

第二章 研究背景

2-1 靜態網頁運作

Apache 就是一種讓電腦能夠提供網頁服務的開放源碼軟體，它的開放式架構，加上已經內建的許多模組，讓功能的增減變得相當容易方便，在效能的調整上也提供了幾個設定作調校以配合實際應用的情況。也因此，Apache 在網頁伺服器的市場佔有率超過了一半[14]。

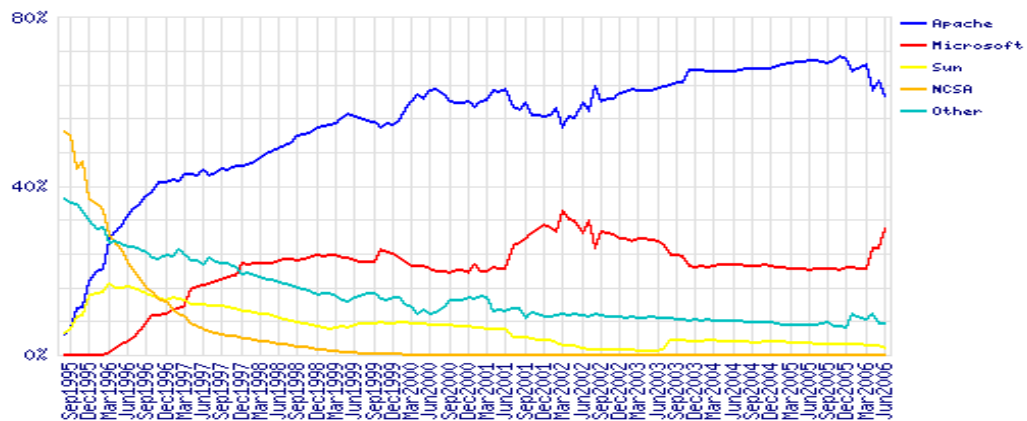


圖 1：網頁伺服器軟體佔有率。

當一個網頁伺服器提供服務的時候，使用者可以使用瀏覽器輸入該伺服器的 URL 連結[15]，經由 HTTP 通訊協定即可取得網頁建構者預先設計安放好要給外來訪客觀看的頁面或資料。

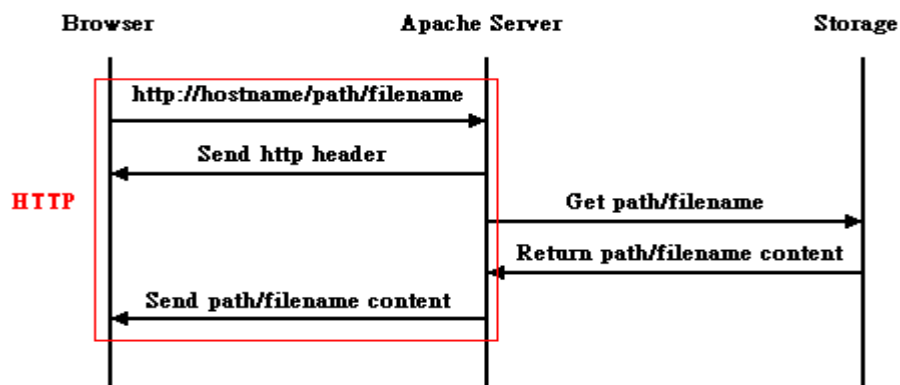


圖 2：靜態網頁運作

2-2 動態網頁運作

隨著資訊技術的發展，單純的靜態網頁已經滿足不了日益增多的資料量和使用者的需要，所以發展出 CGI 的技術來進行資料輸入輸出的處理，能夠傳回資料處理後的結果，甚至是傳回更新的內容而成為動態網頁(dynamic web page)的運作機制。之後更因為 Perl[16]的發展和技術的進步，直譯式語言(interpret language)也就被應用在 CGI 資料處理的用途上，也因此出現了 PHP、Ruby[17]和 Python[18]這些不同流派的直譯式程式語言，各有其特性而有不同的使用者群，而 PHP 甚至能做成 Apache module 來配合 Apache 一併使用，以提高執行效率。

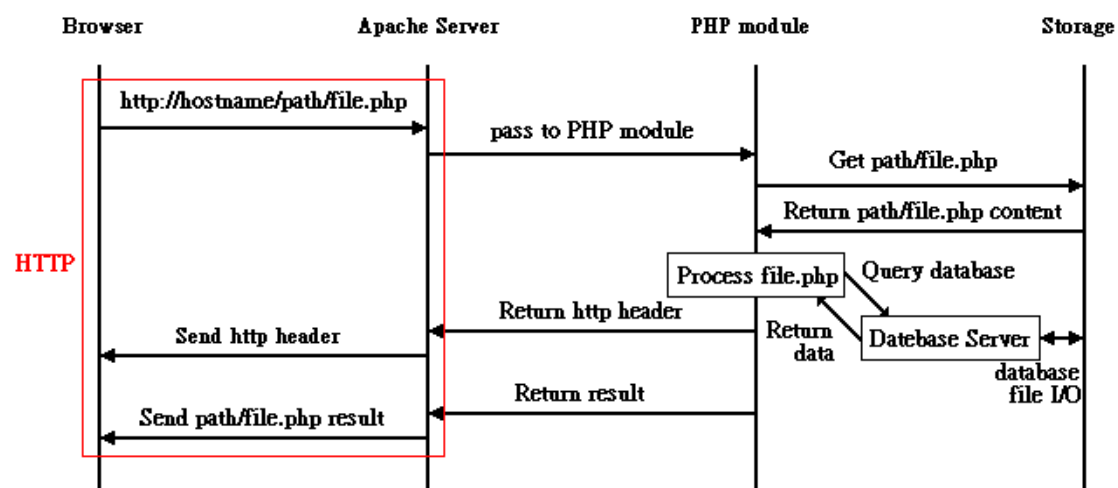


圖 3：動態網頁運作 (以 PHP 為例)

單就使用簡便和維護難易的程度來說，PHP 的入門門檻算是相當低的。最主要的原因是因為 PHP 有提供一個能夠非常方便的網頁介面可以查詢指令用法和程式範例，對初學者來說，這是可以加速學習的一種輔助方式，其他語言雖然也有說明文件和操作手冊，可是使用上的便利性就沒有像 PHP 那麼迅速有效。其次，PHP 發展到目前為止，幾乎所有的資料庫的存取都有支援，無論是開放源碼資料庫還是商業用的資料庫[19]。在共通的程式語法上，要轉換到任一種資料庫也不會有太大的問題。

2-3 資料庫軟體的選用

資料庫的發展從很久以前就有 dbm 這種直接以檔案為單位來處理的資料庫程式套件，不過，在網路多人同時運作的環境下，以直接存取檔案的方式來處理資料，會有資料同步(data synchronization)、檔案鎖定(file locking)和競賽冒險(race hazard)三大問題存在，所以後來就發展出以網路連線來接受資料處理的資料庫軟體，解決因直接對資料檔案進行存取而產生的這些問題。而在開放源碼軟體方面就有 MySQL 和 PostgreSQL[20]這兩套可以使用。

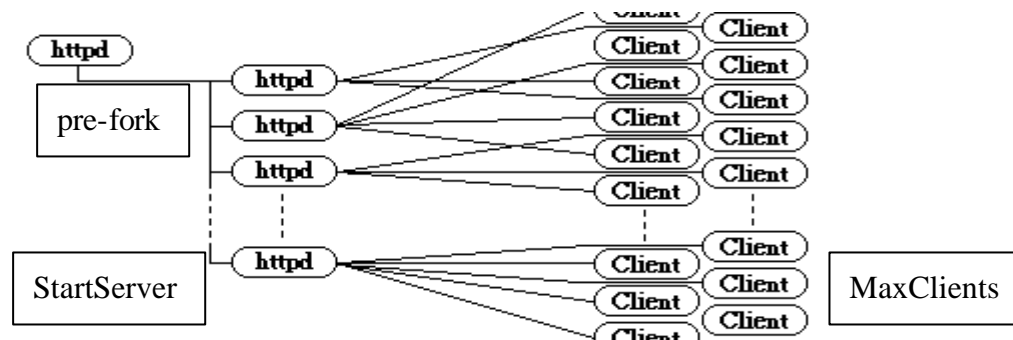
MySQL 在操作和維護兩方面來說，比 PostgreSQL 來得直覺且方便許多，雖然已經有一些的測試資料顯示 PostgreSQL 效能和功能上比 MySQL 強大許多，可是在一般使用的情況下，還是以入門迅速且操作簡便為最優先考量的因素，加上 phpMyAdmin[21]這個軟體套件，更能夠讓不懂資料庫的一般人，能夠方便且簡單的使用資料庫軟體和一般指令。

2-4 Apache 一般調校

Apache 一般的調校方式是以修改 httpd.conf 的設定值來進行。

MinSpareServers	閒置等待服務的子程序的最小數量
MaxSpareServers	閒置等待服務的子程序的最大數量
StartServers	啟動時要 pre-fork 的子伺服器程序數量
MaxClients	同一時間內最多能服務多少個連線數量

表 1：Apache 設定檔案 httpd.conf 中，可調校的參數。



增加 MaxClients 的數量以提高總使用者數量上限，如果需要提高上限超過 512 的話，那就得去修改原始程式碼，提高上限值才能使用；StartServers 增加 Apache 啟動時預先 pre-fork 的程式數量，以撐住啟動或是重啟時突然衝入的大量連線；MinSpareServers 和 MaxSpareServers 這兩個項目則視伺服器記憶體大小和運作時的記憶體用量來調整在系統中的 Apache 維持存在的數量，以提升 Apache 的服務效率 [22]。

此外，從設定檔案中註解或是拿掉用不到的 Apache module，降低 Apache 運作時的記憶體使用量，也是一種提昇系統效率和穩定性的做法。只要程式在運作時，不會因為實體記憶體耗盡而動用到系統的 swap 空間，那麼就不會有因為進行 swap I/O 硬碟而使程式停住數秒的情況，甚至是避開因 swap fail 或是 vm page fault 造成程式不正常結束，導致使用者取得錯誤或是空白的網頁內容。

2-5 PHP 一般調校

PHP 的調校方式大概分成兩種，一種是耗費一些記憶體加掛一些外部程式來取代 PHP 原本內建的部分功能，減少花費在處理動態網頁程式碼的時間。

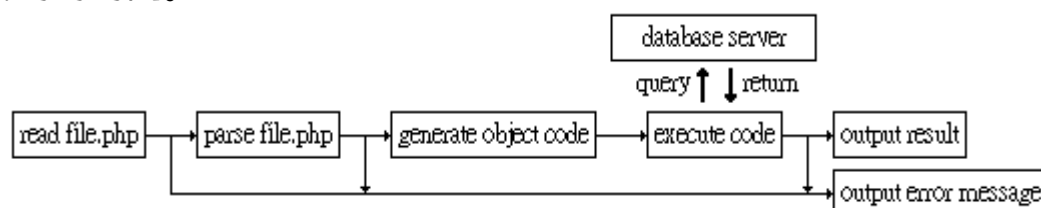


圖 5：PHP 運作流程圖

Zend Optimizer[23]這個軟體可以加速 PHP 程式碼在 parse 階段的速度以及產生更有簡潔的 object code 來執行，以提升效率。Turck MMCache[24]則是以儲存產生出來的 object code 的方式，在原本 PHP 檔案沒有改變的前提下，只要處理執行到相同的檔案，就可以跳過 parse 和產生 object code 的兩個步驟而直接以儲存起來的 object code 來執行，有效的加快執行速度。

另一種方式則是注意 PHP 程式碼的寫法，不讓 PHP 做多餘的執行展開(runtime expression)動作，甚至可以改用別的 PHP 指令來得到同樣的結果，在 Byster.NET[25]已經有一份以各種指令和語法，測試其執行時間的效能報告供參考。

2-6 MySQL 一般調校

MySQL 本身的調校方式一般來說也有兩種。一種是讓 MySQL 以 multithreading 的方式運作，而不是單獨一支 MySQL daemon 來處理所有的指令需求，避免讓 MySQL 卡在很花時間的 Query 指令上，就能夠讓 MySQL 可以執行得更有效率。

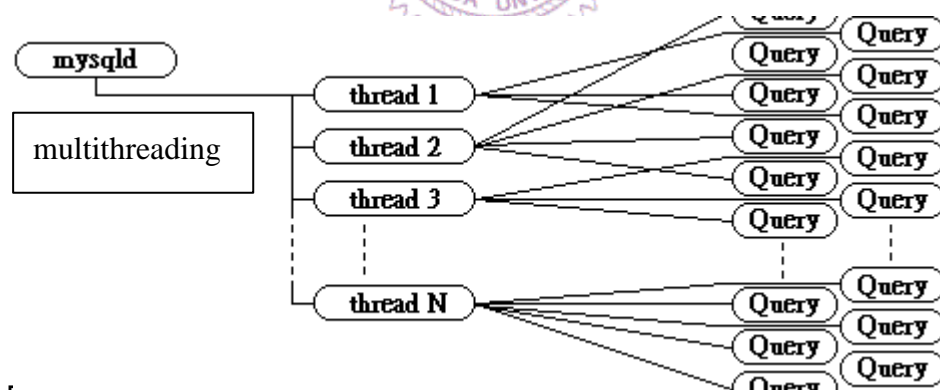


圖 6：MySQL 運作狀況圖

另一種則是修改 MySQL 的設定檔案，加大所有使用記憶體使用限制的上限數字，務必讓所有 SQL 指令執行的動作能夠有足夠的記憶體空間來存放資料，而不會有使用硬碟暫存檔案的情況，甚至讓所有資料庫裡的資料都可以快取(cache)一份在記憶體內直接取用，大量減少硬碟 I/O 的動作，有效提升資料庫的執行效率。

MySQL 套件中有提供四個基本參考設定檔案：

`my-small.cnf`

`my-medium.cnf`

`my-large.cnf`

`my-huge.cnf`

這四個檔案是針對各種不同大小的系統記憶體總量，設定不同的參數給 MySQL 的各項用途來做適當的運用，使用者只需要將這些設定檔案放到指定的位置，重新啟動 MySQL 後就立刻生效。如果有特殊的應用或是特別的運作環境，那麼就需要使用者額外調整設定檔案中的數值。

