

**99 學年度大學校院
網路通訊軟體與創意應用競賽**

**結合雲端分散儲存與 Nutch 搜尋
引擎之影音網站**

參賽組別：網際應用與服務組

作品編號：WC SOA-3-21

參賽隊員：李哲成、駱彥呈、鄭陳嶸、梁峻瑞

中華民國 100 年 04 月 07 日

作品報告書

摘要

一、作品設計動機與目的

最近幾年來雲端運算被視作是最熱門的技術之一，對於雲端運算運作的方式與理論基礎我們深感興趣，有鑑於此，我們希望能透過參與競賽的機會，利用開放原始碼的軟體套件做系統整合，把雲端架構運用在影音服務上面，改進其搜尋的效能、使用雲端分散式儲存並利用雲端分散式轉檔，達成雲端運算的實體應用。

我們的最終目標，是希望提供各中小型企業、組織或各級學校系所，可以自由下載並快速且簡單的建立私有影音雲服務平台，提供他人線上觀看影片，使用者可以將影片存放在雲端的硬碟空間，可以快速的分享影片連結給其他人用多種端裝置觀看。

二、系統功能介紹

本系統實作於網頁上，網址為：<http://www.voc.idv.tw/>

1. 首頁



Figure 1 首頁

右上角紅色框中分別為種類選擇按鈕、登入按鈕和註冊按鈕，而中間紅色框中則為搜尋框。如果已經有申請帳號且已通過認證，則可以按 Login 來登入，而想註冊新帳號者可按 Register 來註冊新帳號，沒有登入者只能觀看影片，留言及觀看影片需登入後方可執行。

2. 註冊新帳號

Figure 2 註冊頁面

點擊 Register 後進入此頁面，而後將項目填寫完畢之後按下 Submit Form，如無其他錯誤則會出現 Figure 3 的頁面。

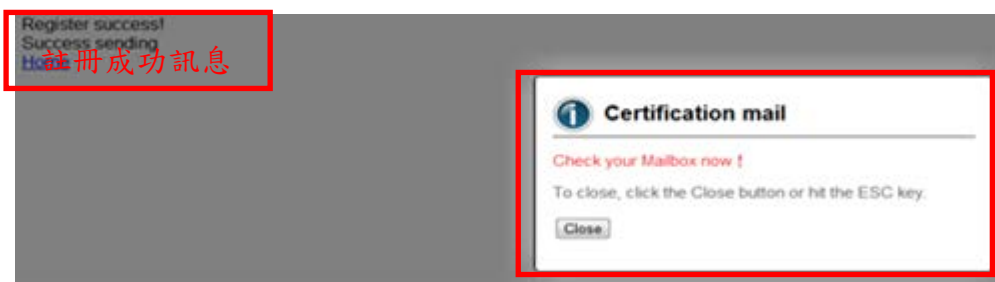


Figure 3 註冊成功訊息及認證信發送

出現類似 Figure 3 的頁面時代表註冊成功，且系統已寄出認證信，然後就去剛剛註冊時輸入的信箱收信，點擊信中的連結後就會到 Figure 4 頁面完成認證。

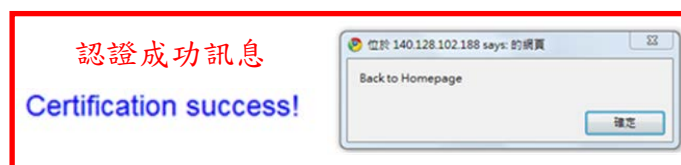


Figure 4 認證成功

3. 登入/上傳影片

登入與否的差別在於是否能上傳影片以及在影片下方留言，未登入者依然能觀看影片。

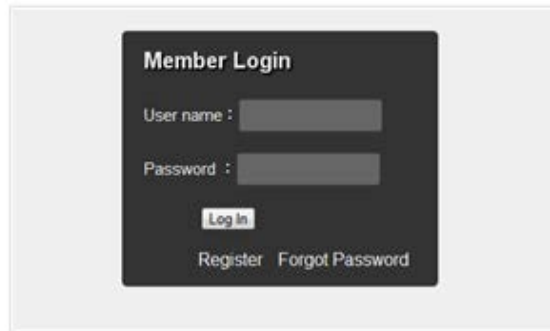


Figure 5 登入頁面

按下 Login 後會到達此頁面，然後分別在兩個欄位中輸入之前註冊時的帳號密碼，輸入完成後按下 Log In 按鈕即可登入。

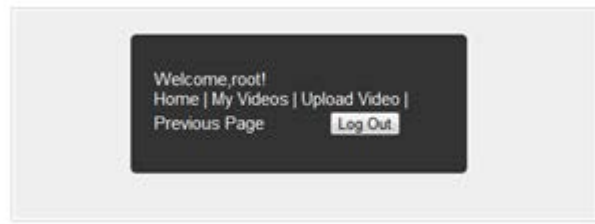


Figure 6 會員頁面

登入後會到達 Figure 6 的頁面，在這裡可以選擇要上傳影片，也可以去看看自己上傳了哪些影片。

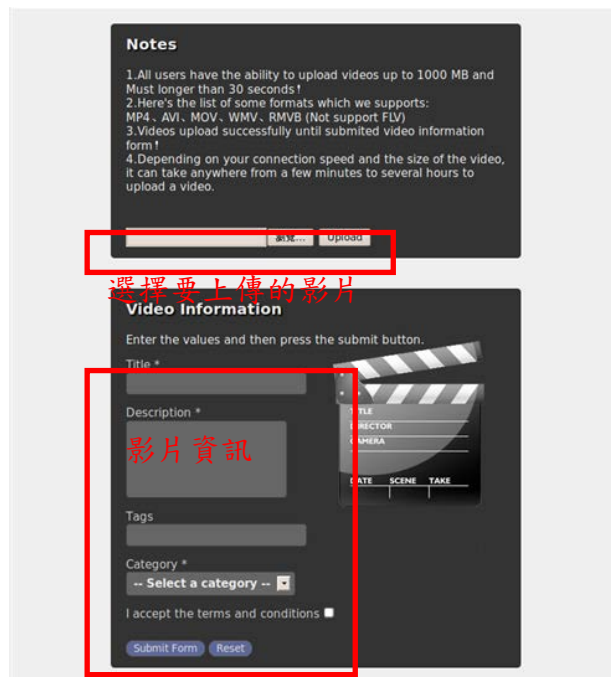


Figure 7 上傳頁面

在 Figure 6 的頁面中按下 Upload Video 會到達 Figure 7 的頁面，在此頁面中可以選擇要上傳的影片，並且可以描述這影片的特性。

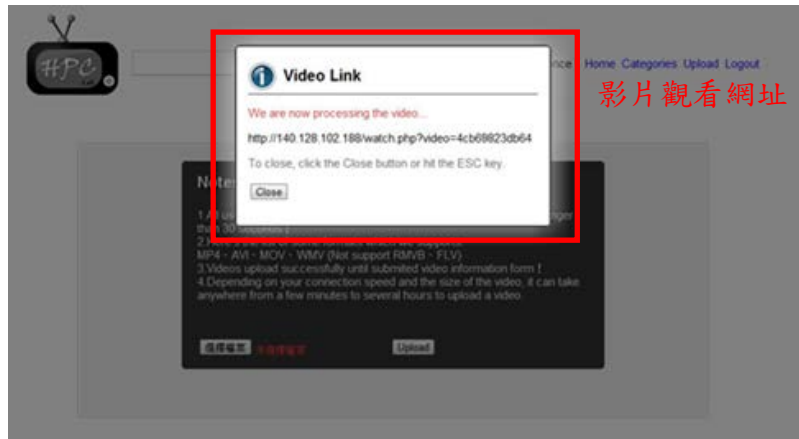


Figure 8 影片上傳完成

等到影片上傳完成後會出現 Figure 8 的頁面，之後就可由該頁面所顯示出的連結網址到影片的頁面來觀看影片。

4. 影片播放/留言



Figure 9 影片播放頁面

在此頁面中可以觀看上傳的影片，影片由 flash 播放器來播放的，影片比例為 16:9，影片解析度固定為 720P(1280x720)，如上傳其餘原始解析度則會自動放大或縮小至此解析度，此外亦可放大至全螢幕觀看，影片下方則提供影片相關資訊。



Figure 10 留言板

在影片播放頁面下方有著像 Figure 10 一樣的留言板，只要是已登入的會員就可以在此針對該影片做留言，而留言的字數最多 300 字，寫完留言後再依據驗證碼的圖案來輸入驗證碼後就可以完成留言。

5. 影片分類/影片錯誤代碼/搜尋頁面

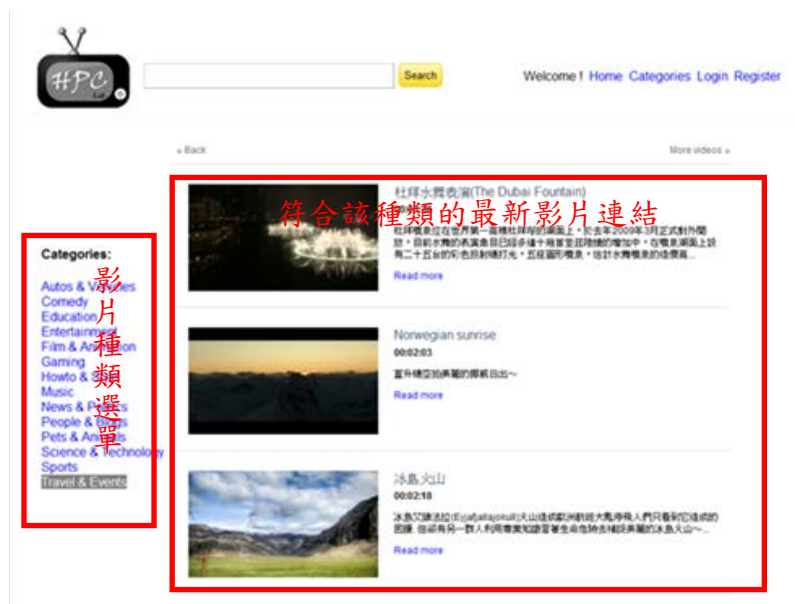


Figure 11 影片分類頁面

由每個頁面右上方的 Categories 可以到達 Figure 11 的頁面，在這頁面中的左方有各種種類，只要按下其中一種右邊就會顯示所有屬於該類型的影片。

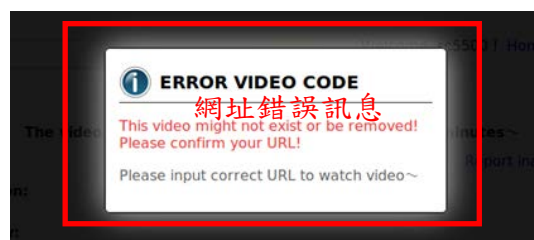


Figure 12 影片錯誤代碼

如果進到觀看影片的頁面時出現此訊息代表該影片不存在或影片已被刪除，又或者是連結輸入錯誤。



Figure 13 搜尋頁面

在任何一個頁面中的搜尋框輸入關鍵字，然後按下 Search 鍵之後就會到類似 Figure 13 的頁面，該頁面所顯示的是所有跟關鍵字相關的內容，點擊搜尋結果就可以去觀看該影片。



Figure 14 影片管理頁面

當登入完之後每個頁面的右上角都會出現一個 MyVideos 的按鍵，按下此按鍵後就會到跟 Figure 14 相似的頁面，此頁面顯示了所有該使用者上傳的影片，在此頁面中可以刪除自己上傳的影片。

三、系統架構

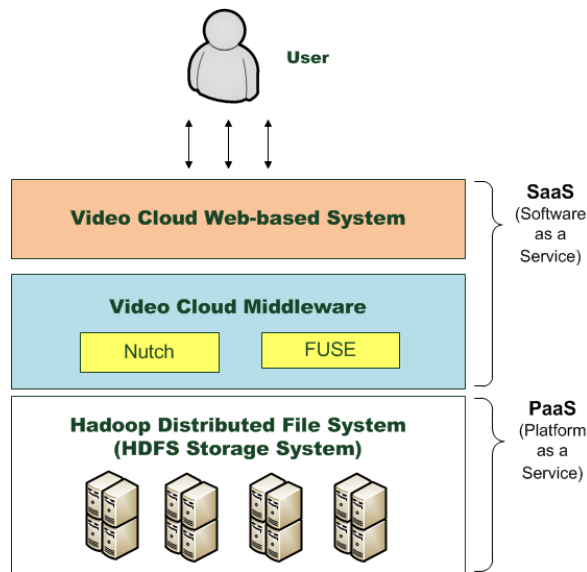


Figure 15 系統雲端架構圖

Figure 15 為此系統整體架構圖，介紹使用到的主要軟體分別屬於何種雲端架構，同時也介紹網站屬於的服務，使用 Hadoop 作為 PaaS (Platform as a Service)，網站架構作 SaaS (Software as a Service) 的雲端架構，藉由此整體系統來實現雲端運算的應用與服務於線上影片觀賞。

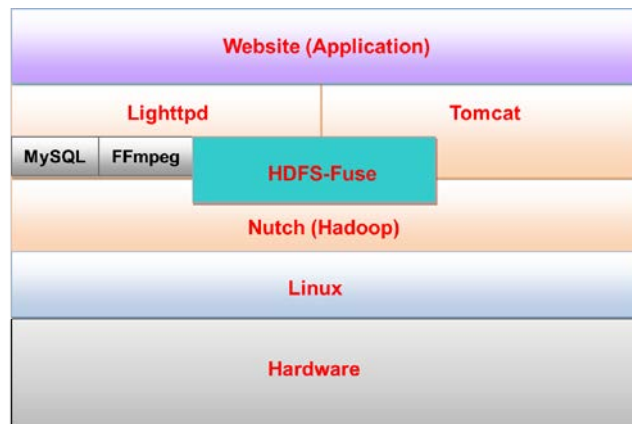


Figure 16 詳細系統架構圖

Figure 16 是詳細的系統架構圖，此圖說明使用到哪些軟體以及各軟體之間的關係為何，接下來將可以藉由此圖來為該系統做更詳細的說明。

本系統完全使用開放原始碼套件來建置，系統架設在 Ubuntu 10.04 Linux 作業系統上，使用 Apache 軟體基金會所研發的開放原始碼的 Hadoop 來運行大量數據運算和建立雲端環境，利用多台主機的叢集環境協同運算，更能徹底發揮所有主機的效能。而在儲存方面，我們運用 FUSE 的虛擬資料夾技術，將上傳影片用的資料夾掛載到 HDFS 中，來完成雲端分散儲存的目標，將資料分段儲存並擁有副本，可防止資料損毀和遺失，並且能充分的利用所有主機的空間，同時也可以使 Hadoop 的儲存系統更具有可用性，上傳以及下載檔案到使用者手中。

網站需要一個網頁伺服器，我們選擇 Lighttpd 做為我們網站的伺服器，除了它是一套免費開放原始碼伺服器之外，它不需耗用主機太多資源的「輕量」特性，特別適合用於影音網站的架設上，也使得測試用的主機不需具備高效運算能力就可順暢的使用並測試網站中各網頁的功能。資料庫方面則是利用 MySQL 來儲存使用者的帳號、密碼和影片的資訊。

Lighttpd 網頁伺服器支援 php 程式語言，於是我們使用 php 程式語言來開發系統應用層網頁介面，並使用 CSS3 和 jQuery 等技術來製作更為人性化的使用者介面，在影片轉檔方面，我們選用 FFmpeg 做為轉檔的軟體工具，它是可自行修改的自由軟體，所以我們加入分散式平行轉檔功能，讓轉檔時間更為快速。而在播放器方面，我們使用開源且支援 H.264 高畫質影音串流播放的 Flowplayer，在網頁伺服器加入相關串流模組，可在觀看影片時拉時間條進行播放，不一定要從頭看到尾。

以下是針對所使用到的軟體更詳細的說明

Hadoop：Hadoop 是由 Apache 軟體基金會所研發開放原始碼的平行運算程式工具和分散式文件系統，與 MapReduce 和 Google 檔案系統的概念類似。以 Java 開發的自由軟體，擁有上千個節點，可處理 Petabyte 等級的資料量，包含了 HDFS、MapReduce。創始者為 Doug Cutting，目前為 Apache 軟體基金會的 top level project。MapReduce 可以將運算分散至其他 node 主機去處理。Hadoop 專案中的檔案系統 HDFS(Hadoop Distributed File System)實現類似 Google File System。一個易於擴充的分散式檔案系統，目的為對大量資料進行分析。能運作於較為廉價的普通硬體之上，且可提供容錯功能。可給大量用戶提供總體性能較高的服務。MapReduce 可以將運算分散至其他 node 主機去處理。

Nutch：Nutch 是一個根基於 Lucene Java 為了搜尋和索引元件建立的開放原始碼搜尋引擎成果。基本上，它是建立在 Hadoop 之上，利用 HDFS 作為儲存搜尋索引的資料庫，並且運用 Map/Reduce 的分散式運算來搜尋索引，進而搜尋到想要的資料，它即是雲端計算的使用實例。由於 Nutch 為開放原始碼，因此使用者可以任意修改內容，成為客製化的搜尋引擎，人人皆可量身訂做自己獨特風格的搜尋引擎。

Lighttpd：Lighttpd（發音為 lighty）是一套開放原始碼的網頁伺服器，以 BSD 許可證釋出。相較於其他的網頁伺服器，lighttpd 僅需少量的記憶體及 CPU 資源即可達到同樣的效能。

FFmpeg：FFmpeg 是一個自由軟體(Free Software)，可以執行音訊和視訊多種格式的錄影、轉檔、串流功能，包含了 libavcodec —這是一個用於多個專案中音訊和視訊的解碼器函式庫，以及 libavformat —一個音訊與視訊格式轉換函式庫。

FUSE：FUSE(Filesystem in Userspace，中文直譯：使用者空間檔案系統)是對於類 Unix 電腦作業系統的一個可負載 kernel 模組，它可以讓未經授權的使用者創造他們自己的檔案系統而無需編輯 kernel 碼。當 FUSE 模組提供一個「橋樑」給

實際的 kernel 介面時，這可以藉由在使用者空間執行 Hadoop HDFS 檔案系統碼來達成。

Hadoop 與 MPI 在資料處理上的差異主要表現在資料儲存與資料處理在系統中位置不同(如圖 4 所示)。MPI 是計算與儲存分離，Hadoop 是計算向儲存遷移。在 MPI 中資料儲存的節點和資料處理的節點往往是不同的，一般在每次計算開始時 MPI 需要從資料儲存節點讀取需要處理的資料分配給各個計算節點對資料進行處理，因此 MPI 中資料儲存和資料處理是分離的。對於計算密集型的應用 MPI 能表現出良好的性能，但對於處理 TB 級資料的資料密集型應用由於網路資料傳輸速度很慢，MPI 的性能會大大降低，甚至會到不可忍受的地步，所以對於建構在 MPI 上的平行計算系統網路通訊速度一直是一個重要的性能指標，用計算換通訊也是 MPI 平行程式設計中的基本原則。

在 Hadoop 中由於有 HDFS 檔案系統的支援，資料是分散式儲存在各個節點的，計算時各節點讀取儲存在自己節點的資料進行處理，從而避免了大量資料在網路上的傳遞，實現計算向儲存的遷移。處理 TB 級的巨量資料這種方式有很大的優勢，這也是為何 Hadoop 架構在搜尋引擎系統中有的良好執行的原因，因為搜尋引擎系統是最典型的資料密集型應用，需要對驚人數量的網頁數據進行管理和處理。

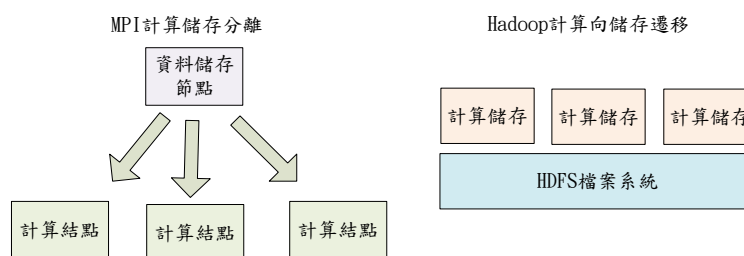


Figure 17 Hadoop 與 MPI 在資料處理上的差異

Hadoop 從上層架構上看是一種典型的主從式結構，主從式的結構在 MPI 平行程式設計中也是一種重要的平行程式設計方法，主節點負責對整個系統的資料和工作進行管理和分發。而 Hadoop 與 MPI 最根本的區別在於 Hadoop 還有一個主從式的檔案系統 HDFS 在底層支撐其 Map/Reduce 的資料處理功能。HDFS 檔案系統在 Hadoop 中可以看作是它的基礎架構，由於有了 HDFS，Hadoop 可以方便地實現『計算向資料儲存位置的遷移』這一策略，從而大大提昇了系統計算效率。主從式基礎儲存和主從式資料處理構成了 Hadoop 的基本架構模型(如圖 5 所示)。

HDFS 檔案系統是一種典型的主從式分散式檔案系統，該檔案系統完全是仿照 Google 的 GFS 檔案系統而設計的，HDFS 的架構如圖 6 所示。HDFS 由一個名叫 Name node 的主節點和多個名叫 Data node 的子節點組成。Name node 儲存著檔案系統的中繼資料，這些中繼資料包括檔案系統的名字空間等，向使用者映射檔案系統，並負責管理檔案的儲存等服務，但實際的資料並不存放在 Name node。Name node 的作用就像是檔案系統的總指揮，並向存取檔案系統的客戶機提供檔案系統的映射，這種做法並不是 Google 或 Hadoop 的創新，這和傳統平行計算系統中的單一系統映射(Single System Image)的做法相同。HDFS 中的 Data node 用於實際對資料的存放，對 Data node 上資料的存取並不透過 Name node，而是與使用者直接建立資料通信。Hadoop 啟動後我們能看到 Name node 和 Data node 這兩個執行緒。

HDFS 的工作過程是這樣的，使用者請求建立檔案的指令由 Name node 進行

接收，Name node 將儲存資料的 Data node 的 IP 傳回給使用者，並通知其他接收備份的 Data node，由使用者直接與 Data node 進行資料傳送。Name node 同時儲存相關的中繼資料。整個檔案系統採用標準 TCP/IP 協定通訊，實際是架設在 Linux 檔案系統上的一個上層檔案系統。主從式是雲端計算系統的一種典型架構方法，系統透過主節點遮罩底層的複雜結構，並向使用者提供方便的檔案目錄映射。有些改進的主從式架構可能會採用分層的主從式方法，以減輕主節點的負荷。

Hadoop 中的 Map/Reduce 也是以主從式架構的，類以於 HDFS 中主節點 Namenode，Hadoop 在 Map/Reduce 上的主程式被稱為 Jobtracker，它負責整個 Map/Reduce 的控制工作。由於 Jobtracker 需要讀取檔案區塊的資訊，所以 Jobtracker 通常和 Namenode 在同一個節點。Jobtracker 負責建立子節點的從屬主作，在 Hadoop 中從屬工作被稱為 TaskTracker。從屬工作直接在子節點上對資料進行處理，完成計算到儲存的遷移。Tasktracker 將狀態和完成的資訊向 Jobtracker 報告。Hadoop 啟動後我們能看到 Jobtracker 和 TaskTracker 兩個執行緒。

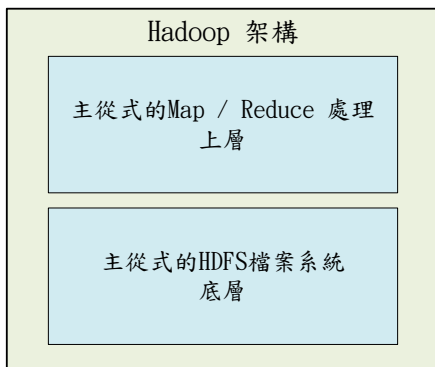


Figure 18 Hadoop 的基本架構

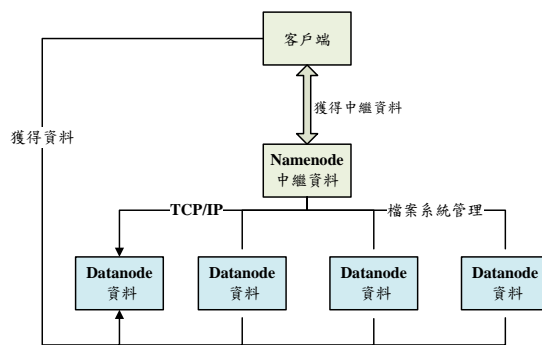


Figure 19 HDFS 的架構

在 Hadoop 中 Map 操作在各節點平行地完成，處理過程中一般沒有資料的傳輸工作，只是在 Reduce 過程中需要向主節點傳送計算結果，對於資料密集型的計算工作這種方法節省了大量的資料傳輸時間。網路資料的傳送一直是平行計算中的一個瓶頸問題，由於網路傳送速率遠小於 CPU 計算速度，我們甚至提出了以計算來換通訊的程式設計策略，但採用計算向儲存遷移的方法後這種情況得到了解決，不過問題在於是否是所有的問題都能實現計算向儲存的遷移，是否有一個較為統一的模型來實現這一方法，Map/Reduce 只是其中的一種解決方案。

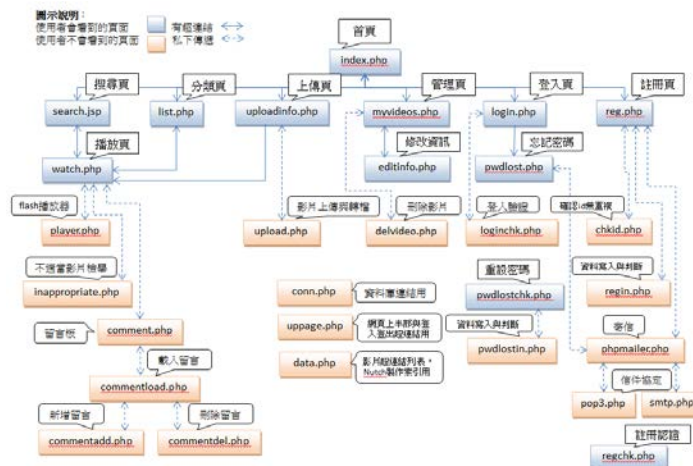


Figure 20 網頁架構圖

Figure 20 是網頁架構圖，這張圖用於展現網路應用層的服務架構，我們將網頁規劃為搜尋、上傳、播放、索引、管理、註冊與登入等部分，並將各部分拆成多個功能頁面進行開發，圖中實線表示使用者可透過超連結前往該分頁，虛線則是用來表示網頁中資料傳遞的方向。

四、結論

從這次的比賽的過程中，我們了解了 Open Source 的各種授權方式，並整合各套 Open Source 軟體提供的資源，結合雲端分散式計算等技術來建立網站。雲端運算的優勢，在於使用者不必了解其背後架構，Server 端可視需求隨時增加電腦數量，並利用叢集和分散式儲存達到加速與備份的目的。

我們使用 Nutch 來做為我們網站的搜尋引擎核心，它做索引資料時，具有將工作分散給 Hadoop 之中設定節點(node)的特性，如此將能降低我們 web server 的使用量，也能充分利用其他台電腦的效能，加上使用多台電腦組成叢集運算，達到快速且低成本的實作，並且叢集環境可應付未來大量的資料搜尋。

我們在設計網頁的過程中，遇到許多有挑戰性的問題，如資料庫的資料表種類和欄位設置，要決定有多少資料表和其要對應多少欄位除了取決於網頁表面的功能外，還要考慮到使用者的行為和其對應的解決方法，如檢舉不良影片和封鎖不肖的使用者等，這真的算是一個腦力激盪的過程。

除了網頁和背後雲端主機的程式碼撰寫外，資訊安全相關的問題更為重要，要防止別人惡意攻擊和取得使用者資料等，在網頁中也對資料庫做了一些基礎的防範，讓使用者可以安心的瀏覽網頁。其它如網頁美化、頁面顯示以及按鈕種類、位置和功能也需參考許多影音網站，並且要有自己獨一無二的風格，如何在這當中取得平衡亦是一項挑戰。

本網站已提供影片社群分享的功能(Facebook、Plunk、Twitter 等)，未來除了原先功能持續的維護外，可望加上一些新奇且實用的功能，朝向更多元服務的網站，例如：遊戲、購物連結等，並建立簡易安裝程序，實現人人可自己架設私有”影音雲”的終極目標邁進。