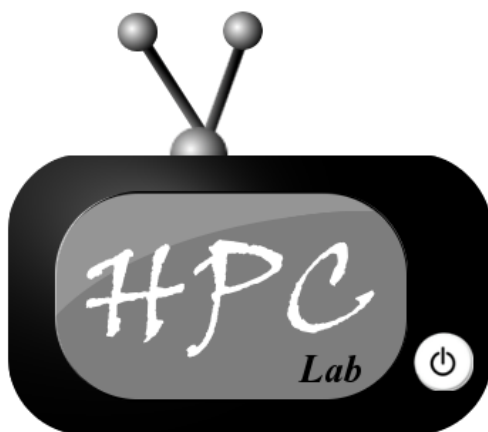


專題書面報告

結合雲端分散儲存與搜尋引擎之影音網站

Video Website with Cloud Distributed Storage and Search Engine



963847 駱彥呈

963808 李哲成

963871 梁峻瑞

963912 鄭陳嶸

963951 劉建安

指導教授：楊朝棟 教授

High-Performance Computing Laboratory
Department of Computer Science
Tunghai University, 40704 Taichung, Taiwan R.O.C.

中華民國 九十九年十二月十八日

目錄

一、摘要	2
二、研究動機與目標	2
三、背景知識介紹	2
四、研究方法及步驟	6
五、系統架構與設計	9
六、安裝與實作流程	12
七、實驗結果與展示	43
八、結論	48
九、未來展望	49
十、參考資料及網站	49

一、摘要

近年來由於全球網路頻寬的提升，大家開始可以互相分享較大的影音檔案，加上雲端運算與儲存的普及，影音網站變得相當熱門。本專題將雲端運算中的 Nutch 搜尋引擎(其內包含 Hadoop)結合影音串流之應用，利用雲端運算的特性，將檔案分散儲存在多台電腦內，並連結多台電腦做平行叢集運算(雲端運算)，來有效縮短搜尋索引庫建置所需花費的時間，讓使用者在影音網站上可以更加快速而精確地搜尋到所要找到的影片。

二、研究動機與目標

(一)為何要做此一專題研究與實驗？

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，我們希望能透過專題研究與實驗的機會，對雲端運算基本概念有所認識，並學習如何使用有關雲端運算的軟體套件，進而把雲端架構運用在影音網站上面，增加其搜尋的效能，達成雲端運算的實體應用。

(二)預計達到的目標

建置一個功能完整且全部使用開源軟體，具備雲端運算特性、具有分散儲存索引搜尋引擎的影音網站，可供一般使用者觀看影片、搜尋影片，而經註冊使用者還可上傳影片和在影片下方留言，並且可以編輯或刪除自己上傳的影片，影片還可連結社群網站(Facebook、Plunk、Twitter)分享。使用者亦可以檢舉不良影片，若影片遭到五人以上檢舉，該影片即會被封鎖，此機制可確保影音網站擁有良好的撥放環境和品質。

三、背景知識介紹

此節將文中所提到的專有名詞(terminology)在此作一概括性的介紹：

Cloud Computing(雲端運算)

- 「雲」指網際網路(Internet)，「端」指使用者端(Client)。
- 大約等於 Network Computing(網路運算)。
- 雲端運算不是技術，而是一種電腦運算的概念。
- 本質上為分散式運算(Distributed Computing)的概念，可視為分散式運算的一種。
- 讓網路上一些(超過一台)不同的電腦同時幫你做事情、進行運算，大幅增進處理速度。
- Paying for what you use.

- 例如：搜尋引擎、網路信箱、智慧型手機(Smart Phone)、衛星導航(GPS)。
- Grid Computing(網格運算)也是分散式運算所延伸出的概念。
- 異質系統之間運算資源的整合。即讓不同等級或作業系統的電腦，彼此間可以透過通訊標準來互相溝通，分享彼此的運算資源。

Hadoop

- Hadoop is a software platform that lets one easily write and run applications that process vast amounts of data.
- 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。

HDFS(Hadoop Distributed File System)

- Hadoop 專案中的檔案系統。
- 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。

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 作為儲存搜尋索引用的資料庫，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，它即是雲端運算的使用實例。

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。

Crawlzilla

Crawlzilla 為修改 Nutch 更進階的客製化搜尋引擎，它簡化了安裝步驟流程和設定，使一般使用者可以更方便的進行安裝使用。

除了增加 GUI(Graphical User Interface：圖形使用者介面)管理頁面外，經過國網中心修改，它亦可直接使用繁體中文介面。

但由於自動設定的地方較多，對於要修改細節部分的使用者反而降低了「客製化」的程度。

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。

特色：

- 提供 FastCGI 及 SCGI 的負載平衡
- 支援 chroot
- 支援 select()/poll()及更有效率的 kqueue/epoll 連線狀態判斷
- 支援條件重寫 (Conditional rewrites)
- 支援 SSL 連線
- 透過 LDAP server 認證
- rrdtool 狀態輸出
- Rule-based downloading
- Server-side includes support
- Virtual hosting
- Modules support
- Cache Meta Language
- Minimal WebDAV support
- Servlet (AJP) support (1.5.x 版後)

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。

"FFmpeg"這個單詞中的 "FF" 指的是 "Fast Forward"。有些新手寫信給 "FFmpeg"的項目負責人，詢問 FF 是不是代表「Fast Free」或者「Fast Fourier」等意思，"FFmpeg"的項目負責人回信說「Just for the record, the original meaning of "FF" in FFmpeg is "Fast Forward"...」

這個項目最初是由 Fabrice Bellard 發起的，而現在是由 Michael Niedermayer 在進行維護。許多 FFmpeg 的開發者同時也是 MPlayer 項目的成員，FFmpeg 在 MPlayer 項目中是被設計為伺服器版本進行開發。

組成元件：

- ffmpeg 是一個命令列工具，用來對視訊檔案轉換格式，也支援對電視卡即時編碼
- ffmpeg 是一個 HTTP 多媒體即時廣播串流伺服器，支援時光平移
- ffmpeg 是一個簡單的播放器，基於 SDL 與 FFmpeg 函式庫
- libavcodec 包含了全部 FFmpeg 音訊／視訊 編解碼函式庫

- libavformat 包含 demuxers 和 muxer 函式庫
- libavutil 包含一些工具函式庫
- libpostproc 對於視訊做前處理的函式庫
- libswscale 對於影像作縮放的函式庫

PHP

PHP (*PHP : Hypertext Preprocessor*) 是一種在電腦上執行的腳本語言，主要用途是在於處理動態網頁，也包含了命令列執行介面 (command line interface)，或者產生圖形使用者介面 (GUI) 程式。

PHP 最早由 Rasmus Lerdorf 在 1995 年發明，而現在 PHP 的標準由 PHP Group 和開放原始碼社群維護。PHP 以 PHP License 作為許可協議，不過因為這個協議限制了 PHP 名稱的使用，所以和開放原始碼許可協議 GPL 不相容。

PHP 的應用範圍相當廣泛，尤其是在網頁程式的開發上。一般來說 PHP 大多執行在網頁伺服器上，透過執行 PHP 程式碼來產生使用者瀏覽的網頁。PHP 可以在多數的伺服器和作業系統上執行，而且使用 PHP 完全是免費的。根據 2007 年 4 月的統計資料，PHP 已經被安裝在超過 2000 萬個網站和 100 萬台伺服器上。

PHP 原本的簡稱為 Personal Home Page，是 Rasmus Lerdorf 為了要維護個人網頁，而用 C 語言開發的一些 CGI 工具程式集，來取代原先使用的 Perl 程式。最初這些工具程式用來顯示 Rasmus Lerdorf 的個人履歷，以及統計網頁流量。他將這些程式和一些表單直譯器整合起來，稱為 PHP/FI。PHP/FI 可以和資料庫連接，產生簡單的動態網頁程式。Rasmus Lerdorf 在 1995 年 6 月 8 日將 PHP/FI 公開釋出，希望可以透過社群來加速程式開發與尋找錯誤。這個釋出的版本命名為 PHP 2，已經有今日 PHP 的一些雛型，像是類似的變數命名方式、表單處理功能、以及嵌入到 HTML 中執行的能力。程式語法上也類似 Perl，有較多的限制，不過更簡單、更有彈性。

JSP

JSP (全稱 JavaServer Pages) 是由 Sun Microsystems 公司倡導和許多公司參與共同建立的一種使軟體開發者可以響應用戶端請求，而動態生成 HTML、XML 或其他格式文檔的 Web 網頁的技術標準。JSP 技術是以 Java 語言作為指令碼語言的，JSP 網頁為整個伺服器端的 Java 函式庫單元提供了一個介面來服務於 HTTP 的應用程式。

JSP 使 Java 代碼和特定的預定義動作可以嵌入到靜態頁面中。JSP 句法增加了被稱為 JSP 動作的 XML 標籤，它們用來呼叫內建功能。另外，可以建立 JSP

標籤函式庫，然後像使用標準 HTML 或 XML 標籤一樣使用它們。標籤函式庫提供了一種和平台無關的擴充功能伺服器性能的方法。

JSP 被 JSP 編譯器編譯成 Java Servlets。一個 JSP 編譯器可以把 JSP 編譯成 JAVA 代碼寫的 servlet 然後再由 JAVA 編譯器來編譯成機器碼，也可以直接編譯成二進制碼。

從架構上說，JSP 可以被看作是從 Servlets 高階提煉而作為 JAVA Servlet 2.1 API 的擴充功能而應用。Servlets 和 JSPs 最早都是由 Sun Microsystems (昇陽公司) 開發的。從 JSP1.2 版本以來，JSP 處於 Java Community Process (有人譯為：JAVA 社群組織) 開發模式下。JSR-53 規定了 JSP 1.2 和 Servlet 2.4 的規範，JSR-152 規定了 JSP 2.0 的規範。2006 年 5 月，JSP 2.1 的規範作為 Java EE 5 的一部份，在 JSR-245 中發布。

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。

四、研究方法及步驟

(一) 初始概念和規劃

一開始我們在各自的電腦上架設多台虛擬機器，全部安裝 Linux 作業系統，並且在多台虛擬機器上安裝 Hadoop 叢集，藉以模擬多台電腦做平行叢集運算(雲端運算)的效果。

接著我們就開始尋找 Hadoop 及包含於子計畫中的 HDFS、MapReduce 能做哪一方面的應用，我們有和研究所的學長和教授討論這個問題，像是人臉辨識、聲音辨識、影片分段載入與字幕搜尋...等想法都有被提出來討論過。而經過可行性、困難度與原創性等多方面的考量後，最終我們決定設計一個影音網站，並將根基於 Hadoop 的 Nutch 搜尋引擎來修改與整合，作為網站搜尋功能的應用，具體呈現具備雲端運算特性儲存和搜尋功能的影音網站。

(二) 系統架設

有了明確的方向後，我們便開始收集相關資料並逐步架設整個系統，由於網站牽扯的層面相當多，因此需要安裝許多各有用處的程式和套件。

首先，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，也使得在測試用的主機不一定要高效運算能力就可順暢的使用並實驗網站中各網頁的功能，因此，它的功能對於做專題研究的我們已經相當足夠了。

Lighttpd 伺服器支援 php 程式語言，於是我們網頁使用 php 來進行開發，在影片轉檔方面，我們選用 FFmpeg 做為轉檔的軟體工具，當然和它是可自行修改的自由軟體有關，如此我們可以用自己的方式使用它的轉檔功能，而在播放器方面，我們安裝了支援影音串流播放的 Flowplayer，可在觀看影片時拉時間條進行播放，不一定要從頭看到尾。

在搜尋引擎方面，我們一開始是架設 Nutch 搜尋引擎來測試，期間也曾嘗試新的 Crawlzilla 搜尋引擎，而在經過一連串的測試與比較後，我們發現 Nutch 和 Crawlzilla 的原理大致相同，而 Crawlzilla 的安裝和設定比較平易近人，相較 Nutch 更具備了 GUI 的管理頁面。

但也因為系統十分方便，許多地方有都幫使用者做自動設定，這樣讓我們想要修改程式內部細節與參數時，反而造成了諸多不便。於是在經過詳細的討論後，我們最後決定使用 Nutch 搜尋引擎來做為我們影音網站的搜尋引擎。

最後在雲端分散儲存方面，我們運用的是 FUSE 的虛擬資料夾技術，將供上傳影片用的資料夾掛載到 HDFS 中，確實完成雲端分散儲存，將資料分段儲存並擁有副本，可防止資料損毀和遺失。

而運用這種機制，儲存資料的 HDFS 所用的 Hadoop 部分可以和 Nutch 共用，不需重新安裝架設另一套 Hadoop，省去許多麻煩，而且經掛載後資料夾的存取使用和其它一般資料夾相同，不需要使用特殊的控制指令或程序來做存取，只需在存取時注意一些權限問題即可，使用起來十分方便。

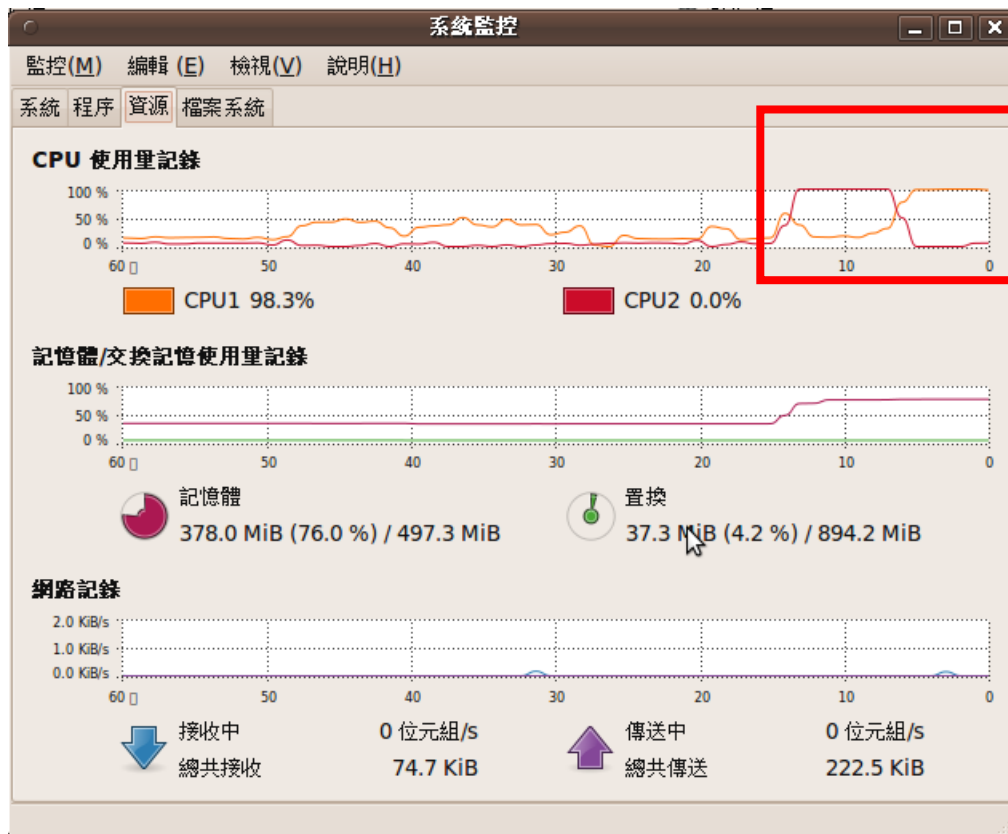
(三)進一步規劃和測試

在網站的各網頁和功能增加時，需要不斷的測試撰寫的新功能是否正常，以及外觀是否符合期望，且不能只在 Internet Explorer 瀏覽器上測試，需要測試其它種類的瀏覽器，如 Mozilla FireFox 和 Google Chrome 上的測試也需正確顯示其外觀和功能；如果功能不齊全或外觀差異過大，版面配置和網頁程式需重新規劃來改進。

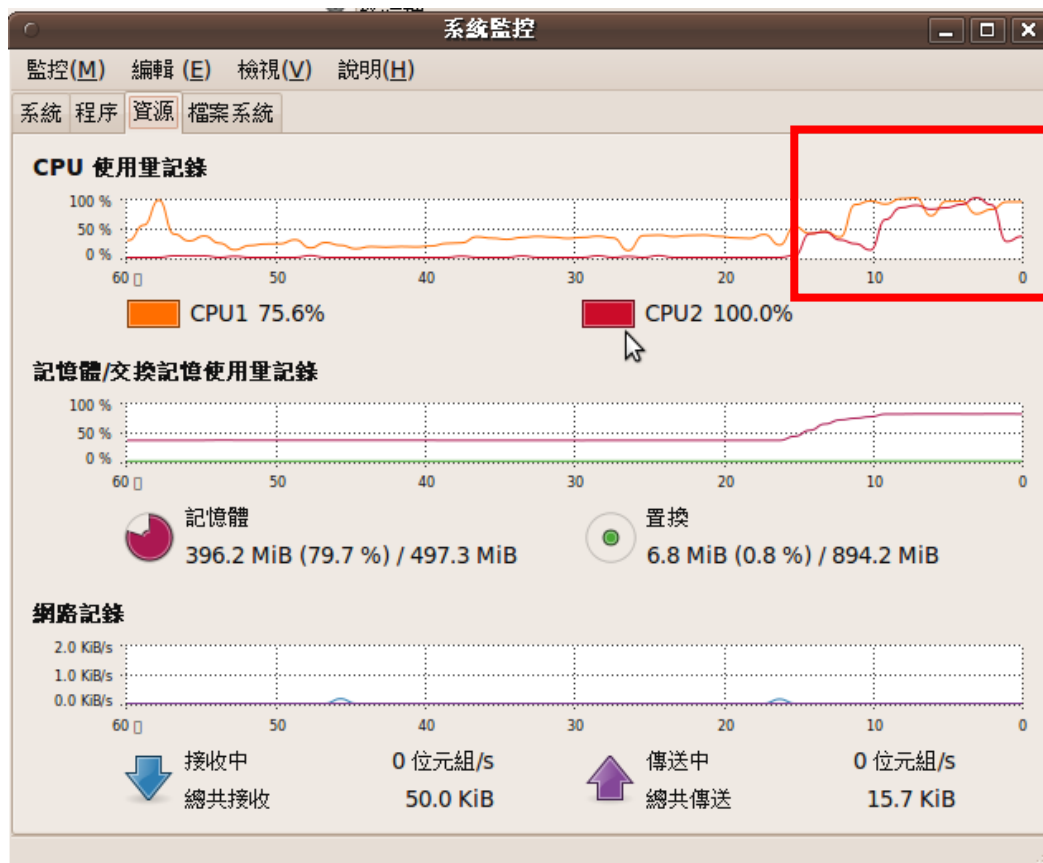
資料庫管理方面，我們使用 mySQL 來做為資料庫，除了檢查其上傳資料是否正確的進入指定的欄位，每當網站功能增加或改變，資料庫的種類和各自欄位都需經過適當的修正，增加新的欄位或改變欄位的名稱，以免和新需求不相符。

套件程式的測試也是很重要的，它們必須要達到我們網站預定要求的功能，例如對於 FFmpeg-mt，在轉檔過程中我們需要測試它是否有別於一般的 FFmpeg，存在多執行緒(多核心)同時運作的特性，如下頁顯示：

1. 一般 FFmpeg



2. FFmpeg-nt



我們發現相較於一般 FFmpeg，同一時間只有單核心獨自運行，一段時間後才交換工作，FFmpeg-mt 在轉檔過程中可多執行緒同時運作，測試結果確實符合我們的預期。

Nutch 搜尋引擎經過修改後，則需要測試搜尋結果是否受到影響，以及我們修改後的搜尋頁面呈現；除了正確列出關鍵字搜尋的項目，搜尋頁面必須要與我們設定的顯示方式相符(包含文字位置、顯示截圖照片…等外觀)，最後要確認它具有定期更新索引庫的能力。

FUSE 儲存方面需要測試影片上傳是否正常，像是上傳後的影片是否可以確實進入資料夾並儲存在 HDFS 中，達到雲端分散儲存的效果；其中在測試時就牽涉到有關資料夾權限、HDFS 上可否進行影片轉檔以及在 HDFS 上的影片是否可以串流播放等問題。經過一些修正後，最後需確認「上傳影片」到「影片可觀看」的整體流程沒有問題存在，可以順暢的一氣呵成。

(四)最後修正與美觀

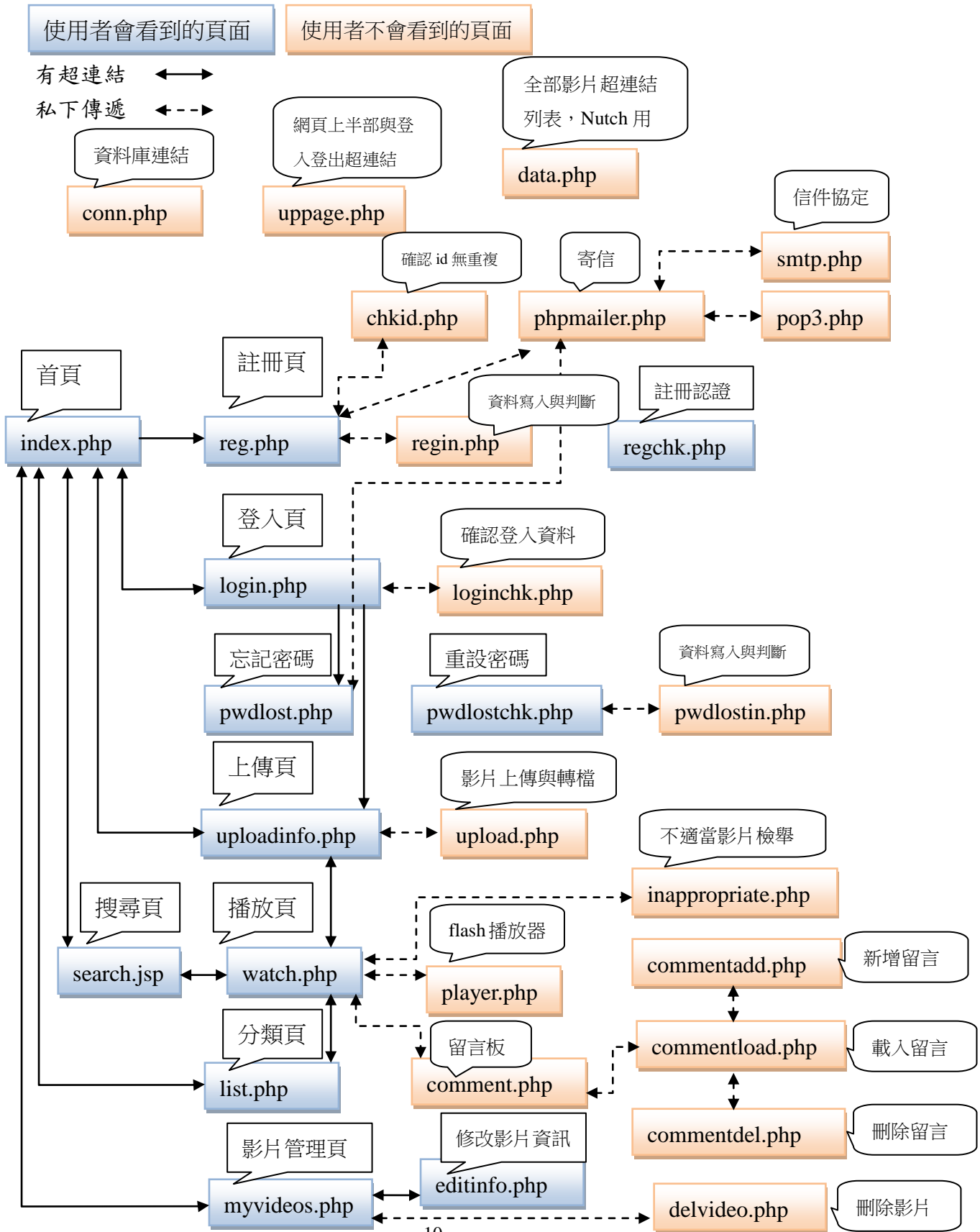
以上工作都完成後，網站的雛型已大致完成，剩下工作只需修正網站各網頁的功能增強、版面配置和外觀美化，以達到我們認為的最佳情形。

例如設計方便註冊使用者編輯和刪除自己上傳影片的功能，可讓使用者更好掌握自己上傳的影片，不一定需要透過管理者，就可以對自己上傳的影片進行標題和注釋的變更，並還可以刪除自己不滿意的上傳影片。對於使用者來說，可說是相當貼心的設計。

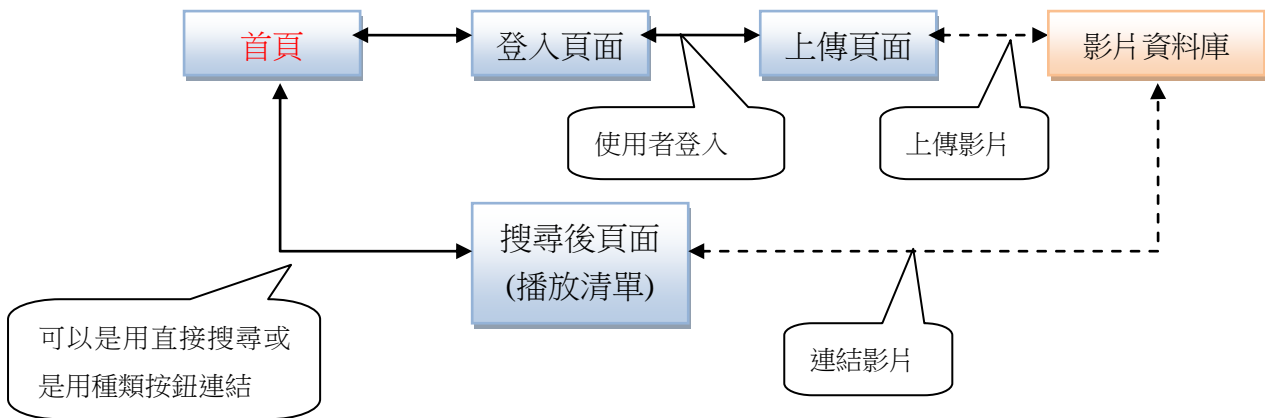
其它方面除了在首頁增加 Logo 之外，設計各個按鈕色彩和位置分配，以及在網頁執行各類工作時加入特效，令使用者登入網站瀏覽時在視覺上有較鮮明的印象，做為我們影音網站的特色。如此，結合雲端分散儲存與搜尋引擎之影音網站即大功告成。

五、系統架構與設計

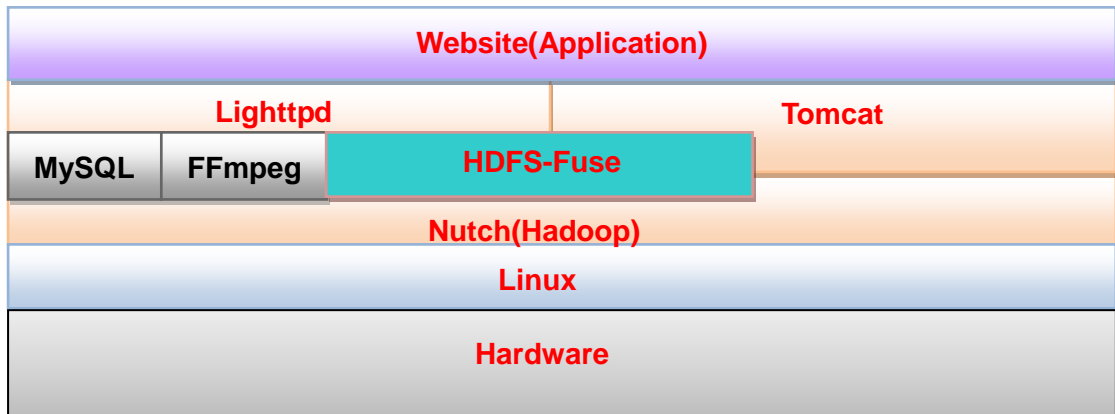
(一) 整體網頁架構：



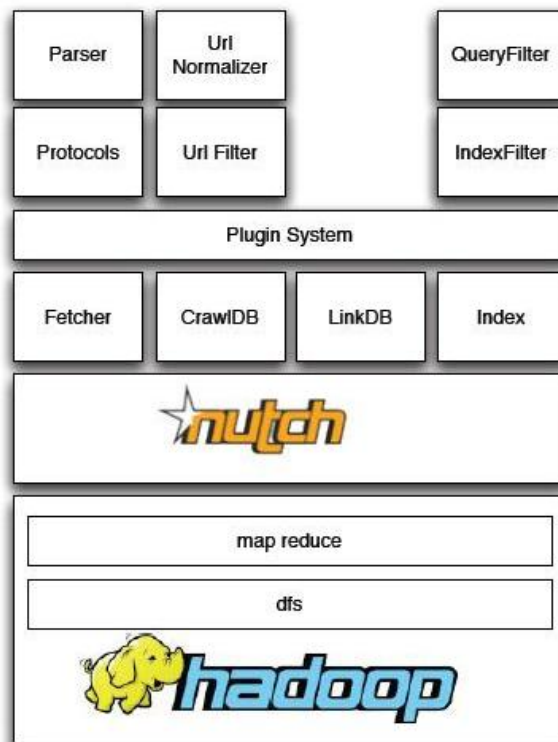
(二) 影片觀看簡易流程架構：



(三) 伺服器系統架構



(四) Nutch 架構



六、安裝與實作流程

(一)前置作業

本次專題實作網頁是使用 PHP 和 JSP 語言來進行開發，採用免費且穩定之 Linux 作業系統作為平台，我們使用的 Linux 發行版本是 Ubuntu 10.04。

(二)安裝 lighttpd

開始先在根目錄下建立一個 lighttpd 資料夾，用來存放接著要下載的壓縮檔及解壓縮的資料夾

```
sudo mkdir -p /lighttpd
```


(加上 home 及 var 資料夾底下的 lighttpd，一共 3 個地方會有 lighttpd 相關的資料)

(以下指令大多為 root 權限，如果 user 權限無法通過，煩請先輸入 sudo su 或在指令之前加上 sudo，更改為 root 權限)

安裝需要的套件(安裝前請先更新套件列表)：

到「系統」→「管理」→「Synaptic 套件管理程式」，搜尋套件名稱，按滑鼠右鍵「標記為安裝」，接著再點選左上角的「套用」即可安裝

zlib1g-dev

 zlib1g-dev	1:1.2.3.3.dfsg-15ub	1:1.2.3.3.dfsg-15ub	compression library - development
--	---------------------	---------------------	-----------------------------------

libbz2-dev

 libbz2-dev	1.0.5-4	1.0.5-4	high-quality block-sorting file compressor library - development
--	---------	---------	--

libpcre++0

libpcre++-dev

S	套件	已安裝版本	最新版本	套件描述
	libpcre++0	0.9.5-3	0.9.5-3	C++ wra
	libpcre++-dev	0.9.5-3	0.9.5-3	C++ wra


libtool

libltdl-dev

automake

php5-cgi

php5-mysql

<input checked="" type="checkbox"/>  php5-mysql	5.3.2-1ubuntu4.2	MySQL module for php5
--	------------------	-----------------------

Lighttpd 上安裝 Mod-H264 Streaming Module

```
cd /lighttpd
sudo wget
http://download.lighttpd.net/lighttpd/releases-1.4.x/lighttpd-1.4.26.tar.gz
sudo tar -zxvf lighttpd-1.4.26.tar.gz
```

```
sudo wget
http://h264.code-shop.com/download/lighttpd-1.4.18_mod_h264_st
reaming-2.2.9.tar.gz
sudo tar -zxvf lighttpd-1.4.18_mod_h264_streaming-2.2.9.tar.gz
```

```
cp lighttpd-1.4.18/src/mod_h264_streaming.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mod_streaming_export.h
lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/moov.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/moov.h lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_io.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_io.h lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_reader.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_reader.h lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_writer.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_writer.h lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_process.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/mp4_process.h lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/output_bucket.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/output_bucket.h lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/output_mp4.c lighttpd-1.4.26/src/
cp lighttpd-1.4.18/src/output_mp4.h lighttpd-1.4.26/src/
```

開啟並修改 Makefile.am :

```
sudo gedit lighttpd-1.4.26/src/Makefile.am
```

在開啟的 Makefile.am 中找尋空白處貼上以下敘述：

```
lib_LTLIBRARIES += mod_h264_streaming.la
mod_h264_streaming_la_SOURCES = mod_h264_streaming.c \
                                mod_streaming_export.h \
                                moov.c moov.h \
                                mp4_io.c mp4_io.h \
                                mp4_reader.c mp4_reader.h \
                                mp4_writer.c mp4_writer.h \
                                mp4_process.c mp4_process.h \
                                output_bucket.c output_bucket.h
\
                                output_mp4.c output_mp4.h
mod_h264_streaming_la_CFLAGS = $(AM_CFLAGS)
-DBUILDING_H264_STREAMING
mod_h264_streaming_la_LDFLAGS = -module -export-dynamic
```

```
-avoid-version -no-undefined
mod_h264_streaming_la_LIBADD = $(common_libadd)
```

```
cd lighttpd-1.4.26
```

```
./autogen.sh
```

```
./configure --enable-maintainer-mode --prefix=/home/lighttpd
```

```
make
```

```
sudo make install (此行指令需要 root 權限方可執行)
```

為配合 lighttpd.conf 的修改內容，需先建立原先不存在的資料夾及空白 log 檔案：

```
sudo mkdir -p /var/lighttpd/logs (建立資料夾)
```

```
sudo gedit /var/lighttpd/logs/lighttpd.error.log (建立檔案，須儲存後再關閉視窗)
```

```
sudo gedit /var/lighttpd/logs/lighttpd.breakage.log (建立檔案，須儲存後再關閉視窗)
```

開啟並修改 lighttpd.conf (詳細路徑：/lighttpd/lighttpd-1.4.26/tests/lighttpd.conf)：

```
gedit /tests/lighttpd.conf
```

在開啟的 lighttpd.conf 中全選並貼上以下敘述來取代原先的敘述(IP位址需修改)：

```
# lighttpd configuration file
#
# use it as a base for lighttpd 1.0.0 and above
#
# $Id: lighttpd.conf,v 1.7 2004/11/03 22:26:05 weigon Exp $

##### Options you really have to take care of
#####

## modules to load
# at least mod_access and mod_accesslog should be loaded
# all other module should only be loaded if really necessary
# - saves some time
# - saves memory
server.modules          = (
    ...
```

```

        "mod_h264_streaming",
        ...
    )
... (中間省略)
...
# mimetype mapping
mimetype.assign          = (
    ...
    ".css"               =>    "text/css",
    ...
    # default mime type
    ""                   =>    "application/octet-stream",
)
...
##### Options that are good to be but not necessary to be changed
#####
## 64 Mbyte ... nice limit
#server.max-request-size = 65000

## bind to port (default: 80)
server.port               = 80

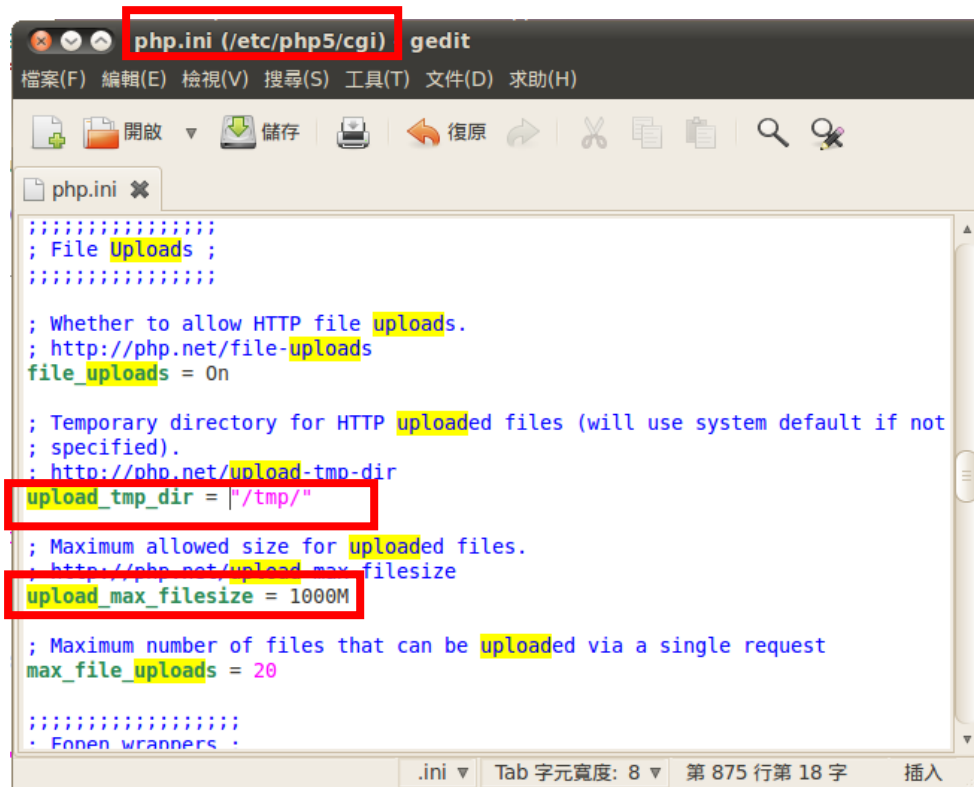
## bind to localhost (default: all interfaces)
server.bind               = "192.168.xxx.xxx" (此處請修改為自己主機的 IP)
... (中間省略)
#### proxy module
## read proxy.txt for more info
#proxy.server            = ( ".php" =>
#                          ( "localhost" =>
#                            (
#                              "host" => "192.168.0.101",
#                              "port" => 80
#                            )
#                          )
#                        )
#
#
... (以下省略)
...

```

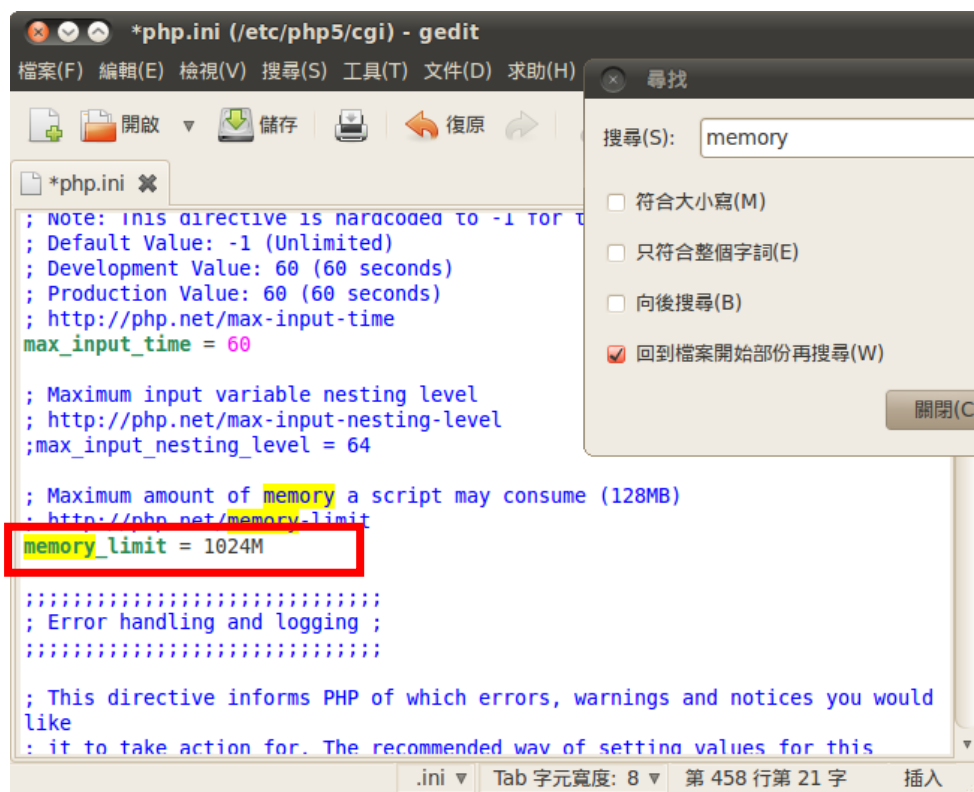

接著修改 php 設定檔(以下步驟須針對自己主機的記憶體大小作修改)

```
sudo gedit /etc/php5/cgi/php.ini
```

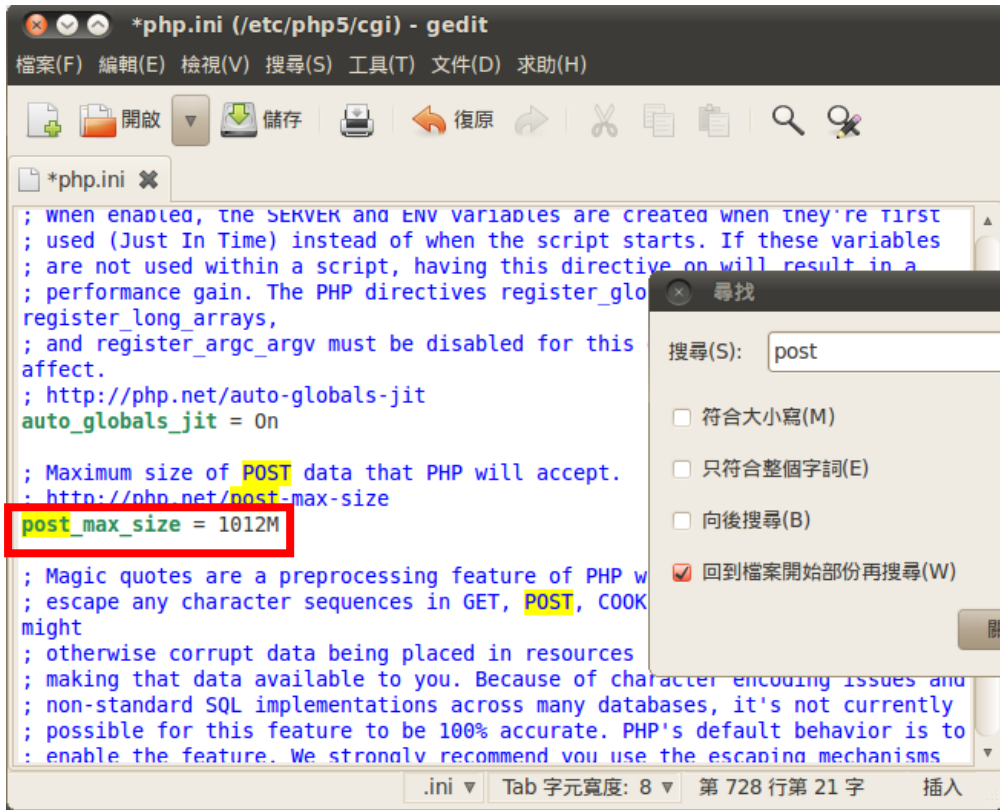
修改上傳暫存資料夾路徑及最大上傳檔案大小：



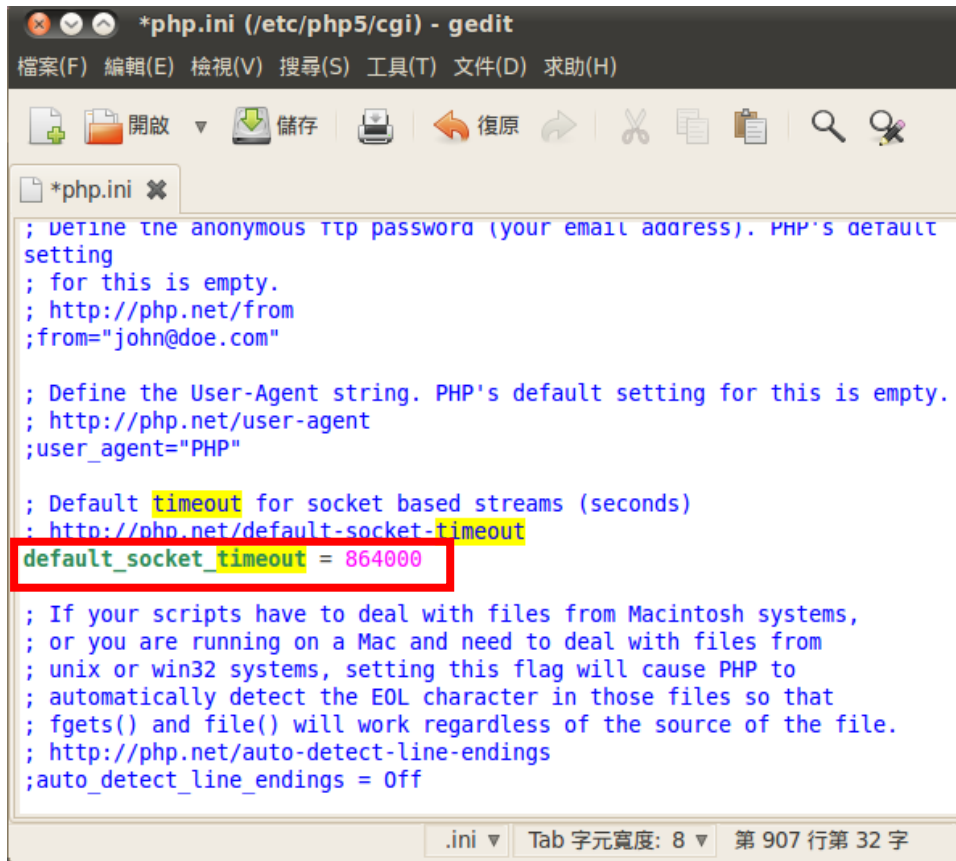
輸入自己主機的記憶體大小：



修改最大傳輸檔案大小：



修改 timeout 的時間(上傳超過這個時間就會失敗)：



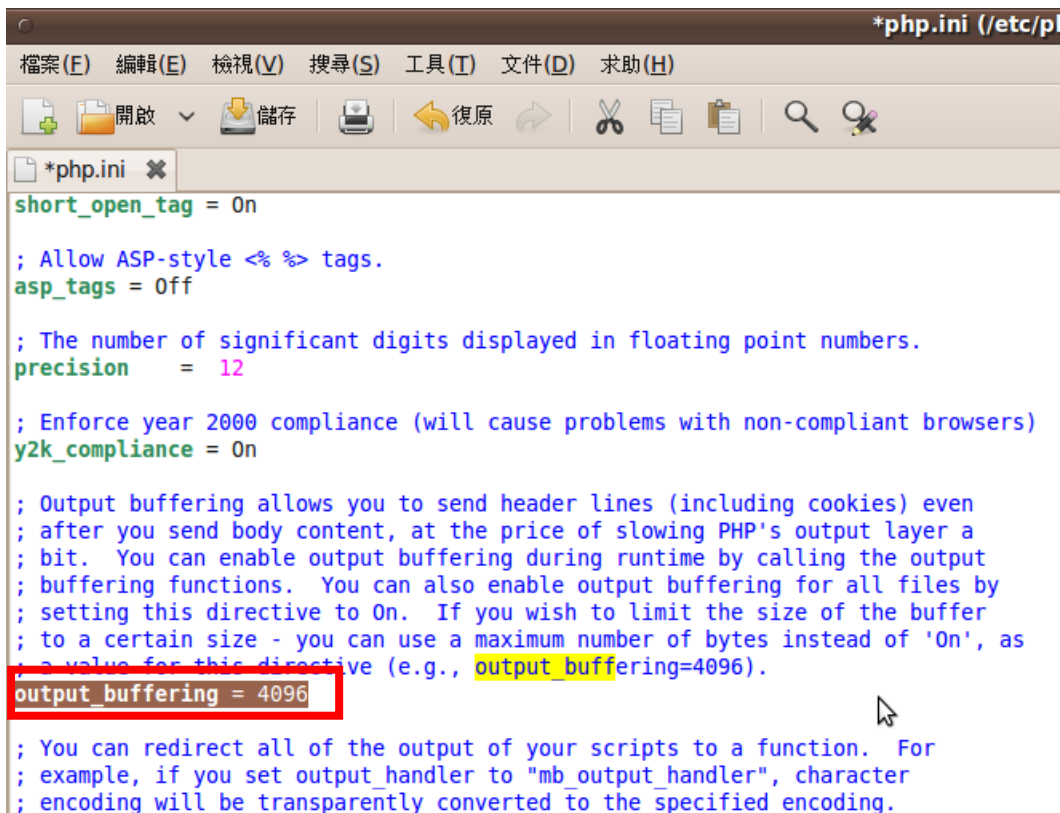
修改 maxlifetime 的時間(用來控制 session 連線時間)：

```
; Defines the probability that the 'garbage collection' process is started on every
; session initialization. The probability is calculated by using the following equation:
; gc_probability/gc_divisor. Where session.gc_probability is the numerator and
; session.gc_divisor is the denominator in the equation. Setting this value to 1
; when the session.gc divisor value is 100 will give you approximately a 1% chance
; the gc will run on any give request. Increasing this value to 1000 will give you
; a 0.1% chance the gc will run on any give request. For high volume production servers,
; this is a more efficient approach.
; Default Value: 100
; Development Value: 1000
; Production Value: 1000
; http://php.net/session.gc-divisor
session.gc_divisor = 1000

; After this number of seconds, stored data will be seen as 'garbage' and
; cleaned up by the garbage collection process.
; http://php.net/session.gc_maxlifetime
session.gc_maxlifetime = 864000

; NOTE: If you are using the subdirectory option for storing session files
; (see session.save_path above), then garbage collection does *not*
; happen automatically. You will need to do your own garbage
; collection through a shell script, cron entry, or some other method.
; For example, the following script would is the equivalent of
; setting session.gc_maxlifetime to 1440 (1440 seconds = 24 minutes):
; cd /path/to/sessions; find -cmin +24 | xargs rm
```

如發生錯誤請檢查 output_buffering 的設定：



The image shows a screenshot of a text editor window titled '*php.ini (/etc/p...'. The editor displays the configuration for the 'output_buffering' directive. The line 'output_buffering = 4096' is highlighted with a red box. Other visible configuration lines include 'short_open_tag = On', 'asp_tags = Off', 'precision = 12', and 'y2k_compliance = On'. The editor's menu bar includes options like '檔案(E)', '編輯(E)', '檢視(V)', '搜尋(S)', '工具(T)', '文件(D)', and '求助(H)'. The toolbar contains icons for opening, saving, printing, and undo/redo.

```
*php.ini (/etc/p...
檔案(E) 編輯(E) 檢視(V) 搜尋(S) 工具(T) 文件(D) 求助(H)
開啟 儲存 復原
*php.ini x
short_open_tag = On
; Allow ASP-style <% %> tags.
asp_tags = Off
; The number of significant digits displayed in floating point numbers.
precision = 12
; Enforce year 2000 compliance (will cause problems with non-compliant browsers)
y2k_compliance = On
; Output buffering allows you to send header lines (including cookies) even
; after you send body content, at the price of slowing PHP's output layer a
; bit. You can enable output buffering during runtime by calling the output
; buffering functions. You can also enable output buffering for all files by
; setting this directive to On. If you wish to limit the size of the buffer
; to a certain size - you can use a maximum number of bytes instead of 'On', as
; a value for this directive (e.g., output_buffering=4096).
output_buffering = 4096
; You can redirect all of the output of your scripts to a function. For
; example, if you set output_handler to "mb_output_handler", character
; encoding will be transparently converted to the specified encoding.
```

建立 pages 資料夾（此資料夾將用來存放網頁原始碼，名稱路徑可自定）：

```
sudo mkdir -p /var/lighttpd/servers/www.example.org/pages
```

由於剛剛在/var 底下建立的 /lighttpd/pages 資料夾是 root 權限，需要使用下列的指令來更改為 user 權限：

```
sudo chown -R 使用者名稱 /var/lighttpd //改變 log owner
```

接著把所有「網頁原始碼」和資料夾檔案複製一份到自己設立的資料夾中(參考路徑：/var/lighttpd/servers/www.example.org/pages/)內

(3)安裝 MySQL

安裝 MySQL 相關套件：

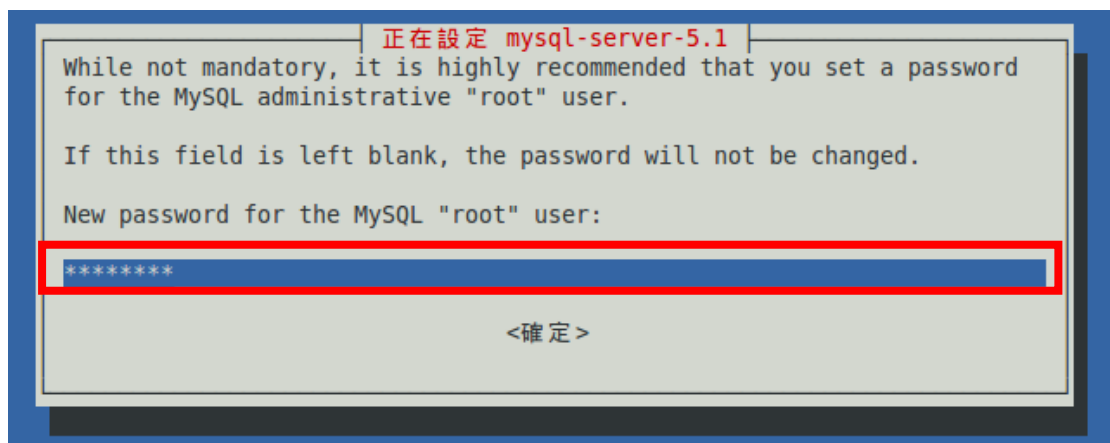
```
sudo apt-get install mysql-server mysql-client
```



```
lawrence@ubuntu:~$ sudo apt-get install mysql-server mysql-client
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
以下套件是被自動安裝進來的，且已不再會被用到了：
  linux-headers-2.6.32-21 memcached libmemcache0
  linux-headers-2.6.32-21-generic spawn-fcgi libterm-readline-perl-perl
使用 'apt-get autoremove' 來將其移除。
下列的額外套件將被安裝：
  libdbd-mysql-perl libdbi-perl libhtml-template-perl libnet-daemon-perl
  libplrpc-perl mysql-client-5.1 mysql-client-core-5.1 mysql-server-5.1
  mysql-server-core-5.1
建議套件：
  dbshell libipc-sharedcache-perl tinyca mailx
下列【新】套件將會被安裝：
  libdbd-mysql-perl libdbi-perl libhtml-template-perl libnet-daemon-perl
  libplrpc-perl mysql-client mysql-client-5.1 mysql-client-core-5.1
  mysql-server mysql-server-5.1 mysql-server-core-5.1
升級 0 個，新安裝 11 個，移除 0 個，有 0 個未被升級。
需要下載 21.3MB 的套件檔。
```

MySQL 密碼設定：

(為了讓我們在開發期間方便修改內容，暫時先把密碼設為 123456)



```
正在設定 mysql-server-5.1
While not mandatory, it is highly recommended that you set a password
for the MySQL administrative "root" user.

If this field is left blank, the password will not be changed.

New password for the MySQL "root" user:
*****
<確定>
```

安裝 phpMyAdmin 網頁型資料庫管理：

到 http://www.phpmyadmin.net/home_page/downloads.php 下載

phpMyAdmin-3.3.5.1-all-languages.tar.gz 套件，下載完後解壓縮，並把到解壓縮後的資料夾整個移到 pages 資料夾內(請將資料夾名稱由原先的 phpMyAdmin-3.3.5.1-all-languages.tar.gz，改為 phpMyAdmin，方便管理)

到 phpMyAdmin 資料夾中將 config.sample.inc.php 複製一份，並重新命名為 config.inc.php 設定檔：

```
cd /var/lighttpd/servers/www.example.org/pages/phpMyAdmin
cp config.sample.inc.php config.inc.php
```

修改 phpMyAdmin 資料夾中的 config.inc.php 設定檔：

```
/*
 * This is needed for cookie based authentication to encrypt password in
 * cookie
 */
$cfg['blowfish_secret'] = '123456'; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */

/*
 * Servers configuration
 */
$i = 0;

/*
 * First server
 */
$i++;
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'cookie'
/* Server parameters */
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
/* Select mysqli if your server has it */
$cfg['Servers'][$i]['extension'] = 'mysql';
$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

執行 MySQL：

```
sudo /etc/init.d/mysql start
```

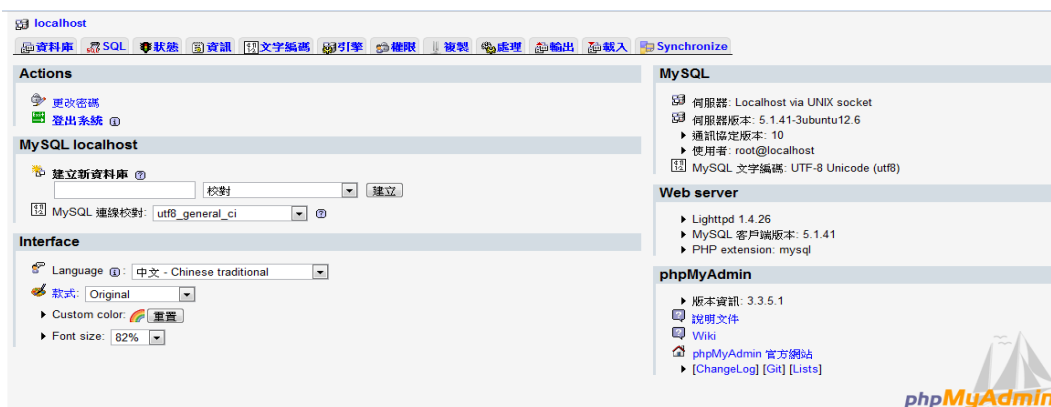
執行測試 lighttpd：

```
$ cd /home/lighttpd/sbin
$ sudo ./lighttpd -D -f
/lighttpd/lighttpd-1.4.26/tests/lighttpd.conf
```

開啟 phpmyadmin 頁面(在 lighttpd 執行中才可開啟)：

網址輸入 http://自己主機的 IP 位址/phpMyAdmin(即先前解壓縮修改的資料夾名稱)/

(可先將語言(Language)改為繁體中文，方便管理與對照說明)



建立資料庫

建立新 web 資料庫：

校對選擇 utf8_unicode_ci，設定完後點選『建立』



新增資料表 video (存放影片資料)

localhost > web > video

瀏覽 結構 SQL 搜索 新增 輸出 載入 管理 清空 刪除

欄位	型態	校對	屬性	Null	預設值	附加
<input type="checkbox"/> v_id	varchar(12)	utf8_unicode_ci		否	None	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_title	varchar(255)	utf8_unicode_ci		否	None	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_text	text	utf8_unicode_ci		否	None	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_tag	varchar(255)	utf8_unicode_ci		是	NULL	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_category	tinyint(2)		UNSIGNED	否	None	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_length	time			是	NULL	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_time	datetime			否	None	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> u_id	varchar(255)	utf8_unicode_ci		否	None	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_lock	tinyint(1)		UNSIGNED	否	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_convert	tinyint(1)		UNSIGNED	否	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> v_visit	int(10)		UNSIGNED	否	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

↑ 全選 / 全部取消 選擇的資料表:

列印檢視 分析資料表結構

新增 1 個欄位 於資料表尾端 於資料表開頭 在之後 v_id 執行

索引:

執行	鍵名	型態	唯一鍵 UNIQUE	Packed	欄位	組別	校對	Null	Comment
<input type="checkbox"/>	PRIMARY	BTREE	是	否	v_id	23	A		

v_id:影片 id，使用現在時間加上亂數所組成的 16 進位數字作為影片 id 代號
v_title:影片名稱，由使用者輸入
v_text:影片簡介，由使用者輸入
v_tag:影片標籤，由使用者輸入，可以是空值
v_category:影片分類，由使用者選擇類別
v_length:影片長度，轉檔時產生
v_time:上傳時間，上傳時寫入當天日期
u_id:上傳影片的人
v_lock:預設值為 0，當為 5 時影片將被封鎖，當同一會員有 3 部影片被封鎖時，此會員將無法登入
v_convert:當值為 1 時表示影片已轉檔完成
v_visit:統計影片觀看人次

```
CREATE TABLE `web`.`video` (  
  `v_id` VARCHAR( 12 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,  
  `v_title` VARCHAR( 255 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL ,  
  `v_text` TEXT CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,  
  `v_tag` VARCHAR( 255 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL  
  DEFAULT NULL ,  
  `v_category` TINYINT( 2 ) UNSIGNED NOT NULL ,  
  `v_length` TIME NULL DEFAULT NULL ,  
  `v_time` DATE NOT NULL ,  
  `u_id` VARCHAR( 255 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,  
  `v_lock` TINYINT( 1 ) UNSIGNED NOT NULL DEFAULT '0',  
  `v_convert` TINYINT( 1 ) UNSIGNED NOT NULL DEFAULT '0',  
  `v_visit` TINYINT( 10 ) UNSIGNED NOT NULL DEFAULT '0',  
  PRIMARY KEY ( `v_id` )  
 ) ENGINE = MYISAM ;
```

新增資料表 login_info (存放會員資料)

localhost ▶ web ▶ login_info

瀏覽 結構 SQL 搜索 新增 輸出 載入 管理 清空 刪除

欄位	型態	校對	屬性	Null	預設值	附加
<input type="checkbox"/> ID	mediumint(6)		UNSIGNED	否	None	AUTO_INCREMENT
<input type="checkbox"/> u_id	varchar(255)	utf8_unicode_ci		否	None	
<input type="checkbox"/> u_pwd	varchar(255)	utf8_unicode_ci		否	None	
<input type="checkbox"/> u_name	varchar(255)	utf8_unicode_ci		否	None	
<input type="checkbox"/> u_mail	varchar(255)	utf8_unicode_ci		否	None	
<input type="checkbox"/> u_time	datetime			否	None	
<input type="checkbox"/> u_chk	tinyint(1)		UNSIGNED	否	None	
<input type="checkbox"/> u_logintime	datetime			是	NULL	

全選 / 全部取消 選擇的資料表:

列印檢視 分析資料表結構

新增 1 個欄位 於資料表尾端 於資料表開頭 在之後 ID 執行

索引:

執行	鍵名	型態	唯一鍵 UNIQUE	Packed	欄位	組別	校對	Null	Comment
<input checked="" type="checkbox"/>	PRIMARY	BTREE	是	否	ID	6	A		

ID:流水編號

u_id:帳號

u_pwd:密碼，存入時有加密

u_name:註冊人的真實姓名

u_mail:註冊時的信箱，發送信件到此信箱











u_time:註冊的日期與時間






u_chk:是否註冊已經認證

u_logintime:最後一次登入的時間

```
CREATE TABLE `web`.`login_info` (
  `ID` MEDIUMINT( 6 ) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `u_id` VARCHAR( 255 ) NOT NULL ,
  `u_pwd` VARCHAR( 255 ) NOT NULL ,
  `u_name` VARCHAR( 255 ) NOT NULL ,
  `u_mail` VARCHAR( 255 ) NOT NULL ,
  `u_time` DATETIME NOT NULL ,
  `u_chk` TINYINT( 1 ) UNSIGNED NOT NULL ,
  `u_logintime` DATETIME NULL DEFAULT NULL
) ENGINE = MYISAM ;
```


新增資料表 lockd (存放暫時封鎖的帳號)



欄位	型態	校對	屬性	Null	預設值	附加	執行
<input type="checkbox"/> id	varchar(255)	utf8_unicode_ci		否	None		    
<input type="checkbox"/> locktime	varchar(30)	utf8_unicode_ci		否	None		    

↑ 全選 / 全部取消 選擇的資料表:     

列印檢視 分析資料表結構

新增 1 個欄位 於資料表尾端 於資料表開頭 在之後 id 執行

索引:

執行	鍵名	型態	唯一鍵 UNIQUE	Packed	欄位	組別	校對	Null	Comment
 	PRIMARY	BTREE	是	否	id	0	A		

id:密碼錯誤多次，被封鎖的帳號

locktime:封鎖時間，預設 5 分鐘

```
CREATE TABLE `web`.`lockd` (
  `id` VARCHAR( 255 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,
  `locktime` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
  NULL ,
  PRIMARY KEY ( `id` )
) ENGINE = MYISAM ;
```

新增資料表 comment (儲存留言內容)

localhost ▶ web ▶ comment

瀏覽 結構 SQL 搜索 新增 輸出 載入 管理 清空 刪除

欄位	型態	校對	屬性	Null	預設值	附加	執行
<input type="checkbox"/> v_id	varchar(12)	utf8_unicode_ci		否	None		    
<input type="checkbox"/> u_postid	varchar(255)	utf8_unicode_ci		否	None		    
<input type="checkbox"/> u_owner	varchar(255)	utf8_unicode_ci		否	None		    
<input type="checkbox"/> c_time	varchar(30)	utf8_unicode_ci		否	None		    
<input type="checkbox"/> c_text	text	utf8_unicode_ci		否	None		    

↑ 全選 / 全部取消 選擇的資料表:     

列印檢視 分析資料表結構

新增 1 個欄位 於資料表尾端 於資料表開頭 在之後 v_id 執行

索引:

執行	鍵名	型態	唯一鍵 UNIQUE	Packed	欄位	組別	校對	Null	Comment
 	PRIMARY	BTREE	是	否	c_time	24	A		
					v_id	24	A		

請注意留言板的 primary key 為 **c_time + v_id**

v_id:頁面影片編號，根據此欄位區別不同的留言板

u_postid: 留言發言者，自己可以刪除自己的留言

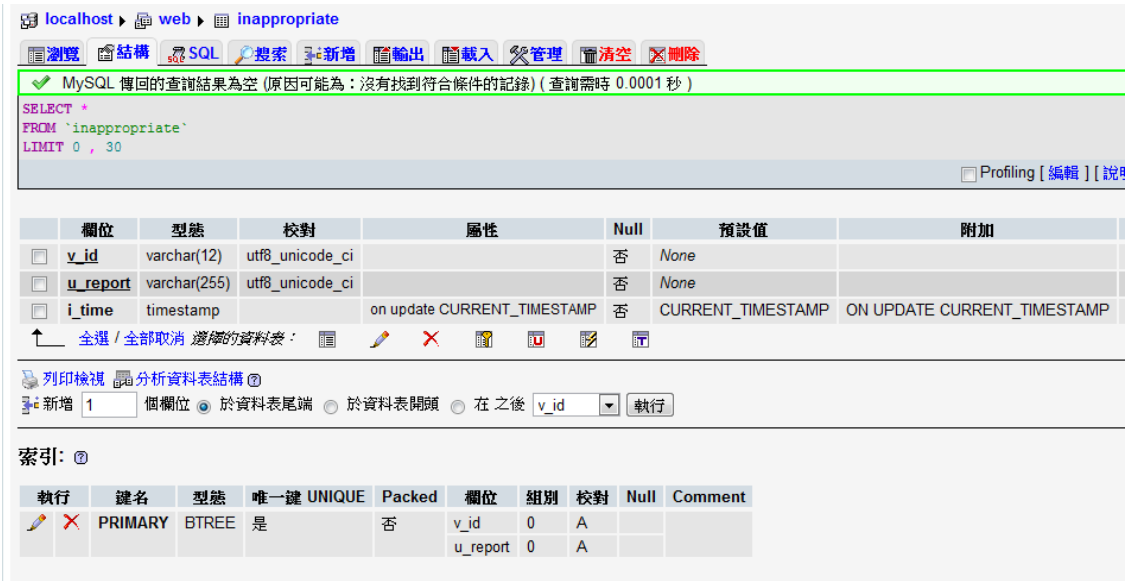
u_owner: 影片所有人，影片 owner 可以刪除自己頁面的所有留言

c_time: 發言的時間，用此時間來判斷哪一則留言

c_text: 留言的內容

```
CREATE TABLE `web`.`comment` (  
  `v_id` VARCHAR( 12 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,  
  `u_postid` VARCHAR( 255 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL ,  
  `u_owner` VARCHAR( 255 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL ,  
  `c_time` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT  
  NULL ,  
  `c_text` TEXT CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,  
  PRIMARY KEY ( `v_id` , `c_time` )  
 ) ENGINE = MYISAM ;
```

新增資料表 inappropriate (記錄被檢舉的影片資訊)



MySQL 傳回的查詢結果為空 (原因可能為：沒有找到符合條件的記錄) (查詢需時 0.0001 秒)

```
SELECT *  
FROM `inappropriate`  
LIMIT 0 , 30
```

欄位	型態	校對	屬性	Null	預設值	附加
<input type="checkbox"/> v_id	varchar(12)	utf8_unicode_ci		否	None	
<input type="checkbox"/> u_report	varchar(255)	utf8_unicode_ci		否	None	
<input type="checkbox"/> i_time	timestamp		on update CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP

索引:

執行	鍵名	型態	唯一鍵 UNIQUE	Packed	欄位	組別	校對	Null	Comment
<input type="checkbox"/>	PRIMARY	BTREE	是	否	v_id	0	A		
<input type="checkbox"/>					u_report	0	A		

請注意檢舉影片的 primary key 為 v_id+ u_report

v_id: 被檢舉影片 id，並記錄到 video 資料表 v_lock 中，累計超過 5 次不同會員檢舉，影片即被封鎖

u_report: 檢舉者的帳號，會員才可以進行檢舉，每位會員只能對同一部影片檢舉一次

i_time: 檢舉的時間註記

```
CREATE TABLE `web`.`inappropriate` (
  `v_id` VARCHAR( 12 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,
  `u_report` VARCHAR( 255 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
  NULL ,
  `i_time` TIMESTAMP ON UPDATE CURRENT_TIMESTAMP NOT NULL DEFAULT
  CURRENT_TIMESTAMP ,
  PRIMARY KEY ( `v_id` , `u_report` )
) ENGINE = MYISAM ;
```

新增資料表 convert (暫時記錄轉檔已完成的影片)

The screenshot shows the phpMyAdmin interface for the 'convert' table. The table structure is as follows:

欄位	型態	校對	屬性	Null	預設值	附加	執行
<input type="checkbox"/> v_id	varchar(12)	utf8_unicode_ci		否	None		

Below the table structure, there are options for adding a new column: '新增 1 個欄位'. The '於資料表開頭' radio button is selected. The '執行' button is visible.

The '索引' (Indexes) section shows a primary key:

執行	鍵名	型態	唯一鍵 UNIQUE	Packed	欄位	組別	校對	Null	Comment
	PRIMARY	BTREE	是	否	v_id	1	A		

請注意本資料表必須預設一筆值為 0 的資料新增在表內

v_id: 暫存轉檔完成影片的 id, 待資料寫入資料表 video 後即會自動刪除本表資料

```
CREATE TABLE `web`.`convert` (
  `v_id` VARCHAR( 12 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,
  PRIMARY KEY ( `v_id` )
) ENGINE = MYISAM ;
```

(4)安裝 FFmpeg (FFmpeg-mt 版：Multi-Thread)

1.安裝前請先 install 這些套件

```
sudo apt-get install libtool
sudo apt-get install git-core
sudo apt-get install subversion
sudo apt-get install g++
sudo apt-get install yasm
```

2.下載所需編譯檔

FFmpeg-mt

```
git clone
http://git.gitorious.org/~astrange/ffmpeg/ffmpeg-mt.git
或是
git clone git://gitorious.org/~astrange/ffmpeg/ffmpeg-mt.git
```

FAAC (faac-1.28.tar.gz)

<http://www.audiocoding.com/downloads.html>

x264

```
sudo git clone git://git.videolan.org/x264.git
```

libswscale

```
git clone git://git.ffmpeg.org/libswscale/
```

複製全部資料到 ffmpeg-mt 中

```
cp -a libswscale/ /home/使用者名稱/ffmpeg-mt/
```

3.編譯各檔案 (ffmpeg 需最後編譯)

FAAC

```
sudo tar -zxvf faac-1.28.tar.gz
```

編譯前請先載入外掛以免發生錯誤

```
cd faac-1.28
```

vim 一個檔案為 faac-1.28.diff (沒有該檔案)

寫入 http://abechin.sakura.ne.jp/sblo_files/k-tai-douga/ffmpeg/faac-1.28.diff 的資料

編寫存檔完後輸入

```
sudo patch -p1 < faac-1.28.diff    載入完成
sudo ./bootstrap
sudo ./configure --prefix=/usr/local
sudo make
sudo make install
```

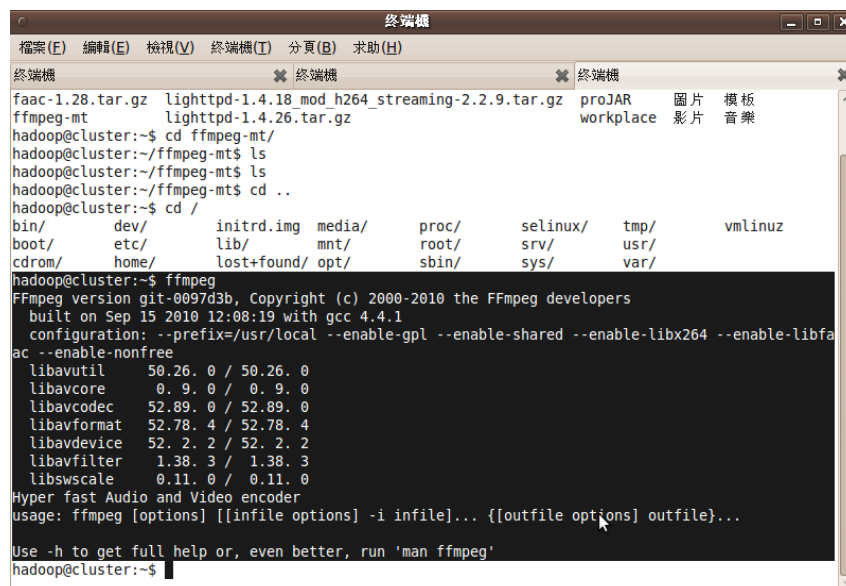
x264

```
cd x264
sudo ./configure --prefix=/usr/local --enable-shared
sudo make
sudo make install
```

FFmpeg-mt

```
cd ffmpeg-mt/
sudo ./configure --prefix=/usr/local --enable-gpl --enable-shared
--enable-libx264 --enable-libfaac --enable-nonfree
sudo make
sudo make install
ffmpeg
```

出現類似下面的圖案就表示成功了



```
终端機
檔案(E) 編輯(E) 檢視(V) 终端機(T) 分页(B) 求助(H)
终端機
faac-1.28.tar.gz  lighttpd-1.4.18_mod_h264_streaming-2.2.9.tar.gz  proJAR  图片  模板
ffmpeg-mt        lighttpd-1.4.26.tar.gz  workplace  影片  音乐
hadoop@cluster:~$ cd ffmpeg-mt/
hadoop@cluster:~/ffmpeg-mt$ ls
hadoop@cluster:~/ffmpeg-mt$ ls
hadoop@cluster:~/ffmpeg-mt$ cd ..
hadoop@cluster:~$ cd /
bin/      dev/      initrd.img  media/     proc/      selinux/   tmp/       vmlinuz
boot/     etc/      lib/        mnt/       root/      srv/       usr/
cdrom/    home/     lost+found/ opt/        sbin/      sys/       var/
hadoop@cluster:~$ ffmpeg
FFmpeg version git-0097d3b, Copyright (c) 2000-2010 the FFmpeg developers
  built on Sep 15 2010 12:08:19 with gcc 4.4.1
  configuration: --prefix=/usr/local --enable-gpl --enable-shared --enable-libx264 --enable-libfaac --enable-nonfree
  libavutil      50.26. 0 / 50.26. 0
  libavcore      0. 9. 0 / 0. 9. 0
  libavcodec     52.89. 0 / 52.89. 0
  libavformat    52.78. 4 / 52.78. 4
  libavdevice    52. 2. 2 / 52. 2. 2
  libavfilter    1.38. 3 / 1.38. 3
  libswscale     0.11. 0 / 0.11. 0
Hyper fast Audio and Video encoder
usage: ffmpeg [options] [[infile options] -i infile]... [[outfile options] outfile]...
Use -h to get full help or, even better, run 'man ffmpeg'
hadoop@cluster:~$
```

(5)安裝 Nutch

Nutch1.1 安裝

```
cd /opt
sudo wget
http://ftp.mirror.tw/pub/apache//nutch/apache-nutch-1.1-bin.tar.gz
sudo tar -zxvf apache-nutch-1.1-bin.tar.gz
sudo mv apache-nutch-1.1-bin nutch
```

```
sudo chown -R 使用者名稱:使用者名稱 /opt/nutch/  
sudo mkdir /var/nutch-hadoop  
sudo chown -R 使用者名稱:使用者名稱 /var/nutch-hadoop  
cd nutch/
```

1. 改寫 hadoop-env.sh 檔

```
gedit conf/hadoop-env.sh
```

可以 cp 之前 hadoop 的 hadoop-env.sh (如果刪掉的話就依照之前的 hadoop 安裝, 以下是我自己的設定檔)

```
export JAVA_HOME=/lib/jdk1.6.0_21  
export HADOOP_HOME=/opt/nutch  
export HADOOP_CONF_DIR=/opt/nutch/conf  
export HADOOP_SLAVES=/opt/nutch/conf/slaves  
export HADOOP_LOG_DIR=/var/nutch-hadoop/logs  
export HADOOP_PID_DIR=/var/nutch-hadoop/pids  
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true  
export HADOOP_NAMENODE_OPTS="-Dcom.sun.management.jmxremote  
$HADOOP_NAMENODE_OPTS"  
export  
HADOOP_SECONDARYNAMENODE_OPTS="-Dcom.sun.management.jmxremote  
$HADOOP_SECONDARYNAMENODE_OPTS"  
export HADOOP_DATANODE_OPTS="-Dcom.sun.management.jmxremote  
$HADOOP_DATANODE_OPTS"  
export HADOOP_BALANCER_OPTS="-Dcom.sun.management.jmxremote  
$HADOOP_BALANCER_OPTS"  
export HADOOP_JOBTRACKER_OPTS="-Dcom.sun.management.jmxremote  
$HADOOP_JOBTRACKER_OPTS"
```

2. 改寫 core-site.xml, mapred-site.xml, hdfs-site.xml, slaves, masters, nutch-site.xml, crawl-urlfilter.txt

```
gedit conf/core-site.xml
```

```
<configuration>  
  <property>  
    <name>fs.default.name</name>  
    <value>hdfs://主機名:9000</value>  
  </property>  
  <property>  
    <name>hadoop.tmp.dir</name>
```

```
<value>/var/nutch-hadoop/hadoop- $\{user.name\}$ </value>
</property>
</configuration>
```

gedit conf/mapred-site.xml

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>主機名:9001</value>
  </property>
</configuration>
```

gedit conf/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

gedit conf/slaves

刪掉預設內容,改成各 node 主機名

gedit conf/masters

刪除所有內容

gedit conf/nutch-site.xml

```
<configuration>
  <property>
    <name>http.agent.name</name>
    <value>nutch</value>
    <description>HTTP 'User-Agent' request header. </description>
  </property>
  <property>
    <name>http.agent.description</name>
    <value>MyTest</value>
    <description>Further description</description>
  </property>
```

```

<property>
  <name>http.agent.url</name>
  <value>cluster</value>
  <description>A URL to advertise in the User-Agent header.
</description>
</property>
<property>
  <name>http.agent.email</name>
  <value>test@test.org.tw</value>
  <description>An email address
</description>
</property>
<property>
  <name>plugin.folders</name>
  <value>/opt/nutch/plugins</value>
  <description>Directories where nutch plugins are located.
</description>
</property>
<property>
  <name>plugin.includes</name>

  <value>protocol-(http|httpclient)|urlfilter-regex|parse-(text
|html|js|ext|msexcel|mspowerpoint|msword|oo|pdf|rss|swf|zip)|
index-(more|basic|anchor)|query-(more|basic|site|url)|respons
e-(json|xml)|summary-basic|scoring-opic|urlnormalizer-(pass|r
egex|basic)</value>
  <description> Regular expression naming plugin directory
names</description>
</property>
<property>
  <name>parse.plugin.file</name>
  <value>parse-plugins.xml</value>
  <description>The name of the file that defines the associations
between
  content-types and parsers.</description>
</property>
<property>
  <name>db.max.outlinks.per.page</name>

```



```

    <value>-1</value>
    <description> </description>
</property>
<property>
    <name>http.content.limit</name>
    <value>-1</value>
</property>
<property>
    <name>indexer.mergeFactor</name>
    <value>500</value>
    <description>The factor that determines the frequency of Lucene
segment
merges. This must not be less than 2, higher values increase
indexing
speed but lead to increased RAM usage, and increase the number
of
open file handles (which may lead to "Too many open files"
errors).
NOTE: the "segments" here have nothing to do with Nutch segments,
they
are a low-level data unit used by Lucene.
</description>
</property>

<property>
    <name>indexer.minMergeDocs</name>
    <value>500</value>
    <description>This number determines the minimum number of
Lucene
Documents buffered in memory between Lucene segment merges.
Larger
values increase indexing speed and increase RAM usage.
</description>
</property>

<!--<property>
    <name>urlfilter.order</name>

```

```

<value>org.apache.nutch.urlfilter.regex.RegexURLFilter</value
>
  <description>The order by which url filters are applied.
  If empty, all available url filters (as dictated by properties
  plugin-includes and plugin-excludes above) are loaded and
  applied in system
  defined order. If not empty, only named filters are loaded and
  applied
  in given order. For example, if this property has value:
  org.apache.nutch.urlfilter.regex.RegexURLFilter
  org.apache.nutch.urlfilter.prefix.PrefixURLFilter
  then RegexURLFilter is applied first, and PrefixURLFilter
  second.
  Since all filters are AND'ed, filter ordering does not have
  impact
  on end result, but it may have performance implication,
  depending
  on relative expensiveness of filters.
  </description>
</property>-->
</configuration>

```

gedit conf/crawl-urlfilter.txt

修改內容

```

# skip ftp:, & mailto: urls
-^(ftp|mailto):
# skip image and other suffixes we can't yet parse
-\.(gif|GIF|jpg|JPG|png|PNG|ico|ICO|css|sit|eps|wmf|mpg|xls|g
z|rpm|tgz|mov|MOV|exe|jpeg|JPEG|bmp|BMP)$
# skip URLs containing certain characters as probable queries,
etc.
-[*!@]
# accept hosts in MY.DOMAIN.NAME
+^http://([a-z0-9]*\.)*.*/*
# accept anything else
+.*

```

將相關文件複製到其他主機，形成叢集(複製完畢後，也要修改/opt/nutch/和/var/nutch-hadoop 權限，參考文件前面語法)

```
scp -r /opt/nutch 主機名:/opt/nutch
scp -r /var/nutch-hadoop 主機名:/var/nutch-hadoop
```

3.安裝 Tomcat

```
cd /opt/
sudo wget
http://ftp.nsysu.edu.tw/Apache//tomcat/tomcat-7/v7.0.2-beta/bin/apache-tomcat-7.0.2.tar.gz
sudo tar -xzf apache-tomcat-7.0.2.tar.gz
sudo mv apache-tomcat-7.0.2 tomcat
```

4.修改 /opt/tomcat/conf/server.xml 以修正中文亂碼問題

```
sudo gedit /opt/tomcat/conf/server.xml
    <Connector port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" URIEncoding="UTF-8"
        useBodyEncodingForURI="true" />
```

修改資料夾權限

```
sudo chown -R 使用者名稱:使用者名稱 /opt/tomcat
```

啟動 tomcat

```
/opt/tomcat/bin/startup.sh
```

5.啟動 hadoop

```
cd /opt/nutch/
sudo mv nutch-1.1.war /opt/tomcat/webapps/
```

然後測試該網址 <http://自己 IP:8080/nutch-1.1/>

```
bin/hadoop namenode -format
bin/start-all.sh
mkdir urls
gedit urls/urls.txt
```

打上 data.php 的網址(記得啟動 lighttpd)
(記得將 data.php 的內容修改成自己的 IP)

```
bin/hadoop dfs -put urls urls
```

執行網頁爬取

```
bin/nutch crawl urls -dir search -threads 2 -depth 2 -topN 1000
```

如出現 *Error: JAVA_HOME is not set*. 修改下面設定：修改環境變數設定檔

```
sudo gedit /etc/bash.bashrc
```

```
#set java environment
JAVA_HOME=/lib/jdk1.6.0_21
...(中間省略)
export JAVA_HOME
```

```
bin/hadoop dfs -get search /tmp/search
```

```
sudo gedit
```

```
/opt/tomcat/webapps/nutch-1.1/WEB-INF/classes/nutch-site.xml
```

```
<configuration>
  <property>
    <name>searcher.dir</name>
    <value>/tmp/search</value>
  </property>
</configuration>
```

重新啟動 Tomcat 後即可以開始搜尋

P.S.1 關機時記得關閉 tomcat 與 hadoop

P.S.2 每做一次 search 資料夾須先執行下面兩個指令，將前一次執行的檔案刪除

```
bin/hadoop fs -rmr search
bin/hadoop fs -rmr urls
rm -rf /tmp/search/*
```

(六)安裝 FUSE

```
# sudo vim /etc/apt/sources.list
```

加入

```
deb http://us.archive.ubuntu.com/ubuntu/ jaunty multiverse
```

```
deb http://us.archive.ubuntu.com/ubuntu/ jaunty-updates
```

```
# sudo apt-get update
```

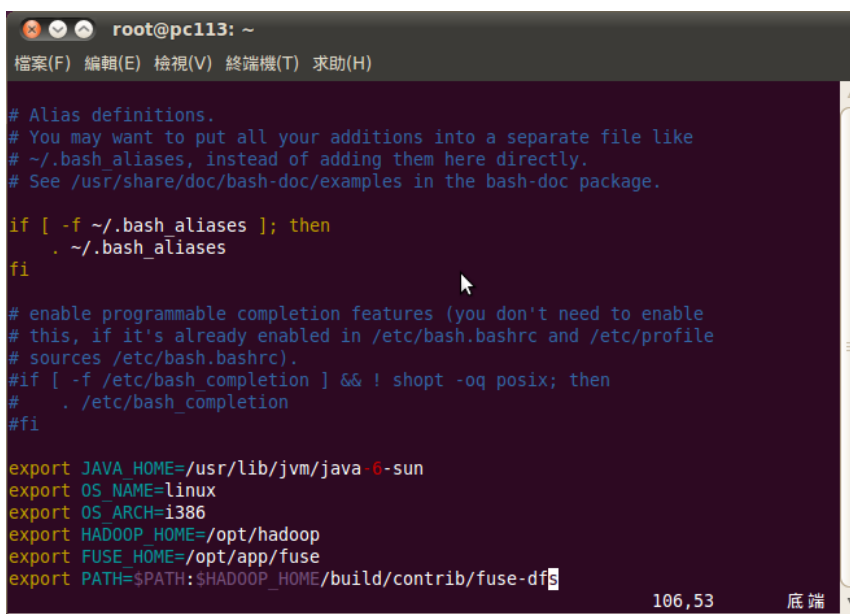
```
# sudo apt-get install subversion ant automake sun-java5-jdk
```

```
sun-java6-jdk
```

```
# sudo vim ~/.bashrc
```

加入 //請依自己電腦套件的路徑跟位置作修改

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun
export OS_NAME=linux
export OS_ARCH=i386
export HADOOP_HOME=/opt/hadoop
export FUSE_HOME=/opt/app/fuse
export PATH=$PATH:$HADOOP_HOME/build/contrib/fuse-dfs
```



A terminal window titled 'root@pc113: ~' showing the contents of a .bashrc file. The file contains alias definitions, completion features, and the export statements for JAVA_HOME, OS_NAME, OS_ARCH, HADOOP_HOME, FUSE_HOME, and PATH. The cursor is positioned at the end of the last export statement.

```
root@pc113: ~
檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi

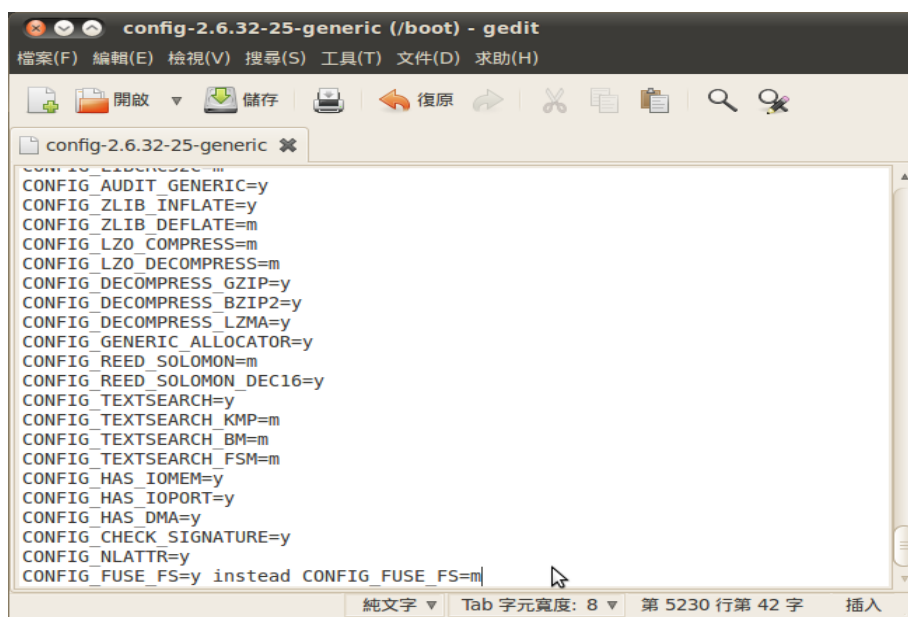
export JAVA_HOME=/usr/lib/jvm/java-6-sun
export OS_NAME=linux
export OS_ARCH=i386
export HADOOP_HOME=/opt/hadoop
export FUSE_HOME=/opt/app/fuse
export PATH=$PATH:$HADOOP_HOME/build/contrib/fuse-dfs
```

儲存離開

```
# source ~/.bashrc
# sudo vim /boot/config-2.6.32-24-generic
```

加入

```
CONFIG_FUSE_FS=y instead CONFIG_FUSE_FS=m
```



A screenshot of the gedit text editor showing the configuration file /boot/config-2.6.32-25-generic. The file contains various kernel configuration options, including CONFIG_AUDIT_GENERIC=y, CONFIG_ZLIB_INFLATE=y, CONFIG_ZLIB_DEFLATE=m, CONFIG_LZO_COMPRESS=m, CONFIG_LZO_DECOMPRESS=m, CONFIG_DECOMPRESS_GZIP=y, CONFIG_DECOMPRESS_BZIP2=y, CONFIG_DECOMPRESS_LZMA=y, CONFIG_GENERIC_ALLOCATOR=y, CONFIG_REED_SOLOMON=m, CONFIG_REED_SOLOMON_DEC16=y, CONFIG_TEXTSEARCH=y, CONFIG_TEXTSEARCH_KMP=m, CONFIG_TEXTSEARCH_BM=m, CONFIG_TEXTSEARCH_FSM=m, CONFIG_HAS_IOMEM=y, CONFIG_HAS_IOPORT=y, CONFIG_HAS_DMA=y, CONFIG_CHECK_SIGNATURE=y, CONFIG_NLATR=y, and CONFIG_FUSE_FS=y instead CONFIG_FUSE_FS=m. The cursor is positioned at the end of the last line.

```
config-2.6.32-25-generic (/boot) - gedit
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 工具(T) 文件(D) 求助(H)

CONFIG_AUDIT_GENERIC=y
CONFIG_ZLIB_INFLATE=y
CONFIG_ZLIB_DEFLATE=m
CONFIG_LZO_COMPRESS=m
CONFIG_LZO_DECOMPRESS=m
CONFIG_DECOMPRESS_GZIP=y
CONFIG_DECOMPRESS_BZIP2=y
CONFIG_DECOMPRESS_LZMA=y
CONFIG_GENERIC_ALLOCATOR=y
CONFIG_REED_SOLOMON=m
CONFIG_REED_SOLOMON_DEC16=y
CONFIG_TEXTSEARCH=y
CONFIG_TEXTSEARCH_KMP=m
CONFIG_TEXTSEARCH_BM=m
CONFIG_TEXTSEARCH_FSM=m
CONFIG_HAS_IOMEM=y
CONFIG_HAS_IOPORT=y
CONFIG_HAS_DMA=y
CONFIG_CHECK_SIGNATURE=y
CONFIG_NLATR=y
CONFIG_FUSE_FS=y instead CONFIG_FUSE_FS=m
```

安裝 hadoop-0.20.2 確定啟動成功，再關閉

```
# cd $HADOOP_HOME
# ant clean
# ant compile-c++-libhdfs -Dlibhdfs=1
```

安裝 forrest-0.8

```
# wget
http://apache.ntu.edu.tw//forrest/apache-forrest-0.8.tar.gz
# tar xzf apache-forrest-0.8.tar.gz
# chmod +x apache-forrest-0.8/bin/forrest
# cd /opt/hadoop
```

修改 build.xml 否則無法 ant

```
# sudo gedit build.xml
```

第 1059 行

改成<target name="package" depends="compile, jar, examples, tools-jar, jar-test, ant-tasks, package-librecordio"

第 1110 行

刪除

```
<copy todir="${dist.dir}/docs">
    <fileset dir="${build.docs}"/>
</copy>
```

儲存離開

```
# ant package -Djava5.home=/usr/lib/jvm/java-1.5.0-sun
-Dforrest.home=$(pwd)/apache-forrest-0.8
```

下載 fuse-2.8.5.tar.gz

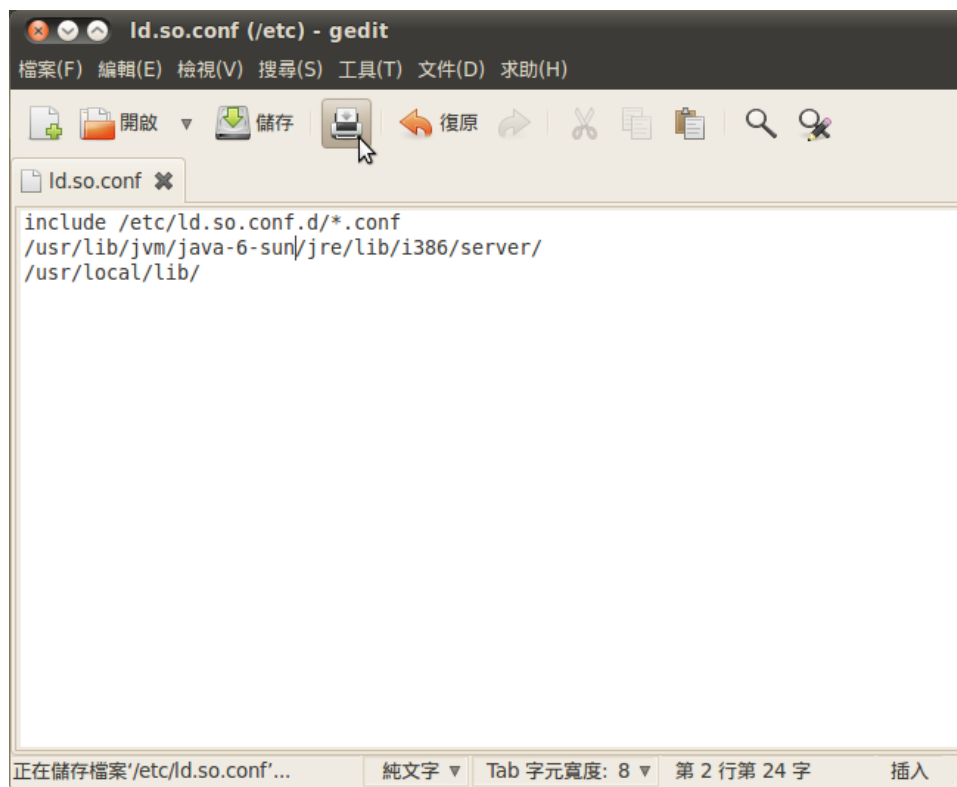
<http://sourceforge.net/projects/fuse/files/fuse-2.X/2.8.5/fuse-2.8.5.tar.gz/download>

```
# tar xzf fuse-2.8.5.tar.gz
# cd fuse-2.8.5
# mkdir -p /opt/app/fuse; ./configure --prefix=/opt/app/fuse
# make -j2
```

```
# make install
# modprobe fuse

# cd $HADOOP_HOME
# ant compile -Dcompile.cplusplus=true -Dlibhdfs=true (在 hadoop 目錄下)
# ant package
# cd build
# mkdir libhdfs
# cp build/c++/Linux-i386-32/lib/* $HADOOP_HOME/build/libhdfs
# ant compile-contrib -Dlibhdfs=1 -Dfusedfs=1

# gedit /etc/ld.so.conf
加入
/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/server/
/usr/local/lib/
```



```
# ldconfig
```

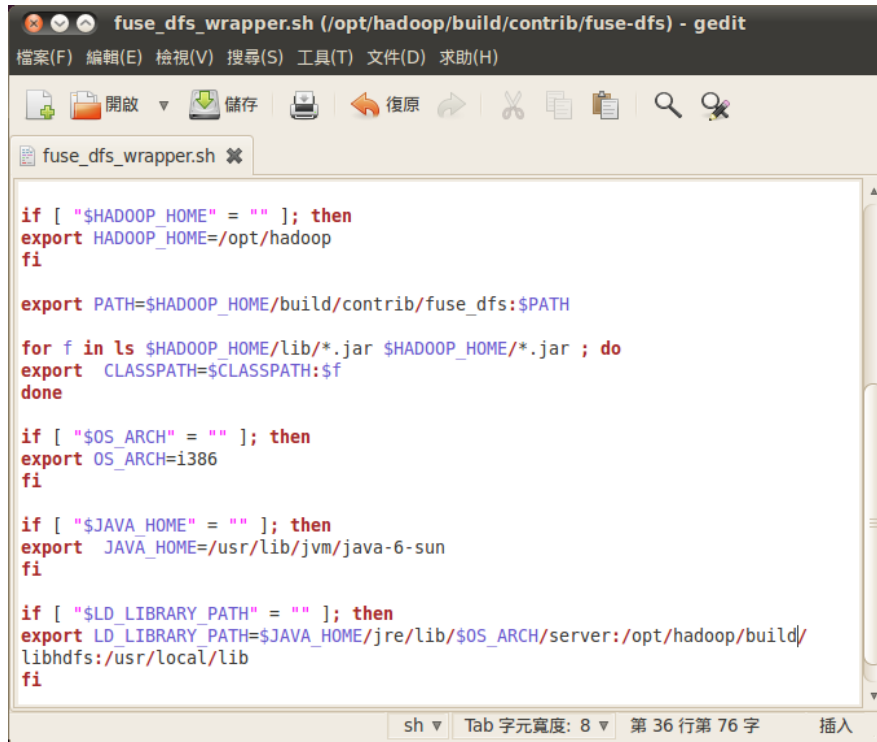
啟動 hadoop

```
# cd $HADOOP_HOME
```

```
# cd build/contrib/fuse-dfs
# chmod +x fuse_dfs_wrapper.sh
# mkdir /var/hdfs
```

修改 fuse_dfs_wrapper.sh 文件

```
# gedit fuse_dfs_wrapper.sh
```



```
fuse_dfs_wrapper.sh (/opt/hadoop/build/contrib/fuse-dfs) - gedit
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 工具(T) 文件(D) 求助(H)

if [ "$HADOOP_HOME" = "" ]; then
export HADOOP_HOME=/opt/hadoop
fi

export PATH=$HADOOP_HOME/build/contrib/fuse_dfs:$PATH

for f in ls $HADOOP_HOME/lib/*.jar $HADOOP_HOME/*.jar ; do
export CLASSPATH=$CLASSPATH:$f
done

if [ "$OS_ARCH" = "" ]; then
export OS_ARCH=i386
fi

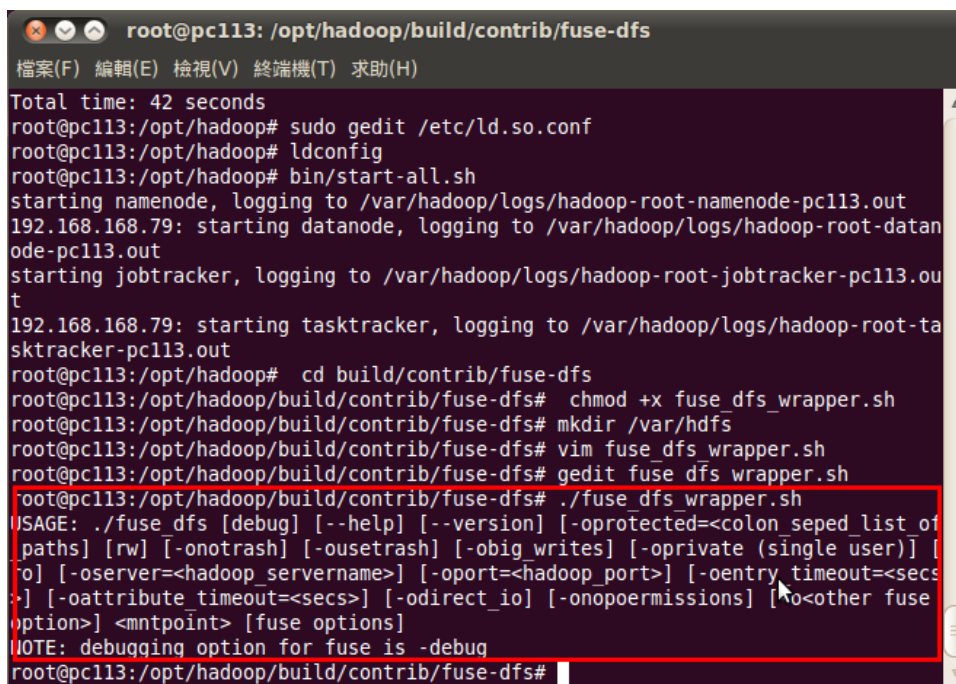
if [ "$JAVA_HOME" = "" ]; then
export JAVA_HOME=/usr/lib/jvm/java-6-sun
fi

if [ "$LD_LIBRARY_PATH" = "" ]; then
export LD_LIBRARY_PATH=$JAVA_HOME/jre/lib/$OS_ARCH/server:/opt/hadoop/build/
libhdfs:/usr/local/lib
fi

sh Tab 字元寬度: 8 第 36 行第 76 字 插入
```

```
# ./ fuse_dfs_wrapper.sh
```

出現 下圖表示成功



```
root@pc113: /opt/hadoop/build/contrib/fuse-dfs
檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)
Total time: 42 seconds
root@pc113:/opt/hadoop# sudo gedit /etc/ld.so.conf
root@pc113:/opt/hadoop# ldconfig
root@pc113:/opt/hadoop# bin/start-all.sh
starting namenode, logging to /var/hadoop/logs/hadoop-root-namenode-pc113.out
192.168.168.79: starting datanode, logging to /var/hadoop/logs/hadoop-root-datanode-pc113.out
starting jobtracker, logging to /var/hadoop/logs/hadoop-root-jobtracker-pc113.out
192.168.168.79: starting tasktracker, logging to /var/hadoop/logs/hadoop-root-tasktracker-pc113.out
root@pc113:/opt/hadoop# cd build/contrib/fuse-dfs
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# chmod +x fuse_dfs_wrapper.sh
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# mkdir /var/hdfs
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# vim fuse_dfs_wrapper.sh
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# gedit fuse_dfs_wrapper.sh
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# ./fuse_dfs_wrapper.sh
USAGE: ./fuse_dfs [debug] [--help] [--version] [-oprocted=<colon_seped_list_of_paths>] [rw] [-onotrash] [-ousetrash] [-obig_writes] [-oprivate (single user)] [-o] [-oserver=<hadoop_servername>] [-oport=<hadoop_port>] [-oentry_timeout=<secs>] [-oattribute_timeout=<secs>] [-odirect_io] [-onopoermissions] [no<other fuse option>] <mntpoint> [fuse options]
NOTE: debugging option for fuse is -debug
root@pc113:/opt/hadoop/build/contrib/fuse-dfs#
```


修改/etc/fuse.conf 文件，執行如下命令。

```
# chmod a+r /etc/fuse.conf
```

```
# vim /etc/fuse.conf
```

注：將 fuse.conf 文件中

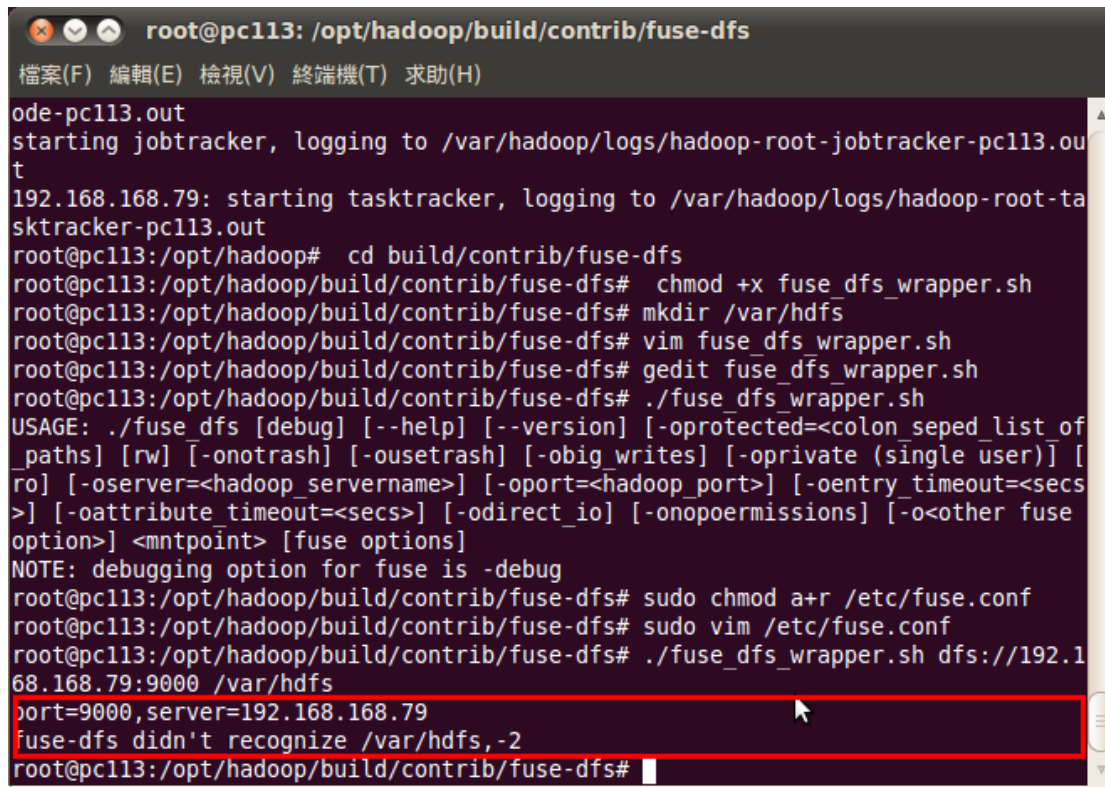
```
{{  
#mount_max = 1000  
#user_allow_other  
}}
```

前面的“#”號刪除。

掛載所指定的資料夾

```
# ./fuse_dfs_wrapper.sh dfs://自己的ip:9000 /var/hdfs
```

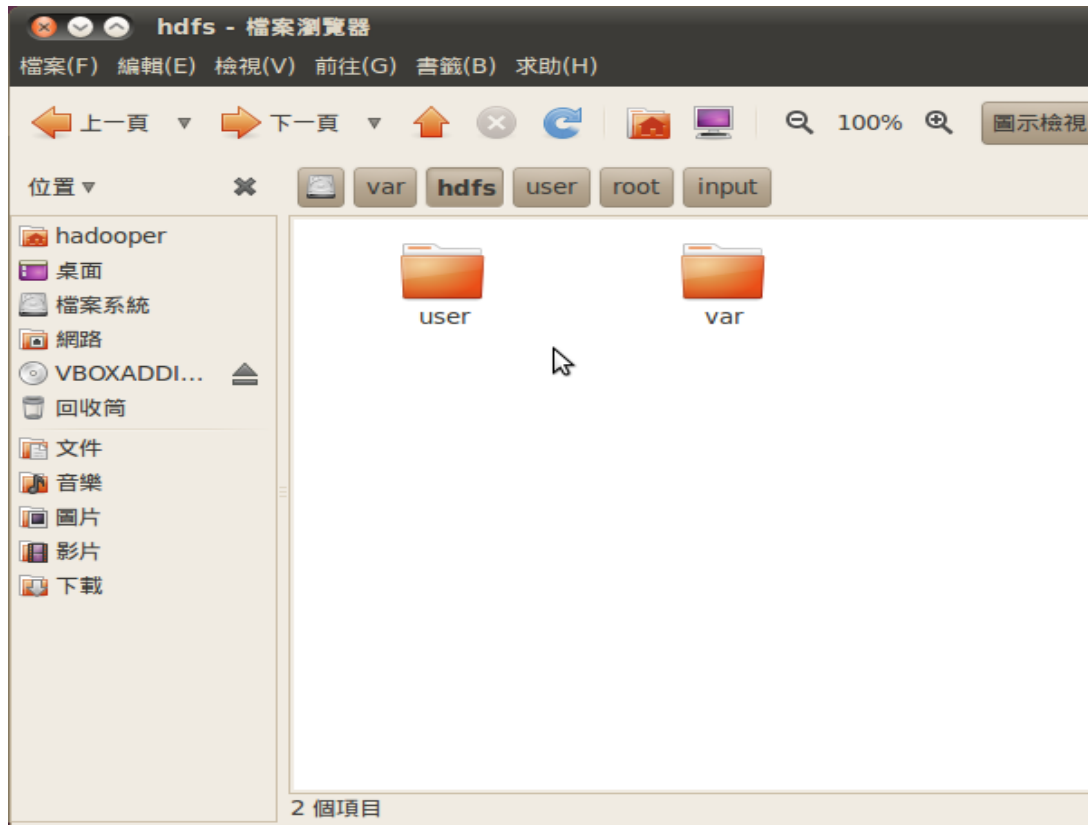
可看到



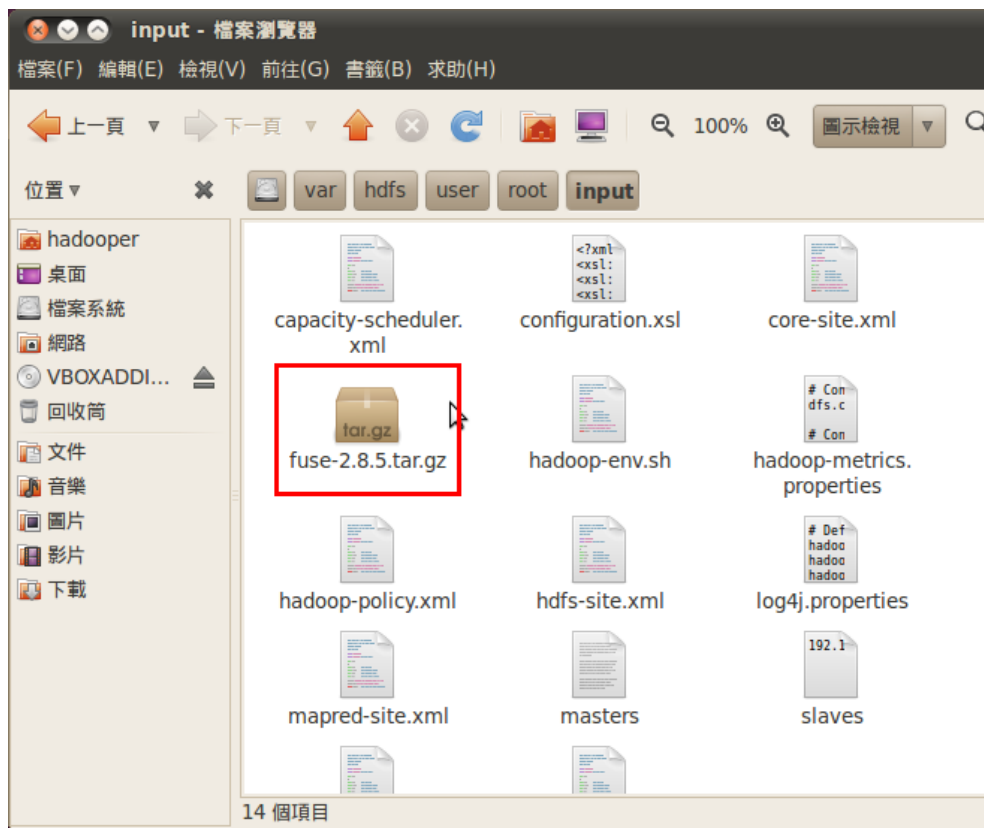
```
root@pc113: /opt/hadoop/build/contrib/fuse-dfs  
檔案(F) 編輯(E) 檢視(V) 終端機(T) 求助(H)  
ode-pc113.out  
starting jobtracker, logging to /var/hadoop/logs/hadoop-root-jobtracker-pc113.out  
192.168.168.79: starting tasktracker, logging to /var/hadoop/logs/hadoop-root-tasktracker-pc113.out  
root@pc113:/opt/hadoop# cd build/contrib/fuse-dfs  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# chmod +x fuse_dfs_wrapper.sh  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# mkdir /var/hdfs  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# vim fuse_dfs_wrapper.sh  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# gedit fuse_dfs_wrapper.sh  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# ./fuse_dfs_wrapper.sh  
USAGE: ./fuse_dfs [debug] [--help] [--version] [-oprotected=<colon seped list of  
_paths] [rw] [-onotrash] [-ousetrash] [-obig_writes] [-oprivate (single user)] [ro] [-oserver=<hadoop_servername>] [-oport=<hadoop_port>] [-oentry_timeout=<secs>] [-oattribute_timeout=<secs>] [-odirect_io] [-onopoermissions] [-o<other fuse option>] <mntpoint> [fuse options]  
NOTE: debugging option for fuse is -debug  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# sudo chmod a+r /etc/fuse.conf  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# sudo vim /etc/fuse.conf  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs# ./fuse_dfs_wrapper.sh dfs://192.168.168.79:9000 /var/hdfs  
port=9000, server=192.168.168.79  
fuse-dfs didn't recognize /var/hdfs, -2  
root@pc113:/opt/hadoop/build/contrib/fuse-dfs#
```

表示已掛載成功

打開 /var/hdfs 可看到以掛載的資料夾



例: mv 一個檔案到/var/hdfs



開啟 HDFS 頁面可看到

The screenshot shows a web browser window titled "HDFS:/user/root/input - Mozilla Firefox". The address bar contains the URL "http://192.168.168.79:50075/browseD". Below the address bar, there are navigation buttons and a search bar. The main content area displays a table of files in the HDFS directory. The file "fuse-2.8.5.tar.gz" is highlighted with a red border.

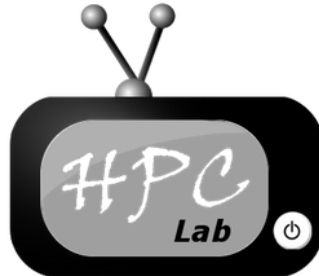
Name	Type	Size	Replication	Block Size	Modification Time	Permissions
capacity-scheduler.xml	file	3.84 KB	2	64 MB	2010-11-18 14:43	rw
configuration.xml	file	0.52 KB	2	64 MB	2010-11-18 14:43	rw
core-site.xml	file	0.24 KB	2	64 MB	2010-11-18 14:43	rw
fuse-2.8.5.tar.gz	file	491.55 KB	3	64 MB	2010-11-18 15:03	rw
hadoop-env.sh	file	2.37 KB	2	64 MB	2010-11-18 14:43	rw

資料直接上傳至 HDFS

七、實驗結果與展示

(一) 首頁：包含搜尋框、種類選擇按鈕、登入按鈕和註冊按鈕(後三者皆在右上角)。

Welcome ! [Categories](#) [Login](#) [Register](#)



Share Your Videos on the Cloud



(二) 註冊功能頁面：尚未註冊的使用者無法上傳影片，註冊後會寄發認證信，需至註冊時填寫的信箱中接收認證信。

Create an Account

Please enter your login information !

User name *

Password *

Verify Password *

Real name *

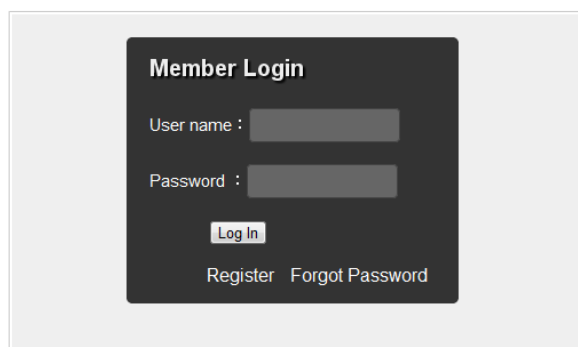
Email Address *

You are agreeing to the Terms of Service

Your preferred username to be used when logging in

TITLE	<input type="text"/>	
DIRECTOR	<input type="text"/>	
CAMERA	<input type="text"/>	
DATE	SCENE	TAKE
<input type="text"/>	<input type="text"/>	<input type="text"/>

(三)登入頁面：登入或註冊帳號後即可上傳影片並可觀賞影片後留言。

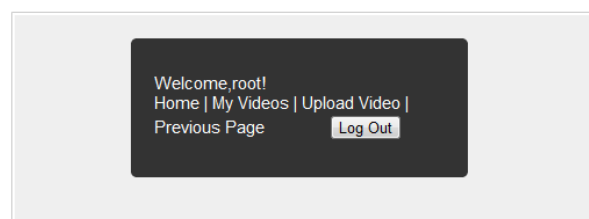


Member Login

User name :

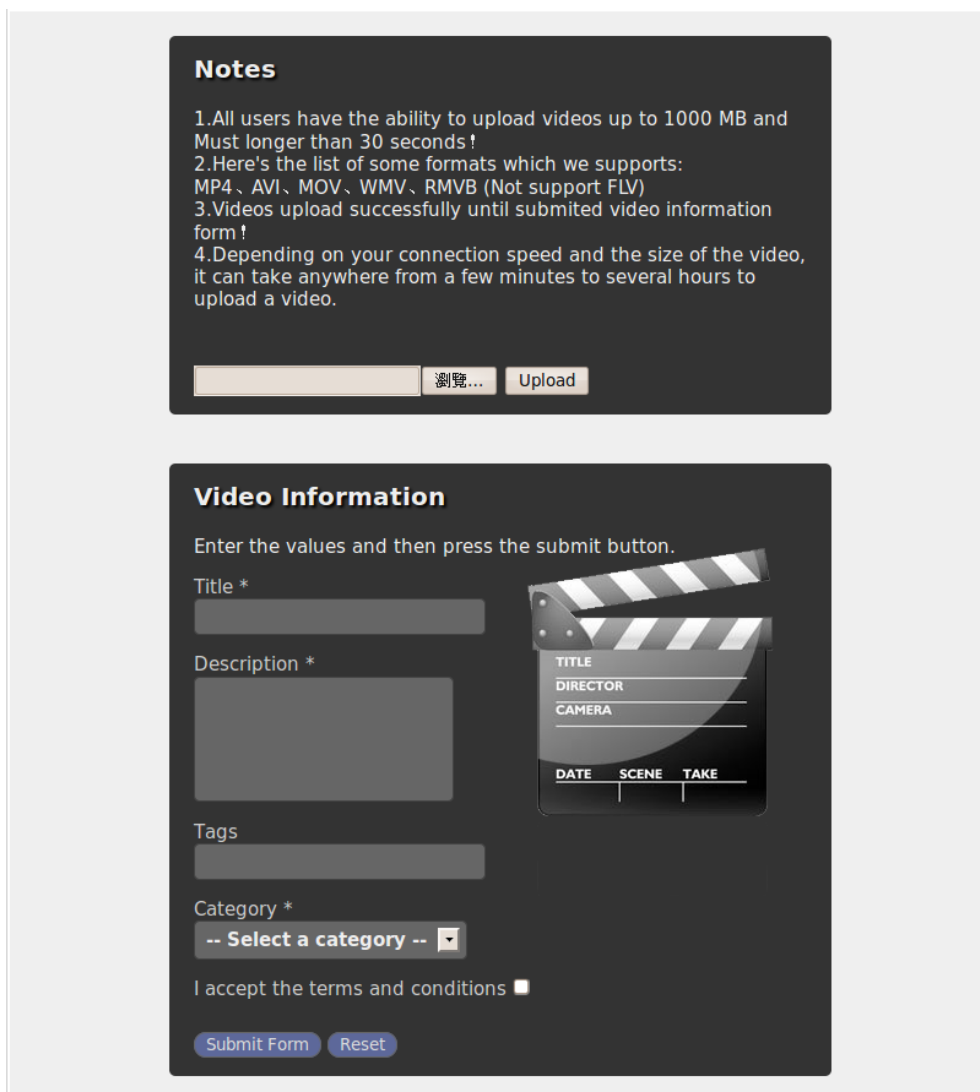
Password :

[Register](#) [Forgot Password](#)



Welcome,root!
Home | My Videos | Upload Video |
Previous Page

(四)上傳影片頁面：需上傳合乎格式之影片檔，並且表單要確實填寫才能送出。



Notes

- 1.All users have the ability to upload videos up to 1000 MB and Must longer than 30 seconds!
- 2.Here's the list of some formats which we supports: MP4、AVI、MOV、WMV、RMVB (Not support FLV)
- 3.Videos upload successfully until submitted video information form !
- 4.Depending on your connection speed and the size of the video, it can take anywhere from a few minutes to several hours to upload a video.

Video Information

Enter the values and then press the submit button.


Title *

Description *

Tags

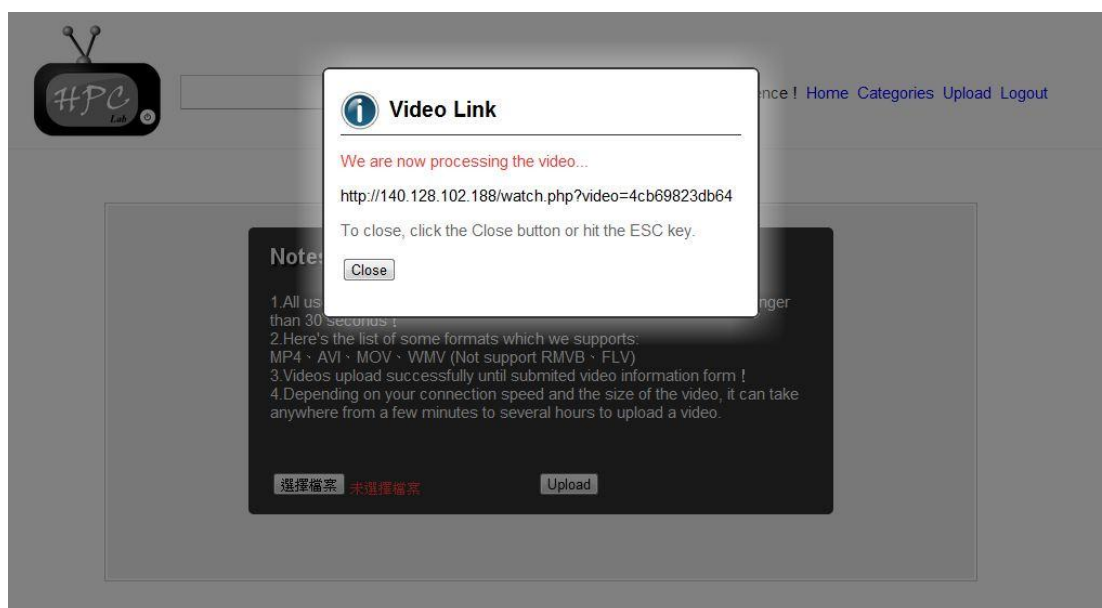
Category *

I accept the terms and conditions



TITLE
DIRECTOR
CAMERA
DATE SCENE TAKE

檔案上傳完成後即可前往影片連結網址



(五)影片播放頁面：觀看上傳的影片，使用 flash 播放器，影片比例為 16:9，影片解析度固定為 720P(1280×720)，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。



(六)播放頁面留言板：最多只能留言 300 字，下方會顯示剩餘可輸入字數，需登入會員並且要輸入驗證碼才能新增留言至留言板，可刪除或回覆留言。

Good!|



Enter The Code

295 characters remaining

All Comments(1)

NO.1	test	2010/10/15 07:34:21
	Good	

(七)影片種類選取頁面：可選取要觀看的影片種類，頁面會顯示該種類最新上傳的 15 部影片。



Search

Welcome! [Home](#) [Categories](#) [Login](#) [Register](#)

« Back

More videos »

Categories:

- [Autos & Vehicles](#)
- [Comedy](#)
- [Education](#)
- [Entertainment](#)
- [Film & Animation](#)
- [Gaming](#)
- [Howto & Style](#)
- [Music](#)
- [News & Politics](#)
- [People & Blogs](#)
- [Pets & Animals](#)
- [Science & Technology](#)
- [Sports](#)
- [Travel & Events](#)



杜拜水舞表演(The Dubai Fountain)

00:03:36

杜拜噴泉位在世界第一高樓杜拜塔湖面上，於去年2009年3月正式對外開放，目前水舞的表演曲目已經多達十幾首並且陸續的增加中，在噴泉湖面上設有二十五台的彩色投射機打光，五座圓形噴泉，估計水舞噴泉的造價高...

[Read more](#)



Norwegian sunrise

00:02:03

直升機空拍美麗的挪威日出~

[Read more](#)



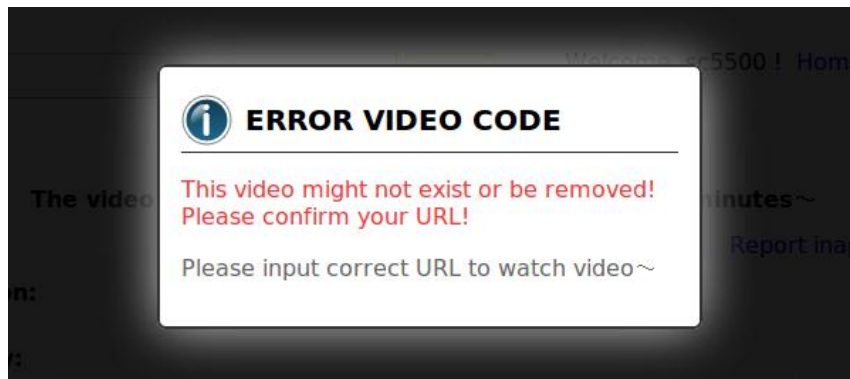
冰島火山

00:02:18

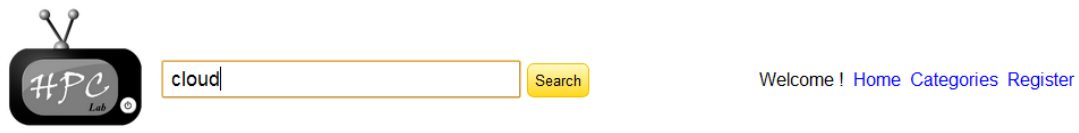
冰島艾維法拉(Eyjafjallajökull)火山造成歐洲航班大亂停飛人們只看到它達成的困擾，但卻有另一群人利用專業知識冒著生命危險去捕捉美麗的冰島火山~...

[Read more](#)

(八)影片代碼錯誤：若影片代碼錯誤或影片已被移除時會出現此頁面。



(九)搜尋頁面：在搜尋框輸入關鍵字，即可搜尋相關影片頁面。



"cloud" results 1 ~5 of 5



Basic Cloud Computing

00:03:37 / 15 views

該影片介紹了不少雲端作業系統。

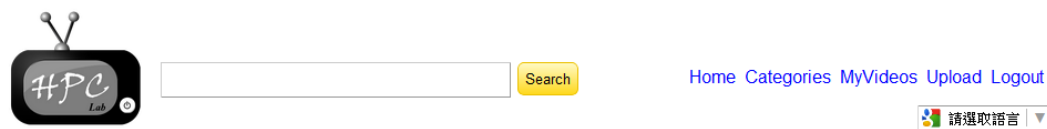


Cloud Computing

00:01:48 / 31 views

Google宣傳雲端的影片，內容展現在未來雲端科技如何從生活中幫助我們。

(十)影片管理頁面：可刪除和修改自己上傳的影片。



My Videos



Did You Know 社群媒體

2010-10-23 00:00:00

"你知道嗎？4.0版"，片中囊括了2009年10月在紐約召開Media Convergence Forum所收集的資料與數據，告訴你許多你不知道的事~...



碳交易的故事 The Story of Cap and Trade

2010-10-22 00:00:00

曾獲選為時代雜誌環保英雄的Annie Leonard團隊又推出「破交易的故事」，告訴你為何二氧化碳的「總量管制與交易制度」(Cap & Trade)，不能真正解決全球暖化危機，達到節能減碳的目的。...



Out of Sight

2010-10-21 00:00:00

This is a graduation Production made by three students graduated from the National Taiwan University...



Basic Cloud Computing

2010-10-21 00:00:00

該影片介紹了不少雲端作業系統。

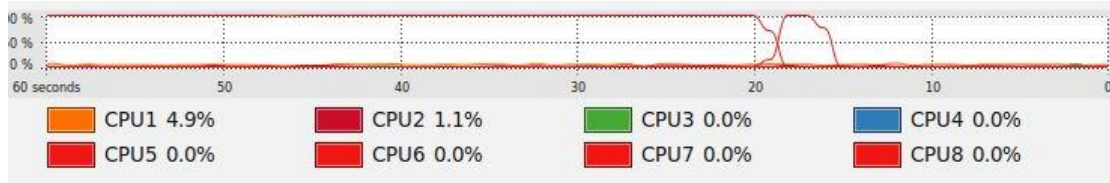
效能測試(FFmpeg-mt)

測試檔大小：149mb

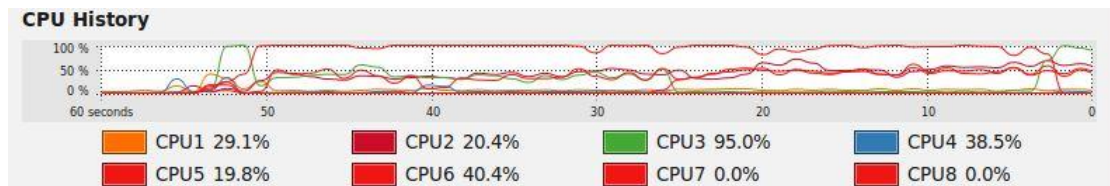
測試檔格式：avi

測試檔長度：9 分 56 秒

當 Thread=1 轉檔完成所花時間：7 分鐘



當 Thread=2 轉檔完成所花時間：3 分鐘 22 秒



八、結論

我們設計網頁的過程發現資料庫的資料表種類和欄位設置是一件要動腦筋且花不少功夫的部分，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。

除了網頁的程式碼製作外，資訊安全問題更是不容小覷，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。

有關搜尋的部份是由 HDFS 儲存索引資料，經過連結多台主機後，可將資料分散儲存在各個主機，並存在副本，降低因主機損壞所造成的資料遺失風險，而經設定後的 Nutch 搜尋引擎每經過固定時間便更新索引庫，確保索引資料和更新的資料(新上傳的影片)有相對應，使其盡量保持在最新的狀態，這正是雲端運算的優勢所在，使用者不必了解其結構，而上傳的資料卻可因為其分散儲存的特性而不易損壞。

此外，我們發現以多執行緒(Multi-Thread)的 FFmpeg 作上傳影片轉檔所需花費的時間明顯較原先的 FFmpeg 影片轉檔要短，增進了上傳影片的效率，由此可見轉檔程序平行處理是可行且有效的。除了減少使用者等待影片上傳後到可觀看的時間外，伺服器主機的運算效能也可以做到較充分的利用，這對管理者而言，更是省去了不少麻煩。

九、未來展望

目前本網站主要是支援一般電腦網路用戶為主，對於現在愈來愈多人擁有智慧型手持裝置，未來可以開發手機相關平台，讓網站可支援平台更加的多元，隨時隨地都可以存取網站。

功能的擴充也是未來可加強的部份，現今的影音網站若只有觀看影片和上傳影片已經無法滿足使用者的需求，所以勢必要增加一些額外的功能，本網站已提供影片社群分享的功能(Facebook、Plunk、Twitter 等)，未來可望加上一些更新奇且實用的功能，例如：遊戲、購物連結等。

而網站管理機制也可以再改善，開發管理者頁面，方便網站管理者可以更直接且有效地管理網站，期望這個影音網站未來能克服某些影片格式無法辨識的問題，對所有使用者上傳的影片格式皆能做轉檔播放。

另外，影片支援上傳檔案大小可以根據實際需求來做調整，但是支援太大的影片會使伺服器的主機負荷量大幅增加，若未來的硬體資源持續擴增，就可以允許任意大小的影片上傳。

當然，網站還有一個十分重要的部份，那就是資訊安全；雖然網站在設計其中填入表格時，用了一些防護措施，如限制輸入字數和一些內容，避免一些有害程式或使用者的惡意破壞；但道高一尺魔高一丈，我們仍有相當多的資安議題是需要學習的，這部分亦是本專題內容未來需持續維護和修正的地方。

十、參考資料及網站

1.Cloud Computing 雲端運算 – MMDays

<http://mmdays.com/2008/02/14/cloud-computing/>

2.Cloud Computing(NCHC Cloud Computing Research Group)

<http://trac.nchc.org.tw/cloud>

<http://trac.nchc.org.tw/cloud/wiki/>

3.雲端運算基礎課程 (Hadoop 簡介、安裝與範例實作)

<http://trac.nchc.org.tw/cloud/wiki/NCHCcloudCourse090914>

<http://trac.nchc.org.tw/cloud/wiki/NCHCcloudCourse090428>

4.Hadoop_Lab5 – Cloud Computing

http://trac.nchc.org.tw/cloud/wiki/Hadoop_Lab5

5.Hadoop_Lab7 – Cloud Computing

http://trac.nchc.org.tw/cloud/wiki/Hadoop_Lab7

6.waue-2010-0211 – Cloud Computing

<http://trac.nchc.org.tw/cloud/wiki/waue/2010/0211>

7.waue-Hadoop_Eclipse – Cloud Computing

http://trac.nchc.org.tw/cloud/wiki/waue/Hadoop_Eclipse

8.Encoding - h264 – Trac

<http://h264.code-shop.com/trac/wiki/Encoding>

9.waue-2009-nutch_install – Cloud Computing

http://trac.nchc.org.tw/cloud/wiki/waue/2009/nutch_install

10.雲端運算 Cloud Computing 的概念與應用

<http://eblog.cisanet.org.tw/post/Cloud-Computing.aspx>

11.Index of -hadoop-core

<http://apache.stu.edu.tw/hadoop/core/>

12.[Linux] 安裝單機版 Hadoop 0.20.1 Single-Node Cluster (Pseudo-Distributed) @

Ubuntu 9.04 @ 第二十四個夏天後 痞客邦 PIXNET

<http://changyy.pixnet.net/blog/post/25245658>

13.svn - Revision 61969 -trunk-src(google chrome 的 source code)

<http://src.chromium.org/svn/trunk/src/>

14.Cluster Setup

http://hadoop.apache.org/common/docs/current/cluster_setup.html

15.Cloudera » Apache Hadoop for the Enterprise

<http://www.cloudera.com/>

16.中國文化大學圖書館

<http://www.lib.pccu.edu.tw/>

17.Media College - Video, Audio and Multimedia Resources

<http://www.mediacollege.com/>

18.傳說中的挨踢部門 Ubuntu Server 安裝筆記 - yam 天空部落

<http://blog.yam.com/leo2016/article/19679466>

19.LongTail Video Home of the JW Player

<http://www.longtailvideo.com/>

20.red5 - Project Hosting on Google Code

<http://code.google.com/p/red5/>

21.Flowplayer - Flash Video Player for the Web

<http://flowplayer.org/tutorials/introduction-to-streaming-servers.html>

<http://flowplayer.org/tutorials/ffmpeg.html>

22.Pseudostreaming - Flowplayer streaming plugin

<http://flowplayer.org/plugins/streaming/pseudostreaming.html>

- 23.RTMP - Flowplayer streaming plugin
<http://flowplayer.org/plugins/streaming/rtmp.html>
- 24.Basic pseudostreaming setup
<http://flowplayer.org/demos/plugins/streaming/>
- 25.Apache Tomcat runtime.getRuntime().exec() Privilege Escalation (win) - Forums
<http://www.governmentsecurity.org/forum/index.php?showtopic=30916>
- 26.Chmod from Runtime.Exec
<http://www.velocityreviews.com/forums/t133458-chmod-from-runtime-exec.html>
- 27.我,一張一張照...FFMpeg 編譯安裝 (Ubuntu 懶人包) - 樂多日誌
<http://blog.roodo.com/cinatit/archives/6026095.html>
- 28.【系統】Lighttpd 安裝 H264 Streaming Module @ My Life 隨意窩 Xuite 日誌
<http://blog.xuite.net/chingwei/blog/32821961>
- 29.Eclipse.org home
<http://www.eclipse.org/>
- 30.東方和風語 cannot send session cache limiter-headers already sent 錯誤
<http://hatsukiakio.blogspot.com/2009/07/cannot-send-session-cache-limiter.html>
- 31.課程 982-219095 行動與雲端運算應用
<http://moodle.ncnu.edu.tw/course/view.php?id=11061>
- 32.Source Codes - MapReduce -1
https://docs.google.com/View?id=dckwxdq9_63c8z4kfw4
- 33.jQuery TOOLS - The missing UI library for the Web
<http://flowplayer.org/tools/demos/index.html>
- 34.在 Linux 作業系統下 phpMyAdmin 的安裝方法
<http://163.20.160.21/xoops22/t167/phpmyadmin/list.htm>
- 35.Overview (Hadoop-common 0.21.0 API)
<http://hadoop.apache.org/common/docs/current/api/overview-summary.html>
- 36.PHP 中调用外部程序，及其参数与返回值 - bunny 的技术之旅 - 51CTO 技术博客
<http://cutebunny.blog.51cto.com/301216/58597>
37. 王鵬，《雲端運算的關鍵技術與應用實例》，佳魁資訊，民國 99 年 2 月 25 日
- 38.Linux 下如何安裝 ffmpeg - yezi - JavaEye 技术网站
<http://yezi.javaeye.com/blog/139399>

39. 臥龍小三，〈Linux shell 程式設計實務〉，悅知文化，民國 98 年 8 月 20 日
40. 迴圈的語法共有三種分別是
<http://hyh.mis.dwu.edu.tw/jsp/array.htm>
41. ffmpeg-mt in FFmpeg – Gitorious
<http://gitorious.org/ffmpeg/ffmpeg-mt>
42. hadoop 官方網站(Single Node Setup)
http://hadoop.apache.org/common/docs/current/single_node_setup.html
43. crawlzilla - Project Hosting on Google Code
<http://code.google.com/p/crawlzilla/>
44. Crawlzilla 系統安裝教學
http://trac.nchc.org.tw/cloud/wiki/crawlzilla/install_zh
45. Crawlzilla 系統管理介面使用說明
http://trac.nchc.org.tw/cloud/wiki/crawlzilla/sysmanagement_zh
46. Crawlzilla 網頁操作說明
http://trac.nchc.org.tw/cloud/wiki/crawlzilla/webui_zh
47. YouTube - CrawlZilla 叢集安裝與設定示範
<http://www.youtube.com/watch?v=bRWQ3BXEj4A>
48. 阿碼外傳－阿碼科技非官方中文 Blog HackAlert Enterprise - 企業級掛馬監控方案
<http://armorize-cht.blogspot.com/2008/10/hackalert-enterprise.html>
49. 简介百度优化应多久更新一次及深度
<http://www.paimingyouhua.com/china/649.Html>
50. 站内链接优化 - 上海 SEO 优化网站优化公司
<http://shanghai.ckcf.com.cn/shanghai-Wangzhanyouhua/ZhanNeiLianJieYouHua/>
51. 鳥哥的 Linux 私房菜
<http://linux.vbird.org/>
52. Dongogooooo » [Linux] 使用 crontab
<http://matrix.csie.org/blogs/dongogo/2010/03/06/linux-%E4%BD%BF%E7%94%A8-crontab/>
53. HTM 裡「上一頁」的語法 - HTML-DHTML - 程式設計俱樂部
<http://www.programmer-club.com.tw/showSameTitleN/html/1162.html>

- 54.Shuffle Error Exceeded MAX_FAILED_UNIQUE_FETCHES; bailing-out(页 1) - Hadoop 综合 - Hadoop 技术论坛 - Powered by Discuz! Archiver
<http://www.hadoopor.com/archiver/tid-513.html>
- 55.hadoop-0.20.1 部署手记
<http://blog.formyz.org/?p=83>
- 56.HowTos-Tomcat - Scalix Wiki
<http://www.scalix.com/wiki/index.php?title=HowTos/Tomcat>
- 57.Nutch 本地索引更新策略_我的空间_百度空间
<http://hi.baidu.com/zhumulangma/blog/item/2554de2a0b57cf25d42af1a1.html>
- 58.HDFS over FTP - Hadoop
<http://www.hadoop.iponweb.net/Home/hdfs-over-ftp>
- 59.FUSE-HDFS 安装_我思故我睿..._百度空间
<http://hi.baidu.com/siruiyang/blog/item/4547f808fe6502a32fddd46e.html>
- 60.waue-2009-1005 – Cloud Computing
<http://trac.nchc.org.tw/cloud/wiki/waue/2009/1005>
- 61.fuse-hdfs_逆风行_新浪博客
http://blog.sina.com.cn/s/blog_5cf546320100is1d.html
- 62.MountableHDFS - Hadoop Wiki
<http://wiki.apache.org/hadoop/MountableHDFS>
- 63.fuse-dfs 的设定手册 - sery - 51CTO 技术博客
<http://sery.blog.51cto.com/10037/121110>
- 64.Hadoop Hdfs 配置 挂载 hdfs 文件系统 (二),Linux 应用技巧,Linux 系列教程,Linux
http://dev.firnow.com/course/6_system/linux/linuxjq/20100313/198374_2.html
- 65.Hadoop 分布式文件系统 (HDFS) 初步实践 - 阿泰的菜园
<http://huatai.me/?p=352>
- 66.Filesystem in Userspace Download Filesystem in Userspace software for free at SourceForge.net
<http://sourceforge.net/projects/fuse/>
- 67.Nutch - Wikipedia, the free encyclopedia(英文维基百科)
<http://en.wikipedia.org/wiki/Nutch>
- 68.Lighttpd - 维基百科,自由的百科全书(中文维基百科)
<http://zh.wikipedia.org/zh-tw/Lighttpd>

69.FFmpeg - 維基百科,自由的百科全書(中文維基百科)
<http://zh.wikipedia.org/zh-tw/FFmpeg>

70.PHP - 維基百科,自由的百科全書(中文維基百科)
<http://zh.wikipedia.org/zh-tw/PHP>

71.JSP - 維基百科,自由的百科全書(中文維基百科)
<http://zh.wikipedia.org/zh-tw/JSP>

72.FUSE - 維基百科,自由的百科全書(中文維基百科)
<http://zh.wikipedia.org/zh-tw/FUSE>

73. Filesystem in Userspace - Wikipedia, the free encyclopedia(英文維基百科)
http://en.wikipedia.org/wiki/Filesystem_in_Userspace