

多層式網頁應用程式的性能評估 Performance Evaluation for Multi-Tiered Web Application

朱仁貴

Ren-Guey chu

黎明技術學院電機工程系

Department of Electrical Engineering, Lee-Ming Institute of Technology

吳政遠 王永鐘

Chih-Yuan Wu, Yung-Chung Wang

台北科技大學電機工程系

Department of Electrical Engineering, National Taipei University of Technology,

王振興

Jenn-Shing Wang

景文科技大學資訊工程系

Department of Computer Science and Information Engineering, Jinwen University
of Science and Technology

摘 要

瀏覽人數過多所造成的網站過載，其回應時間隨請求到達率增加而明顯增長，適當地設定網頁伺服器，可避免網站過載以維持既定的回應時間。本論文開發一個效能分析工具軟體，利用此軟體分析的結果設定阿帕契(Apache)模組之控制參數。經由多層式網頁應用程式的實測結果證明，本論文的效能分析工具模組確實能夠分析網站的性能參數。

關鍵詞：網頁應用程式、多層式、阿帕契

Abstract

Website overload caused by the excessive number of visitors, the response time with the request arrival rate increased will be significantly increase. To avoid site overload in order to maintain the response time, an properly set the Web server is necessary. This paper develops a performance analysis software. We can set the control parameters of the Apache module according to the analyze results of this



software. The performance analysis module is effectively able to analyze the performance parameters of the site by the measured results of the multi-tiered web application.

Key Words: web application, multi-tiered, Apache



1. 簡介

全球資訊網(World Wide Web, WWW)發明於 1990 年代初期，超文件標示語言(HyperText Markup Language, HTML)、超文件傳送協定(HyperText Transfer Protocol, HTTP)、網頁伺服器和瀏覽器四項關鍵元件的發展造就今日的網路世界[1]。2005 年 O'Reilly 提出 Web 2.0 的應用概念，將網站視為平台使用，使全球資訊網推向另一個世代[2]。此後提倡多樣化網際網路應用程式(Rich Internet Application, RIA)的使用概念，以網站為基礎開發各類型網頁應用程式(Web Application)，利用網際網路(Internet)部署應用程式於瀏覽器、電腦與行動裝置之上[3]。目前人們依賴全球資訊網的程度已超過 90 年代初期。

全球資訊網是建置於網際網路之上，網站無時無刻接受來自世界各地使用者的請求。在現今高度全球化與網路化的時代，當發生受人們矚目的事件時，網站將會因為瀏覽人數突然地增加而造成過載，其過載原因是瀏覽人數超出系統所設計之容量。普遍的案例有：新聞網站在奧運舉辦期間、政治選舉開票時或發生重大災難後。購物網站的特賣期間或標錯過低價格的搶購。與一般公司網站在產品發表的期間。網站發生過載時，回應使用者請求的時間將會增長，甚至造成連線失敗或網站當機。因為過長的等待時間會讓使用者選擇離開，更進一步造成商業上的損失或是對該網站的負面評價。瀏覽人數突然增加所造成的網站過載必須設法解決，由於這類型的過載問題很難預估瀏覽人數和發生時間，因此建置網站時的容量規劃較無法有效預防，若設計較大的容量將不符合經濟效益。當採用擴充的解決方式則需要時間設置，因而無法即時地處理突然增加的負載。

網頁應用程式執行於網頁伺服器之中，一個網頁伺服器可視為由佇列(Queue)與服務器(Server)組成的佇列系統(Queueing System)[4][5]。佇列系統中的回應時間(Response Time)與到達率(Arrival Rate)之關係如圖 1.1 所示，隨著請求的到達率增加，其回應時間會明顯增長[6]。對網站而言，當請求到達率的增加超出額定值之後，其影響將使服務品質(Quality-of-Service, QoS)顯著下降。

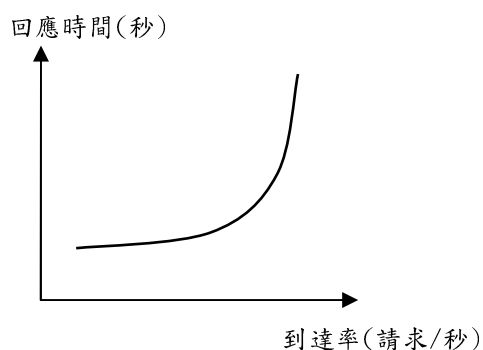


圖 1.1 佇列系統回應時間與到達率之關係

本研究目的將開發一個效能分析工具軟體，利用此軟體分析的結果設定 Apache 模組之控制參數。首先於第二章介紹相關研究，說明多層式架構以及 Apache 網頁伺服器軟體。第三章介紹本論文設計的效能分析工具，說明程式功能和量測方式。接著進行模組安裝和實際量測，評估本研究之控制效果，由第四章說明實驗平台規格與實測流程，並分析實測的數據結果。最後，第五章結論說明本研究的發現和未來研究方向。

2. 多層式網頁應用程式

目前的網頁應用程式廣泛採用多層式架構建置，其優點是可以提高網站的擴充性(Scalability)、可用性(Availability)與安全性(Security)，並且容易與舊有系統進行整合[7]。大型的網頁應用程式採用三層式架



構為主，典型的三層式架構如圖 2.1 所示 [8]。

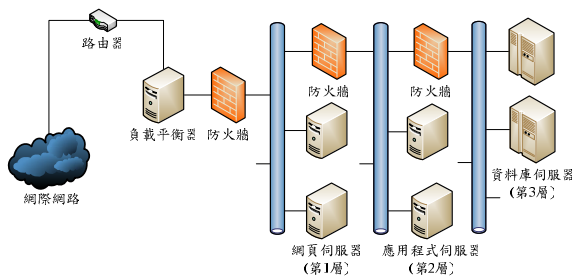


圖 2.1 典型的三層式網頁應用程式架構

第一層為展現層 (Presentation Layer)，負責提供使用者介面呈現的資料，內容以不需程式執行的靜態網頁資料為主，如 JPEG 圖檔、GIF 圖檔、HTML 檔案、JavaScript 檔案以及串接樣式表 (Cascading Style Sheets, CSS) 等。展現層由網頁伺服器提供三項主要功能：(1)接收來自使用者瀏覽器的請求。(2)轉送複雜的動態內容請求至第二層。(3)接收來自第二層的回應，並將此回應轉送至瀏覽器。常用的伺服器軟體有 Apache HTTP Server、Microsoft Internet Information Server (IIS)、nginx 與 lighttpd 等。第二層為商業邏輯層 (Business Logic Layer)，或稱應用層 (Application Layer)，此層功能實現一系列商業規則與工作流程的運算，如進銷存貨管理、生產流程管理等。應用層接收來自展現層的請求，依據此請求向第三層搜尋資料，其回應的資料依據商業邏輯處理產生動態網頁再轉送回第一層。常用的軟體平台有 Apache Tomcat、Sun GlassFish Enterprise Server、JBoss Application Server、Zend Server、Microsoft IIS 與 IBM WebSphere 等。第三層為資料服務層 (Data Service Layer)，通常由資料庫管理系統 (Database Management System, DBMS) 負責搜尋與存取資料，此層負責接收來自應用層的資料請求並回應。常用的資料庫管

理系統有 MySQL、Oracle、PostgreSQL、Microsoft SQL Server 與 IBM DB2 等。三層式架構最前端由負載平衡器 (Load Balancer) 接收來自網際網路的請求，平均分配請求負載給後端的伺服器，避免單一伺服器超載而造成系統的瓶頸，進而提升整體多層式架構的效能。各層之間由防火牆 (Firewall) 隔離，僅有授權的伺服器能相互連線，確保內部設備免於遭受網際網路的攻擊，系統的安全性因此提升，而且每層之中由多台伺服器組成，故可以確保系統的可用性，避免單一伺服器故障而造成系統停擺，另外，各層之間使用網路連接而具有低耦合特性，有利於系統的擴充性。

3. 性能評估工具之設計

3.1 軟體功能介紹

本論文以 Microsoft Visual Studio 開發效能分析工具，採用 Microsoft .NET Framework 3.5 與 C# 程式語言撰寫，軟體介面如圖 3.1 所示，此軟體目的為分析負載對於伺服器的效能影響，其影響情形可由此工具的數據與圖檔分析得知。

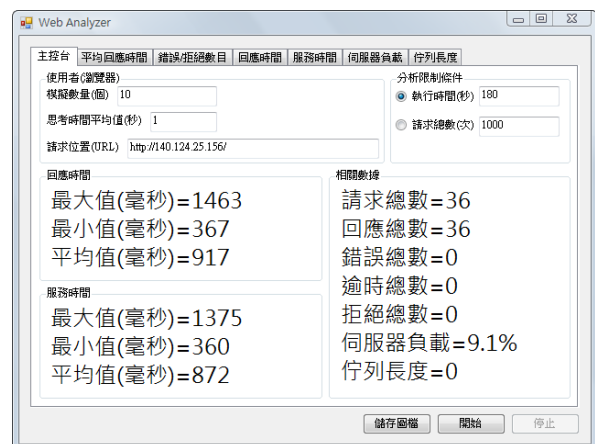


圖 3.1 效能分析工具的執行畫面



效能分析工具可以設定之參數有：使用者數量、思考時間和請求位置，並且可以限制量測之時間或請求次數，當限制條件符合時，本工具將停止繼續量測的工作。量測的數據則包括：回應時間、請求總數、回應總數、錯誤總數、逾時總數與拒絕總數，此數據將提供網站管理員了解網站是否過載。倘若欲測試系統的 Apache 伺服器已經安裝本論文的允入控制模組，此模組的監測功能將進一步提供伺服器的相關資訊。因此，效能分析工具可藉由分析 HTTP 標頭欄位內容取得每次 HTTP 請求的服務時間、伺服器負載與佇列長度之資訊。

本分析工具定義相關量測數據如下所述：

- (1) 請求總數：於執行量測的時間內所發出的 HTTP 請求次數之和，或等於限制條件的請求總數。
- (2) 回應總數：所發出的 HTTP 請求中，成功回應的次數總和。
- (3) 錯誤總數：所發出的 HTTP 請求中，TCP 連線發生錯誤的次數總和。
- (4) 逾時總數：所發出的 HTTP 請求中，未在逾時時間以內回應的次數總和。
- (5) 拒絕總數：所發出的 HTTP 請求中，收到回應 HTTP 狀態碼 503 的次數總和。
- (6) 伺服器負載：每次回應的伺服器負載總和除以回應總數。
- (7) 佇列長度：每次回應的佇列長度總和除以回應總數。

除數據資料外，本軟體具有即時資訊的繪圖功能如圖 3.2 所示，並且可另存圖檔做為紀錄之用。當軟體每次接收 HTTP 回應之後，其繪圖內容將隨即更新，利用圖形可以檢視系統即時狀態並輔助數據的分析。

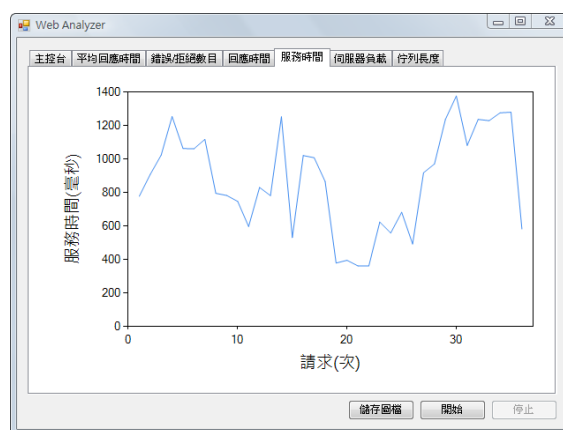


圖 3.2 效能分析工具資訊繪圖功能畫面

3.2 量測方式及數據說明

本軟體將模擬使用者瀏覽網頁的情形，其量測方式及數據如圖 3.3 所示。首先向網站發出 TCP 請求，以三向交握 (Three-Way Handshake) 方式建立連線，隨後發出 HTTP 請求，伺服器端 (Server) 接收 HTTP 請求進行處理，處理完畢之後傳送 HTTP 回應至客戶端 (Client) 的使用者。客戶端發出 HTTP 請求至接收 HTTP 回應的時間定義為該次請求的回應時間。若忽略網路處理的延遲，則回應時間是由兩部分所組成，一部分為網路傳送資料的傳輸時間，另一部分為伺服器處理請求的服務時間。當使用者接收 HTTP 回應之後，經過一段時間之後會再次向伺服器發出 HTTP 請求。客戶端接收 HTTP 回應至下次發出 HTTP 請求的時間定義為思考時間 (Think Time)，其意義是使用者瀏覽網頁所需花費的時間。此思考時間是以指數分佈 (Exponential Distribution) 的機率產生，做為卜瓦松過程 (Poisson Process) [9] [10] 模擬一般網路的使用情況，其思考時間的機率平均值是由軟體輸入參數設定。



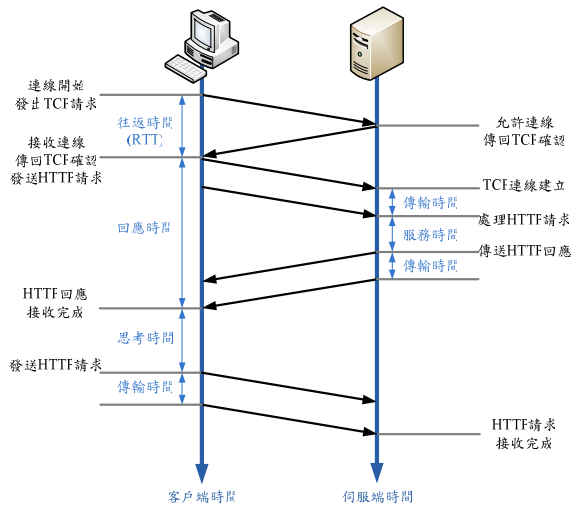


圖 3.3 量測方式與數據定義之示意圖

本軟體發出請求後，倘若未收到回應將產生兩種例外處理：(1)連線發生錯誤，其原因是 TCP 無法成功建立，該量測數據為錯誤總數。(2)請求發生逾時，這是 TCP 已經建立連線之後軟體發出 HTTP 請求，但等候 HTTP 時間過久迫使程式放棄該次請求，本軟體之逾時時間設定為 30 秒，其量測數據為逾時總數。

3.3 程式設計架構

本軟體的設計架構分為主程式與執行緒兩部分，其程式流程如圖 3.4 所示。當使用者按下開始按鈕之後，主程式讀取設定參數以建立執行緒，隨後進入等待狀態直到各執行緒結束為止。執行緒目的為模擬各使用者瀏覽網頁之行為，主程式依據設定參數產生多個執行緒以模擬數個使用者。各執行緒將向指定的 URL 提出同步要求，即執行緒程式會進入等待狀態直到完全接收 HTTP 回應後才繼續執行。HTTP 回應接收完成之後進行相關數據的更新，爾後等待一段隨機產生的思考時間將再次發出 HTTP 請求，該段流程將持續執行直到限制條件符合為止才結束執行緒。

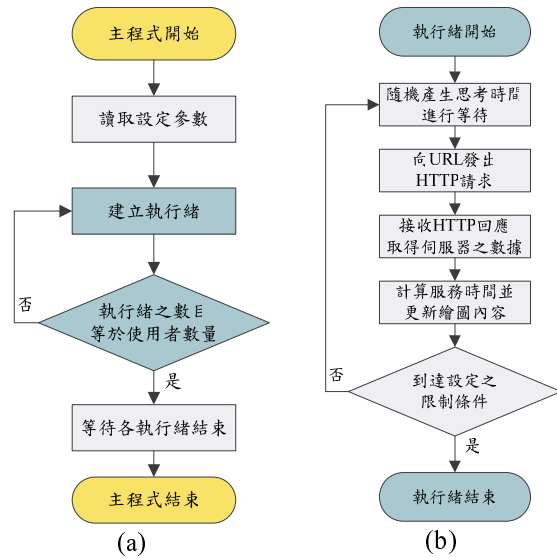


圖 3.4 效能分析工具之程式流程

4. 系統實測

4.1 實驗平台

本研究之實驗平台架構如圖 4.1 所示，三部伺服器建置成為三層式架構，提供多層式網頁應用程式執行之平台，伺服器之間以超高速乙太網路(Gigabit Ethernet, GbE)連接，可將資料傳輸時間減至最低。另外，使用一台筆記型電腦安裝效能分析工具軟體，模擬使用者使用網際網路的情形。

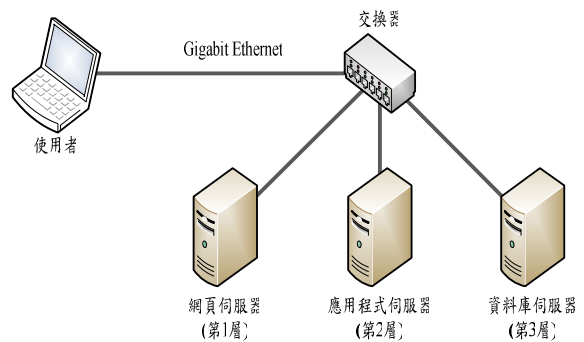


圖 4.1 實驗平台架構



系統之詳細規格如表 4.1 所示，第一層使用 Apache 網頁伺服器軟體，安裝本論文所開發的允入控制模組，此模組監測系統的負載情形以調整請求之許可與否。第二層安裝 Zend Server 應用伺服器軟體，做為執行 PHP 網頁應用程式之平台。第三層則安裝 MySQL 資料庫管理系統，

提供網頁應用程式的資料儲存功能。選用這些伺服器軟體除了符合原有網頁應用程式的執行要求規格之外，並具有下列性質：(1)屬於開放原始碼並且免費使用。(2)效能表現佳。(3)個人網站至大型商業網站皆廣泛使用，具有一定可信度。

表 4.1 測試平台規格

	第一層 網頁伺服器	第二層 應用程式伺服器	第三層 資料庫伺服器
型號	IBM xSeries 206m	IBM System x3250	IBM System x3455
作業系統	Microsoft Windows Server 2003 SP2	Microsoft Windows XP SP3	Microsoft Windows Server 2008
處理器	Intel Pentium 4, 3.4GHz	Intel Xeon 3040, 1.86GHz(Dual-Core)	AMD Opteron 2.2GHz(Dual-Core) x2
記憶體	2GB	4GB	2GB
硬碟	250GB	500GB	250GB
伺服器軟體	Apache HTTP Server 2.2	Zend Server CE 4.0	MySQL CE 5.0

此平台上安裝之網頁應用程式為 Moodle 1.9.3 版[11]，Moodle 為一套開放原始碼的內容管理系統(Content Management System, CMS)與學習管理系統(Learning Management System, LMS)。此網頁應用程式不具有過載控制能力，因此當使用者過多時，其服務品質將受到影響，使用本研究之允入控制模組之後，不需修改此網頁應用程式即可以具有過載保護功能。

4.2 實驗條件

各層伺服器軟體的設定皆使用預設值，其中第一層因安裝允入控制模組，有關此模組運作之 Apache 的參數值特別說明如下[12]：

(1) ThreadPerChild：在 Windows 中預設值為 64，表示 Apache 將開啟 64 個執行緒服務使用者的請求，相當於佇列系統中有 64 個服務器。

(2) ListenBacklog：預設值為 511，用做 TCP 聆聽 80 埠最大可保留的連線數量，相當於佇列系統中的佇列容量。

(3) KeepAliveTimeout：預設值為 5，此功能因 Apache 支援 HTTP 1.1 之持續連線(persistent connection)功能，使用者完成資料接收後，伺服器可以保留 TCP 連線 5 秒，避免 TCP 再次建立連線的時間。此參數將間接影響執行緒和連線的使用情形。

本實驗每次將進行 180 秒，各次實驗之間僅改變其中一個參數進行，以便獲得參數值和結果的關係。

4.3 實測數據

A. 負載測試：

本實驗將各別設定 1 秒、5 秒與 10 秒的思考時間進行量測，其實驗結果分別為表 4.2、表 4.3 與表 4.4，表示該系統在不



同負載之下的反應情形。在表 4.2 之中，當使用者數量高於 60 個以後，可以觀察到逾時情形和請求佇列的發生，並且發生次數隨著使用者而增加。這是因為 Apache 預設為 64 個執行緒，當使用者數量超過 64 個以上，所有的執行緒都處於忙碌狀

態，伺服器將處於滿載工作，因此超過的 HTTP 請求必須進入佇列中等待處理，若等待時間過長時則會發生 HTTP 請求逾時的情形，使用者等候不到伺服器的回應而放棄 HTTP 請求，並且切斷連線重新發出 HTTP 請求。

表 4.2 思考時間=1 秒系統之效能

使用者	請求				平均 回應時間 (ms)	平均 服務時間 (ms)	平均 伺服器 負載(%)	平均 請求佇列 長度
	成功	錯誤	逾時	拒絕				
1	133	0	0	0	350	347	1.6	0
10	854	0	0	0	991	957	13.2	0
20	887	0	0	0	2927	2832	28.7	0
30	882	0	0	0	5044	4860	43.5	0
40	889	0	0	0	7027	6827	58.5	0
50	892	0	0	0	9019	8713	72.2	0
60	889	0	0	0	11179	10753	87.1	2
70	890	0	22	0	12429	11827	95.3	4
80	807	0	115	0	12566	12294	97.1	10
90	861	0	158	0	12430	12007	95.2	23
100	856	0	218	0	12596	11920	93.4	33

表 4.3 思考時間=5 秒系統之效能

使用者	請求				平均 回應時間 (ms)	平均 服務時間 (ms)	平均 伺服器 負載(%)	平均 請求佇列 長度
	成功	錯誤	逾時	拒絕				
1	29	0	0	0	363	359	1.6	0
10	310	0	0	0	415	408	4.6	0
20	643	0	0	0	647	608	10.4	0
30	866	0	0	0	1256	1199	17.5	0
40	893	0	0	0	3265	3131	33.1	0
50	899	0	0	0	4857	4713	44.5	0
60	897	0	0	0	7158	6945	63.4	0
70	912	0	0	0	8875	8410	73.9	1
80	886	0	1	0	11260	10650	89.3	2
90	872	0	38	0	12329	11242	93.3	6
100	794	0	124	0	12566	11092	90.9	14



表 4.4 思考時間=10 秒系統之效能

使用者	請求				平均 回應時間 (ms)	平均 服務時間 (ms)	平均 伺服器 負載(%)	平均 請求佇列 長度
	成功	錯誤	逾時	拒絕				
1	17	0	0	0	370	366	1.6	0
10	179	0	0	0	360	357	3.1	0
20	352	0	0	0	428	411	5.3	0
30	519	0	0	0	508	489	7.0	0
40	758	0	0	0	881	828	13.1	0
50	794	0	0	0	939	873	14.5	0
60	914	0	0	0	2149	2011	25.4	0
70	908	0	0	0	4291	4156	41.4	0
80	929	0	0	0	6230	6011	55.3	0
90	928	0	0	0	7699	7406	66.4	0
100	932	0	0	0	9931	9538	83.3	0

當思考時間增加時，由於請求率減少造成負載降低，故使用者數量必須較大時才具有發生逾時情形和請求佇列的負載條件。表 4.3 為思考時間等於 5 秒的情形，使用者數量必須高於 80 個左右才使得伺服器滿載工作。而思考時間等於 10 秒的情形之下，使用者數量為 100 個以內皆不會使伺服器滿載，因此表 4.4 之逾時情形和請求佇列的發生次數皆為零。平均回應時間與使用者數量的關係如圖 4.3 所示，當使用人數增加時，其回應時間將增加，直到系統滿載為止。以思考時間設定 1 秒為例，系統在使用者數量超過 70 個之後，平均回應時間不再繼續增加，這是因為系統已經滿載，請求將等候於佇列之中，此情形同樣發生於思考時間為 5 秒且使用者數量超過 90 個的情況。由此可知，該系統於滿載的平均回應時間約為 12.5 秒，注意此平均回應時間是以成功的 HTTP 回應而計算求得，發生逾時請求的時間並未列入計算。

圖 4.4 為回應時間與伺服器負載的關係，隨著伺服器負載的增加，其回應時間也將增長，由圖中可得知回應時間僅與伺服器負載有關，而與思考時間無關。

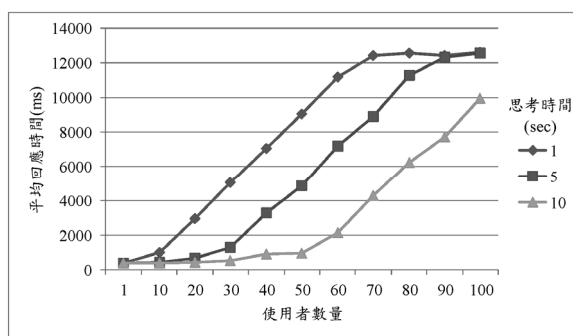


圖 4.3 回應時間與使用者的關係

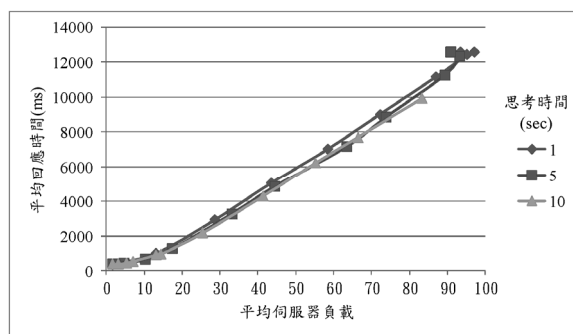


圖 4.4 回應時間與伺服器負載的關係

5. 結論

網際網路與世界的複雜性造成網站容量規劃的困難。瀏覽人數突然增加造成網站過載，其結果將造成服務時間過長、請求逾時或連線失敗等問題，網站性能分析成為重要的問題。本論文為研究多層式網



頁應用程式之性能分析，因此開發客戶端的工具軟體。此客戶端的工具軟體之功用為分析網站的效能表現，對於該軟體提供下列研究建議：

- (1) 自動化負載測試
避免使用手動調整參數與紀錄結果，減少花費的時間與人力。
- (2) 加入暫態測試的功能
實驗進行當中動態調整負載參數，可得知系統對負載的暫態響應。
- (3) 加入模擬使用者行為的功能
藉由設定請求的網址清單以模擬使用者瀏覽網站的行為，可使實驗結果趨於實際發生的情形。

參考文獻

1. J. F. Jurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet (Third Edition)*, New York: Addison-Wesley, 2005, pp. 56-58.
2. What Is Web 2.0 - O'Reilly Media, <http://oreilly.com/web2/archive/what-is-web-20.html>, June 2009.
3. J. Farrell and G. S. Nezelek, "Rich Internet Applications The Next Stage of Application Development," *Proceedings of the 29th International Conference on Information Technology Interfaces*, Cavtat, Croatia, 2007, pp. 413-418.
4. J. Cao, M. Andersson, C. Nyberg and M. Kihl, "Web Server Performance Modeling Using an M/G/1/K*PS Queue," *Proceedings of the 10th International Conference on Telecommunications*, Papeete, Tahiti, 2003, pp. 1501-1506.
5. X. Liu, J. Heo and L. Sha, "Modeling 3-Tiered Web Applications," *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Atlanta, Georgia, pp. 307-310.
6. D. Gross, J. F. Shortle, J. M. Thompson and C. M. Harris, *Fundamentals of Queueing Theory (Fourth Edition)*, New Jersey: John Wiley & Sons, 2008.
7. D. A. Menasce and V. A. F. Almeida, *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*, New Jersey: Prentice Hall PTR, 2000, p. 118.
8. D. A. Menasce and V. A. F. Almeida, *Capacity Planning for Web Services: Metrics, Models, and Methods*, New Jersey: Prentice Hall PTR, 2001, pp. 158-159.
9. K. H. Yeung and C. W. Szeto, "On the Modeling of WWW Request Arrivals," *Proceedings of the 1999 International Workshops on Parallel Processing*, Aizu-Wakamatsu City, Japan, 1999, pp. 248-253.
10. Y. Bholel and A. Popescu1, "Measurement and Analysis of HTTP Traffic," *Network and Systems Management*, vol. 13, no. 4, 2005, pp. 357-371.
11. Moodle.org, <http://moodle.org/>, June 2009.
12. 郭文生譯，*Apache 技術手冊(第三版)*，台北：歐萊禮，2003。

