU 使用率均達到 100%，記憶體與網路流量皆尚未到達極限，因此在此測試時 CPU 為伺服器之瓶頸所在。

圖 4-9 為伺服器承載人數非均勻分布在討論室對於訊息傳遞伺服器每秒所能處理訊息數量之關係曲線。橫軸代表加權討論室佔有總使用者人數之百分比。縱軸為伺服器每秒最多能處理的訊息數。

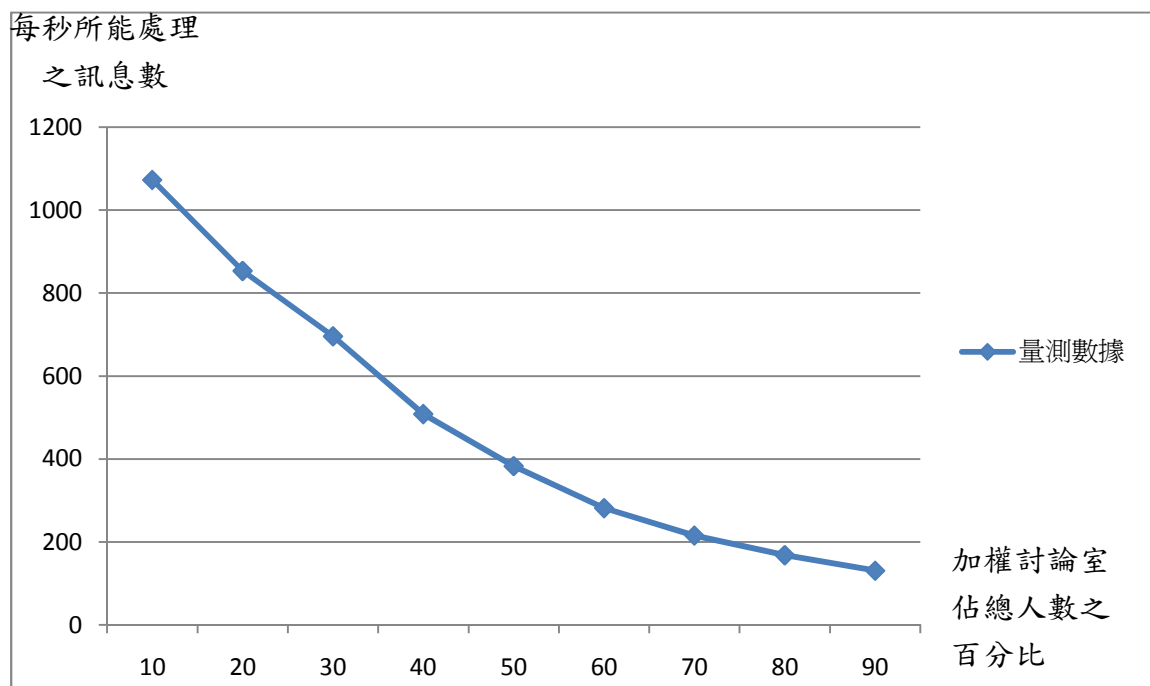


圖 4-9：伺服器承載人數非均勻分布在討論室對於訊息傳遞伺服器處理能力影響之關係曲線圖

數據分析：

首先由伺服器處理訊息的做法方面來考慮其理論上的曲線會是何種類型。推導在各種加權討論室之佔有率下，伺服器處理訊息所需負擔的成本變化。

首先由於測試訊息發送至各討論室的比例是依照各討論室的人數多寡來分配，例如當伺服器內共有 100 位使用者，加權討論室之佔有率為 x ，那麼測試訊息中即會有 $100*x$ 數量之訊息是送給加權討論室，剩餘 $(1-x)$ 數量的訊息則是均勻分配給剩下的九間討論室。如此一來可以推導出伺服器其查詢討論室內使用者名單與傳送訊息的處理成本(往後稱為推導之處理成本)應如下公式。

$$y = x^2 + (1-x)^2 * 9$$

$$z = a / z$$

a = 一常數係數，為推導之處理成本與實測處理成本之間的比值

x = 加權討論室之佔有率， y = 推導之處理成本

而處理成本應與伺服器處理能力成反比，因此加權討論室之佔有率與伺服器處理能力如表 4-X 所示(註：此處伺服器處理能力為估算值，非實際之處理訊息量)。

表 4-7: 加權討論室之佔有率與推導出的伺服器處理能力對應表

加權討論室之佔有率	推導出的伺服器處理能力	加權討論室之佔有率	推導出的伺服器處理能力
10 (%)	$10 * a$	60 (%)	$3.60 * a$
20 (%)	$9.11 * a$	70 (%)	$2.00 * a$
30 (%)	$6.97 * a$	80 (%)	$1.55 * a$
40 (%)	$5.01 * a$	90 (%)	$1.23 * a$
50 (%)	$0.36 * a$		

當加權討論室之佔有率為 10%時，推導出的伺服器處理能力為一秒鐘可處理 $10 * a$ 個訊息，而實際測試到的數據為伺服器一秒鐘可處理 1097 個訊息。若是將 a 代入 109.7，將會得到圖 4-10 之紅色曲線為推導出的伺服器處理能力，藍色曲線則是實際測試到的伺服器處理能力。

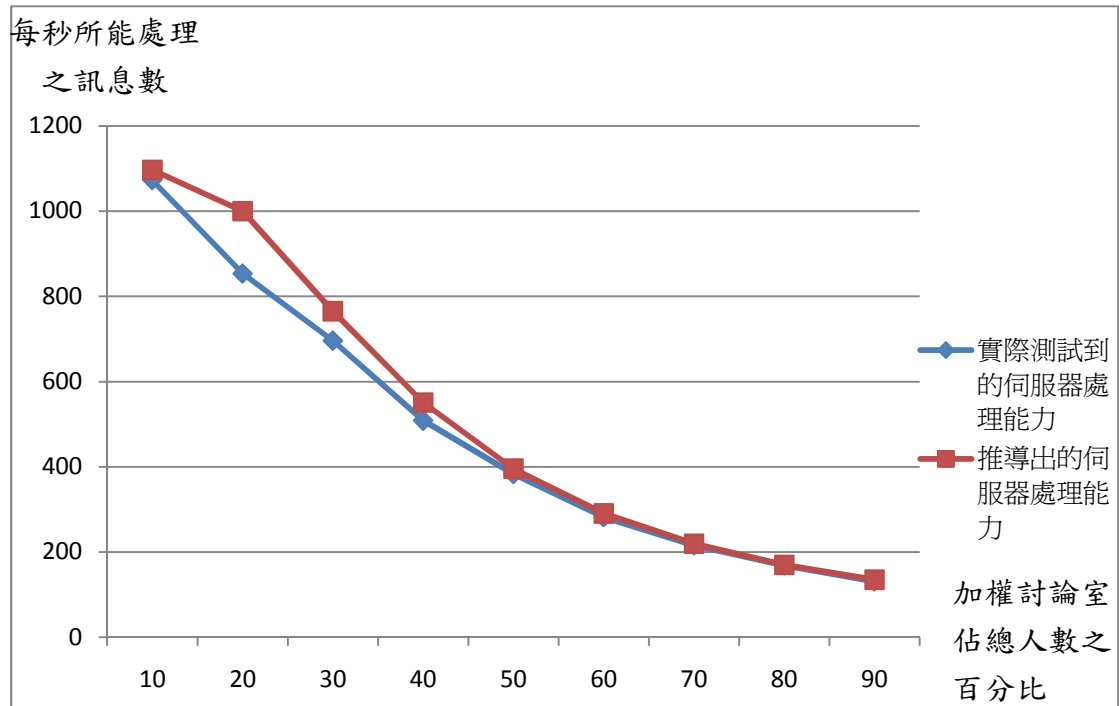


圖 4-10: 在伺服器承載人數非均勻分布在討論室時，推導出的伺服器處理能力數據與實際數據之比較圖

觀察圖 4-10 之紅色曲線與藍色曲線可以發現這兩條曲線大部分皆吻合，代表實際測試數據符合前面所推導出之公式。在使用者過度集中於某討論室時，伺服器每秒所能處理之訊息數與使用者平均分散在各討論室相比將大幅下降。因此決定對每個討論室之使用者人數做出限制。

子測試 D

量測數據：

此測試在各種測試訊息長度下，當伺服器每秒處理的訊息數量達到其最多能處理訊息數量時，伺服器之 CPU 使用率均達到 100%，記憶體與網路流量皆尚未到達極限，因此在此測試時 CPU 為伺服器之瓶頸所在。

圖 4-11 為測試訊息長度與訊息傳遞伺服器每秒所能處理訊息數量之關係曲線。橫軸代表測試訊息為多少字元。縱軸為伺服器每秒最多能處理的訊息數。

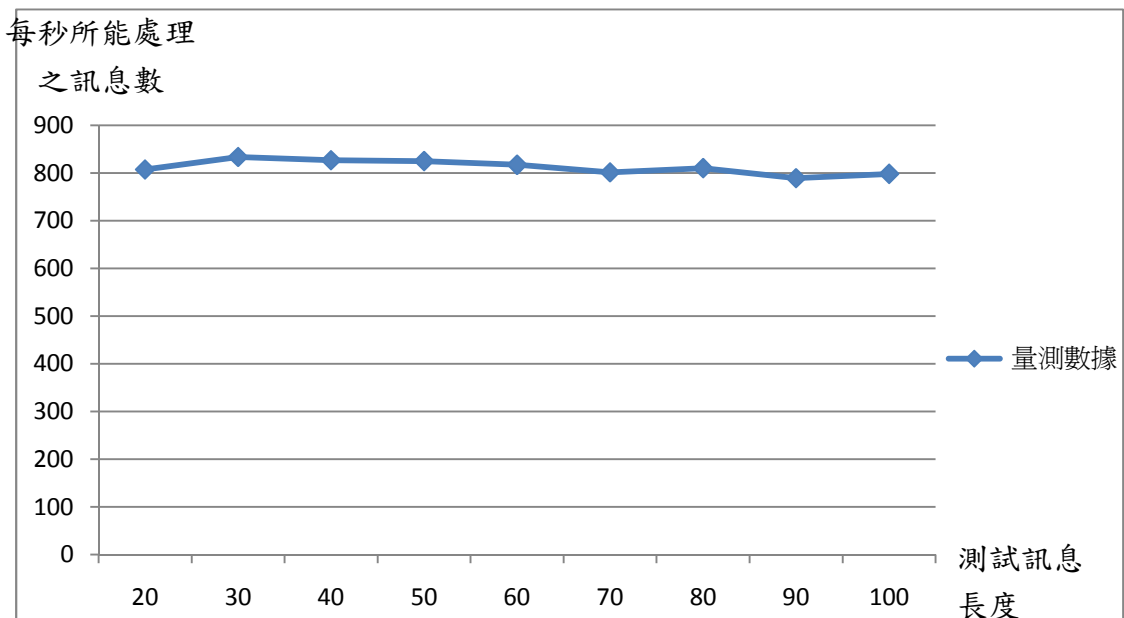


圖 4-11：測試訊息長度對伺服器處理能力影響之關係曲線圖。

數據分析：

由圖 4-11 可以發現，測試訊息的長度與訊息傳遞伺服器每秒所能處理訊息數量之關係曲線的趨勢約是呈現一個線性的持平直線。這代表了測試訊息的長度幾乎不會影響訊息傳遞伺服器每秒所能處理訊息數量。因此畫筆標誌等長度特長之訊息其數目在多也不會影響伺服器每秒所能觸裡之訊息數。

4.3.2 寫入策略對訊息傳遞伺服器之訊息處理能力影響

本節欲測試寫入策略對於訊息傳遞伺服器之訊息處理能力之影響，共分為兩個子測試。

子測試 E.

寫入暫存佇列長度對於訊息傳遞伺服器處理能力的影響：

在固定的伺服器承載人數均勻分布在固定數量的討論室中時，改變寫入暫存佇列長度，觀察訊息傳遞伺服器每秒最多能處理多少訊息數。

子測試 F.

在同樣寫入暫存佇列長度時，所要寫入的文字檔案數量(與討論室為一一對應量)時對於訊息傳遞伺服器處理能力的影響：

固定每個討論室中的使用者人數，在討論室數量增加時，觀察訊息傳遞伺服器每秒最多能處理多少訊息數。

由於寫入策略其影響的部分為硬碟之寫入方面。因此在子測試 D、E 的參數設定上，每個討論室內的使用者人數以及討論室數量遠比子測試 A、B、C 的少。目的是避免因廣播機制所造成的 CPU 滿載，導致伺服器每秒最多能處理多少訊息數下降，反應不出寫入策略的效果。

測試方法：

上述兩項測試中伺服器的參數設定如下，表 4-8 為子測試 E 之參數設定。表 4-9 為子測試 F 之參數設定。

表 4-8：寫入暫存佇列長度對於訊息傳遞伺服器處理能力影響之參數設定表

寫入暫存佇列長度	討論室數量	每個討論室內之使用者人數
10	10	2
50	10	2
100	10	2
150	10	2
200	10	2
250	10	2
500	10	2
750	10	2
1000	10	2
2000	10	2
3000	10	2
4000	10	2
5000	10	2

表 4-9: 在同樣寫入暫存佇列長度時，改變所要寫入的文字檔案數量時對於訊息傳遞伺服器處理能力影響之參數設定表

所要寫入的文字檔案數量	每個討論室內之使用者人數	寫入暫存佇列長度
10	2	1000
20	2	1000
.	.	.
.	.	.
.	.	.
100	2	1000

測試軟體之設定與 4.3.1 同。



測試數據與分析討論：

子測試 E

量測數據：

此測試在所設定之測試條件下，當伺服器每秒處理的訊息數量達到其最多能處理訊息數量時，伺服器之 CPU、記憶體、網路流量皆尚未到達極限。

圖 4-12 與表 4-10 為寫入暫存佇列長度改變時對於訊息傳遞伺服器處理能力之關係成長數據。橫軸代表寫入暫存佇列長度。縱軸為伺服器每秒最多能處理的訊息數。

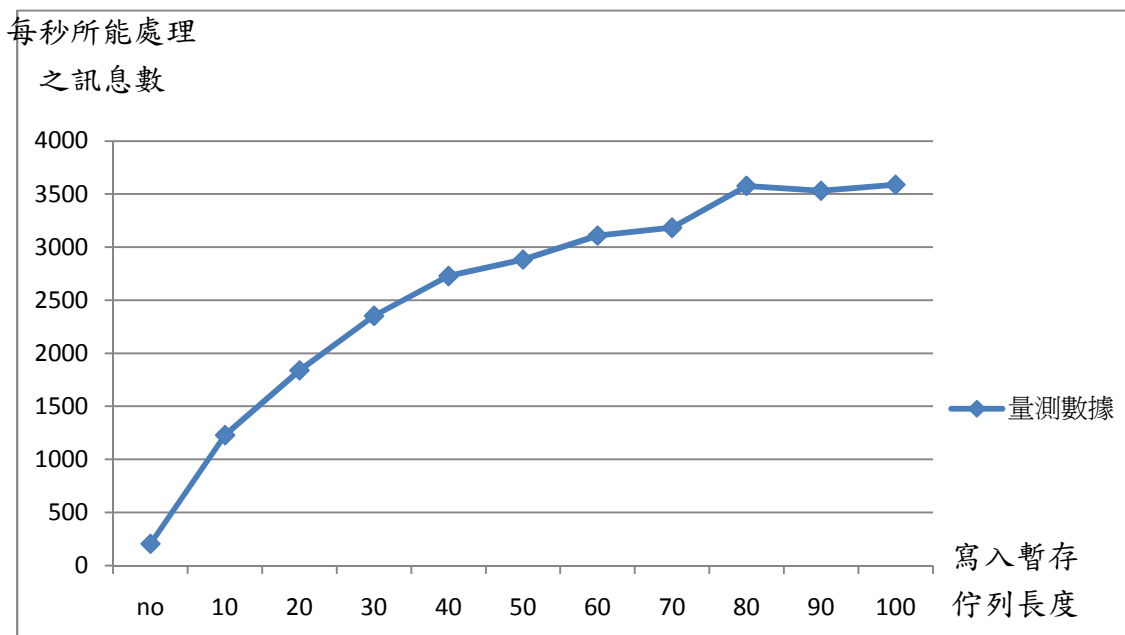


圖 4-12：寫入暫存佇列長度對於訊息傳遞伺服器處理能力之關係曲線圖

表 4-10：寫入暫存佇列長度超過 100 後對於訊息傳遞伺服器處理能力之關係表

寫入暫存佇列長度	伺服器每秒最多能處理的訊息數
150	3875
200	3937
250	3921
500	4219
750	4464
1000	4506
2000	4631
3000	4520
4000	4629
5000	4694

數據分析：

1. 由圖 4-12 可見寫入暫存佇列長度在所設定之參數下對每秒能處理的訊息數為正相關。原因在於把原先需分多筆寫入的訊息聚合至同一批次內一次寫入，將原先之隨機寫入行為轉化為趨近於循序寫入行為，降低整體寫入所需的時間。
2. 由表 4-10 可見過長的寫入暫存佇列長度其對伺服器處理訊息數量的提升效果並不明顯，同時很明顯地會增加記憶體的使用量，因此在實作上將訊息傳遞伺服器之寫入暫存佇列長度取一折衷值 1000。

子測試 F

量測數據：

此測試在所設定之測試條件下，當伺服器每秒處理的訊息數量達到其最多能處理訊息數量時，伺服器之 CPU、記憶體、網路流量皆尚未到達極限。

圖 4-13 為在同樣寫入暫存佇列長度時，改變所要寫入的文字檔案數量時對於訊息傳遞伺服器處理能力之關係成長曲線。橫軸代表所要寫入的文字檔案數量。縱軸為伺服器每秒最多能處理的訊息數。

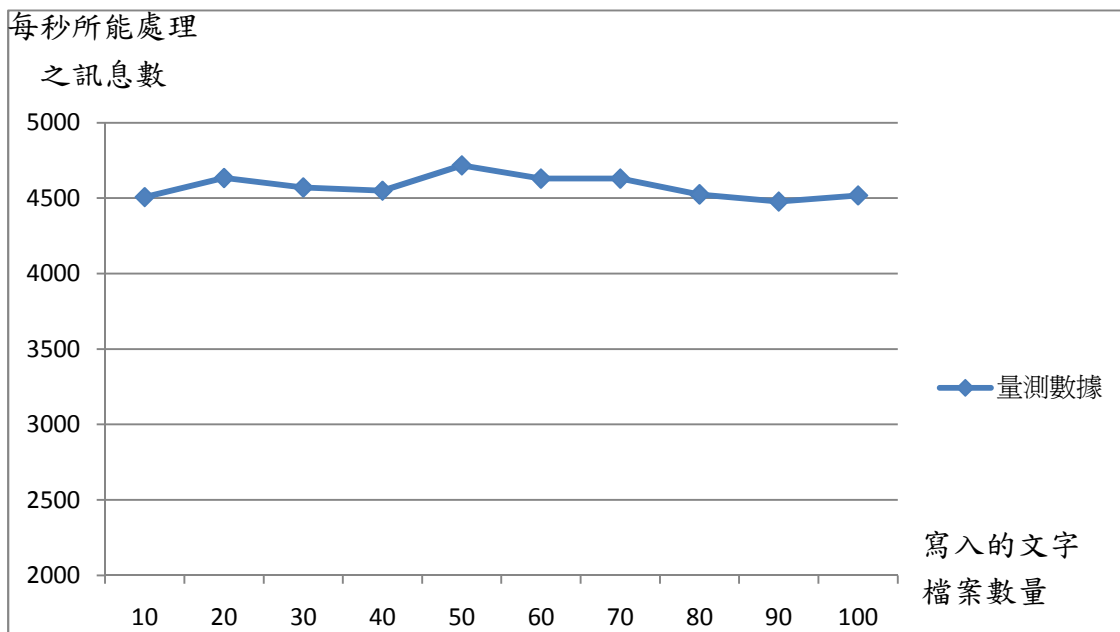


圖 4-13: 所要寫入的目標數量時對於訊息傳遞伺服器處理能力之關係曲線圖

數據分析：

由圖 4-13 可以發現，所要寫入的文字檔案數量與訊息傳遞伺服器每秒所能處理訊息數量之關係曲線的趨勢約是呈現一個線性的持平直線。

這代表了所要寫入的文字檔案數量幾乎不會影響訊息傳遞伺服器每秒所能處理訊息數量。

4.3.3 分散機制對訊息傳遞伺服器服務人數之提升

依 3.4 所提之分散架構，增加訊息傳遞伺服器個數，測試分散式系統每秒最多所能處理之數量。伺服器參數設定皆為：伺服器內共 50 個討論室，每個討論室內人數皆為 100 人，寫入暫存佇列長度為 1000 個訊息。

圖 4-14 為在分散架構下之訊息傳遞伺服器台數與平台服務人數之關係圖，橫軸代表訊息傳遞伺服器數量，縱軸代表分散式系統每秒最多所能處理之訊息數量。

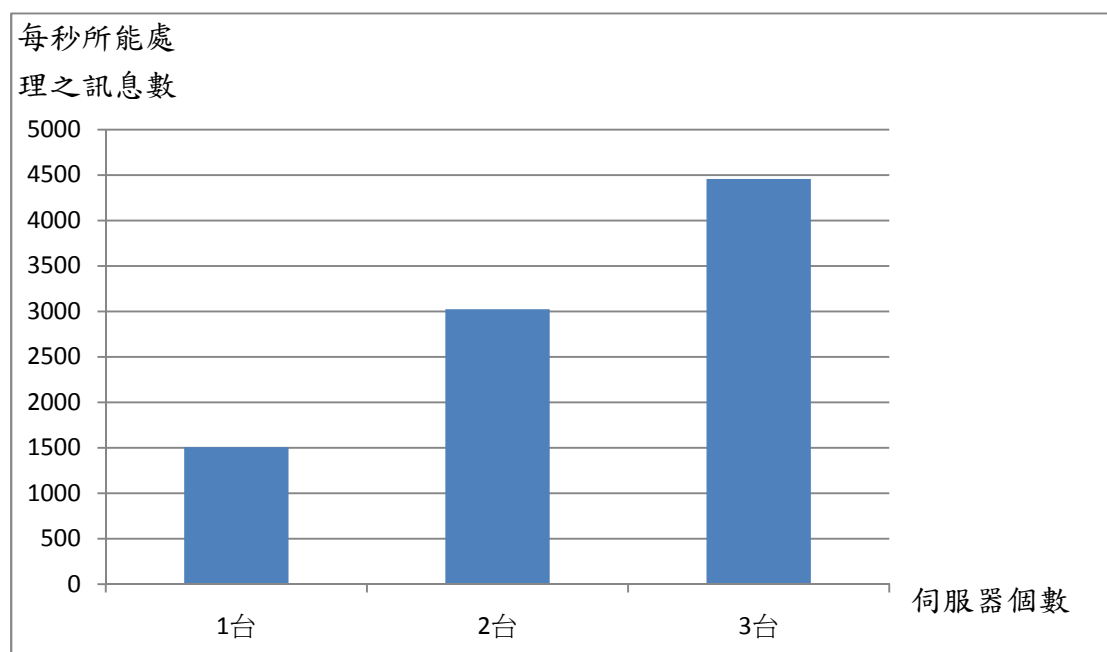


圖 4-14：分散架構下之訊息傳遞伺服器台數與分散式系統每秒最多所能處理之數量之關係圖

由圖 4-14 可以發現增加分散式架構中之訊息傳遞伺服器個數確實可以提升分散式系統中每秒最多所能處理之訊息數量。

4.3.4 分散架構下訊息傳遞伺服器之故障處理成功率

依章節 3.4 故障處理之機制，測試當某訊息傳遞伺服器故障時，在其上之使用者是否正確的轉接到新分配之伺服器並進入原編號之討論室。

測試方法為在欲測試之訊息傳遞伺服器上，配置一定數量之使用者且分散在不同編號之討論室，而後手動將該訊息傳遞伺服器關閉，觀察其內之使用者有無順利導向至新的訊息傳遞伺服器以及進入正確的原編號討論室。以上步驟重複測試 N 次，統計其導向新訊息傳遞伺服器成功率以及新進討論室正確率。

註：導向新訊息傳遞伺服器成功率 = 導向新伺服器成功數 / 導向總數

新進聊天室正確率 = 新進討論室正確數 / 新進總數

表 4-11: 故障處理測試結果

測試人數	新伺服器 導向成功數	新進討論室 正確數	新伺服器 導向成功率	新進討論室 正確率
50	50	50	100%	100%

由表 4-11 可以得知故障處理皆正確運行，沒有發生導向錯誤之情況。

第五章 結論與未來展望

5.1 結論

本論文所實作之訊息傳遞伺服器，為雲端服務視訊互助平台提供了穩定、快速的訊息傳遞服務。並且藉由實施各式策略與機制來提升其所能服務的使用者人數，在限制每間討論室人數上限為 100 人時單台訊息傳遞伺服器可以乘載一萬左右的使用者，同時也能順利的紀錄訊息。往後雲端服務視訊互助平台的使用者人數若是持續成長，可依分散架構進行伺服器的擴充，維持服務品質。

5.2 未來展望

在訊息傳遞伺服器中增加翻譯功能，在討論室中提供使用者即時文字翻譯服務，如此一來使用者可用其擅長之語言與其他來自世界各地的使用者進行文字溝通而不受語言之隔閡影響。將雲端服務視訊互助平台推廣至世界各處。



參考文獻

- [1] 戴永章. 2011. 雲端服務視訊互助平台之網站架構研究.
碩士論文. 新竹：國立交通大學資工學系
- [2] 林金昇. 2011. 雲端服務視訊互助平台之互動影音服務研究.
碩士論文. 新竹：國立交通大學資工學系
- [3] 翁仕全. 2005. 線上即時傳呼系統的設計與效能分析研究
碩士論文. 新竹：國立交通大學資工學系
- [4] 蔡銘韓. 2003. 線上即時訊息系統之研究.
碩士論文. 新竹：國立交通大學資工學系
- [5] 黃旭清. 2001. 網路服務應用環境整合之研究與實作.
碩士論文. 台北：國立台灣大學資訊工程學系
- [6] Alexander Libman with Vladimir Gilbourd. 2005. Comparing Two High-Performance I/O Design Patterns.
- [7] How Windows Live Messenger works.
<http://www.milkaddict.com/how-windows-live-messenger-works/>
- [8] own3d.tv. <http://www.own3d.tv/>
- [9] justin.tv. <http://zh-tw.justin.tv/>
- [10] Wikipedia. <http://www.wikipedia.org/>
- [11] Beej's Guide to Network Programming. <http://beej.us/guide/bgnet/>
- [12] James Greenfield. The Rox Java Nio Tutorial.
<http://rox-xmlrpc.sourceforge.net/niotut/>

附錄：

使用者在視訊連線活動中之身分

在視訊連線活動中，會因為使用者所扮演之角色不同，而有不一樣的功能及權限，以下將針對連線活動中使用者的功能及權限詳細說明：

創造者：啟動連線及受邀參與連線的使用者皆是創造者。在視訊連線時創造者具有收發影音資訊、畫筆標誌、文字溝通之權利，並能於連線結束後保有影音紀錄的權利。

管理者：預設由發起連線的使用者擔任，具有隨時更改活動資訊的權利，當管理者離開時會從創造者中選出下一位繼承者。

旁觀者：自行進入連線活動之使用者。使用者僅能單方接收影音與畫筆資訊，可使用文字與其他線上使用者溝通。

受邀旁觀者：於連線活動建立後，接受邀請進入的旁觀者，具有收發影音資訊、畫筆標誌、文字溝通之權利，但連線結束後無法保有影音紀錄。

附錄表為活動中各角色之對應權力，受邀旁觀者與旁觀者還有一最大差異為，當受邀旁觀者進入視訊連線時，等同於處於連線狀態，不得再新建立與受邀至其他活動中，而旁觀者則無此方面限制。目前一則連線活動只允許 10 人有影音發佈權，旁觀者數量則沒有限制。

附錄表：連線活動中各身分的權力區分

	影音發佈	畫筆標誌	文字溝通	記錄保存	連線資訊更改
管理者	Yes	Yes	Yes	Yes	Yes
創造者	Yes	Yes	Yes	Yes	No
受邀旁觀者	Yes	Yes	Yes	No	No
旁觀者	No	No	Yes	No	No