

私立東海大學
資訊工程與科學所

碩士論文

網站系統品質分析之度量研究

A study on quality measurement of Web-based
system

指導教授：張文貴 博士

研究生：洪昕凱

中華民國九十一年七月

謝 誌

首先感謝恩師張文貴這三年來在學業及待人處事上的指導與關懷，讓我在研究所的修業期間課業上有所增進，在人際關係的上更能學會如何處事圓融與應對進退。

撰寫論文期間，感謝恩師不辭勞苦，逐字檢閱指導完成，並給予我在論文寫作上諸多的指導與建議；研究所這段期間，感謝林助興老師、許文斌老師與鄭國揚老師在學業上之教導及啟發，以及我所有的好朋友在我遇到挫折時給予鼓勵，讓我能繼續的完成學業，感謝同窗好友閔翔在學業與生活上能夠與我並進努力，使得一切能夠順利進行。

最後感謝我的父母親，讓我在學期間能夠專心於課業上的研讀、在我困苦的時候給予我支持與鼓勵。謹以此論文獻給我最重要的家人與在東海的這段期間幫忙我的人，也祝他們一切都能夠順利成功。

謹誌於

東海大學資訊工程與科學所

中華民國九十一年七月

摘 要

由於網際網路的應用日趨普遍，其正確性與軟體品質保證的重要性亦相對遽增，本研究的方向在於發展網站可靠度之驗證機制與網站負載量的評估關係式，可用以驗證與評估網站超鏈結可靠度，另一方面也可分析網站在極短的時間內能夠同時處理多少需求的能量問題。

本文先針對網站品質進行分析探討，而後以現有超鏈結分析方法進行研究，為彌補現有超鏈結分析方法之不足，本研究提出統計使用測試的方法，用以評估網站超鏈結可靠度，最後並舉一實際範例說明此驗證機制之應用，包含使用模式之建構與分析。本研究提出之驗證機制的優點在於整個超鏈結評估流程的完整性、評估流程之自動化設計、網站使用模式的快速建立與準確的超鏈結可靠度評估計算。

在網站效能測試上，使用原先用於評估等待線理論之 Little 關係式加以修改並應用至網際網路環境，配合網際網路之特性用以評估網站之效能分析，最後獲得網站之負載量等效能評估數據，這些數據可以提供網路管理相關人員為參考，並能根據各企業組織的不同需求調整連線逾時時間數據以使網站達到效能之最佳化目的。

關鍵詞：軟體品質保證、網站超鏈結可靠度、統計使用測試、網站使用模式、等待線理論、Little 關係式

Abstract

As the application of Internet gets more and more popular nowadays, its validation and software quality assurance become more and more important at the same time. The purpose of this paper is to develop a Web site reliability certification mechanism, which is used to verify and evaluate hyperlinks of a Website. Another purpose is to evaluate that how many requests a Web server can service in the short time.

After we investigate the issues related to Website quality, we study the methods for hyperlink analysis. To reinforce the present methods for hyperlink analysis, we purpose using statistical usage testing method to evaluate the reliability of the Website hyperlinks. Finally, we demonstrate a real example to explain the Website reliability certification mechanism, including usage model building and analysis. The advantages of this mechanism are the complete and automatic hyperlink evaluation process, rapidly building Website usage modeling, and accurate prediction of hyperlink reliability.

In addition, we revise the Little's formula which is used to evaluated queue model and apply it in Internet. We refer the characteristic of Internet and use the revised Little's formula to evaluate performance of Website. We can provide the result of evaluation for network manager, who can base on requirement of company and adjust server connect time to reach the Website performance optimization.

Keyword :Software Quality Assurance, hyperlink reliability, Statistics usage testing, Website usage model, queuing theory, Little's formula

目 錄

謝誌.....	i
摘要.....	ii
Abstract.....	iii
目錄.....	iv
圖目錄.....	viii
表目錄.....	ix
中英文專有名詞對照表.....	x
符號對照表.....	xii
第一章、 緒論.....	1
1、 研究背景與動機.....	1
1.1. 研究背景.....	1
1.2. 動機.....	1
2、 論文架構.....	3
3、 預期成果.....	4
第二章、 網站品質分析技術.....	5
1、 網站系統之普遍性.....	5
2、 網站品質度量因子.....	5
3、 網站品質分析評估的挑戰與困難.....	6
第三章、 網站超鏈結可靠度之驗證.....	8

1、	網站超鏈結品質.....	8
1.1.	超鏈結特性.....	8
1.2.	現有超鏈結分析方法探討.....	9
2、	網站超鏈結可靠度.....	10
2.1.	軟體失效.....	10
2.2.	軟體可靠度.....	12
2.3.	超鏈結可靠度.....	14
3、	網站超鏈結可靠度之驗證機制.....	16
3.1.	統計使用測試.....	16
3.2.	馬可夫架構模式.....	18
3.3.	軟體使用模式.....	18
3.4.	使用鏈與測試鏈.....	21
3.5.	超鏈結可靠度計算說明.....	21
3.6.	驗證機制.....	23
3.7.	評估流程.....	24
3.8.	系統架構.....	26
3.9.	比較分析.....	29
4、	網站超鏈結可靠度計算之舉例說明.....	30
4.1.	應用範例介紹.....	30
4.2.	建構使用模式.....	31
4.3.	使用模式分析.....	33

4.4. 系統評估	33
第四章、 利用等待線理論分析網站負載量	37
1、 網站效能評估	37
1.1. 網站性能評估	37
1.2. 通訊協定	38
1.3. 網站效能測試	39
2、 等待線理論	42
2.1. 等待線模型	42
2.2. 需求到達率	44
3、 網站系統之負載量分析	44
3.1. 反應時間	44
3.2. 服務率	45
3.3. Little 關係式	45
3.4. 修正後 Little 關係式	48
3.5. 比較分析	51
4、 網站系統反應時間之實例介紹	51
4.1. 應用實例介紹	52
4.2. 資料蒐集	52
4.3. 評估結果	53
第五章、 結論與未來研究方向	55
1、 結論	55

2、 未來工作	56
參考文獻	57

圖 目 錄

圖 1、引起系統失效之情況	12
圖 2、軟體可靠度驗證流程	13
圖 3、超鏈結正常與失效狀態.....	16
圖 4、基於使用模式的統計使用測試	17
圖 5、簡單之網站使用模式範例.....	19
圖 6、超鏈結可靠度計算範例.....	22
圖 7、網站超鏈結驗證系統架構 - DR. HYPERLINK	24
圖 8、完整使用模式架構圖	25
圖 9、系統自動測試架構圖	27
圖 10、AUTOFEA 網站之首頁.....	31
圖 11、AUTOFEA 網站超鏈結架構圖.....	32
圖 12、A 狀態點之放大圖	32
圖 13、B 狀態點之放大圖	33
圖 14、分佈於 507 個測試案例中的可靠度曲線.....	36
圖 15、基本等待線模型	43
圖 16、網站反應時間示意圖	45
圖 18、需求全部等待時間 W_Q 表示圖.....	48
圖 19、3DSPOTLIGHT 網站	52
圖 20、3DSPOTLIGHT 網站最後評估結果	54

表 目 錄

表 1、軟體失效之類別.....	11
表 2、超鏈結訊息種類.....	15
表 3、評估摘要.....	26
表 4、網站品質之評估工具比較.....	30
表 5、 AUTOFEA 使用模式分析報告表.....	33
表 6、 AUTOFEA 網站之測試案例之一.....	34
表 7、 AUTOFEA 網站之失效分析報告表.....	34
表 8、 AUTOFEA 網站超鏈結之評估結果.....	35
表 8、關係式 1、2 與 3 之符號表.....	46
表 9、等待線符號表.....	49
表 10、分析比較表.....	51
表 11、資料蒐集與設定.....	53
表 12、不同需求數下的服務率(%) 與需求等待時間.....	53

中英文專有名詞對照表

英文	中文
<i>Arcs</i>	連線
<i>Arc coverage</i>	連線涵蓋率
<i>Balking</i>	止步
<i>Connect time</i>	連線逾時時間
<i>Complexity metric</i>	複雜度因子
<i>Error seeding model</i>	植錯型模式
<i>Fault avoidance</i>	缺陷預防
<i>Fault detection and removal</i>	缺陷偵測與移除
<i>Fault tolerance</i>	缺陷容忍
<i>Failure rate model</i>	失效率模式
<i>First come first served</i>	先進先出排序
<i>Hyperlink</i>	超鏈結
<i>Hypertext</i>	超文件
<i>Hyperlink reliability</i>	網站超鏈結可靠度
<i>Homepage</i>	首頁
<i>Idle waiting time</i>	無效等待時間
<i>Internet service</i>	網路服務
<i>Jockeying</i>	轉換
<i>Load test</i>	負載量測試
<i>Last come first served</i>	後進先出排序
<i>log</i>	網站伺服器紀錄檔
<i>Level</i>	使用模式的層級
<i>Markov process</i>	馬可夫過程
<i>Markov chain</i>	馬可夫鏈
<i>Mean First Passage</i>	平均第一通道
<i>Probability of Occurrence</i>	發生機率

<i>Queue</i>	等待線
<i>Queuing theory</i>	等待線理論
<i>Response time test</i>	反應時間測試
<i>Reliability growth model</i>	可靠度成長模式
<i>Response status code</i>	反應狀態碼
<i>Response time of Website</i>	網站反應時間
<i>Reneging</i>	主動離開
<i>Stress test</i>	壓力測試
<i>System failure</i>	系統失效
<i>System error</i>	系統錯誤
<i>System fault</i>	系統缺陷
<i>Software metric</i>	軟體因子
<i>Statistics usage testing</i>	統計使用測試
<i>States</i>	狀態點
<i>Software Not Invoked</i>	狀態起點
<i>Software Terminated</i>	狀態終點
<i>State coverage</i>	狀態點涵蓋率
<i>Smoke test</i>	前測測試
<i>Service rate of Websites</i>	網站服務率
<i>Transition probability</i>	轉換機率
<i>Tunneling</i>	隧穿運作
<i>Usage state</i>	使用狀態
<i>WWW</i>	網際網路
<i>Web page</i>	網頁

符號對照表

符號	符號意義
a	需求進入系統時間起點
b	需求離開系統時間終點
c	網站連線逾時時間
h_{s_j}	期望順序數目
Lq(t)	(t)- (t)
Lq	服務需求數
Ln	未被服務之需求數目
n	被服務之需求數目
N	全部需求數目
$p_{i,j}$	狀態點j k的機率
\hat{P}	使用鏈中所有的預估使用機率
R	超鏈結可靠度
$R_{in,term}$	R為可靠度, in 為初始狀態, term 為結束狀態
\bar{R}_s	被服務需求之平均網站反應時間
Rt	被服務需求之總網站反應時間
S_i	使用狀態
t1	第一個需求到達等待線的時間(秒)
t2	最後一個需求到達等待線的時間(秒)
U_s	馬可夫鏈
Wq(i)	第i個需求在等待線中的時間
Wq	需求在等待線中之等待時間

$W_q(i)$	第 <i>i</i> 個需求在等待線中的時間(秒)
	轉換的狀態集合
(<i>t</i>)	<i>t</i> 時間進入系統的需求數目
(<i>t</i>)	<i>t</i> 時間離開系統的需求數目
	需求到達率

第一章、緒論

1、研究背景與動機

茲將研究背景與動機分別說明如下：

1.1. 研究背景

網際網路的快速發展造就了網站的大量興起，連帶人們的生活周遭事物也跟著 e 化，隨著生活網路化，電子商務的腳步也跟隨著網路邁進，各個公司企業的網站紛紛成立以搶食網路這塊廣大市場。然而網站的紛紛建立，隨之而來的是品質的參差不齊，連線時間過久、網站內容出錯都會造成使用者的大量流失，在網路的戰國時代，網站品質成了市場上決勝的關鍵。

本研究是以軟體使用測試為基礎，用於網站之品質評估分析，著重於驗證機制之完整性，建構相對應之網站使用模式，根據測試計畫隨機產生期望的測試案例個數，在相同的測試環境與條件下執行，最後蒐集失效狀態與紀錄，推算網站超鏈結可靠度，提供網站建構者與網管人員為參考依據，以使網站的效能在不變更軟硬體下達到品質最佳化。

本論文共有六章，第一章說明研究背景與動機、目的方法與預期成果；第二章則為相關的文獻探討，包含網站系統之普遍性、網站品質度量因子與網站品質分析評估的挑戰與困難；第三章敘述網站超鏈結可靠度之驗證機制；第四章則說明如何利用等待線理論(Queuing theory)分析網站負載量；在第五章，則是結論與未來研究工作方向。

1.2. 動機

二十一世紀是資訊、科技、快速與便利的世紀，自從 90 年代以來，網際網路的快速發展對於現今世界造成革命性的衝擊，網際網路的出現，也讓資訊傳播邁向個人化時代，只要擁有網路空間與電腦設備，個人也可以

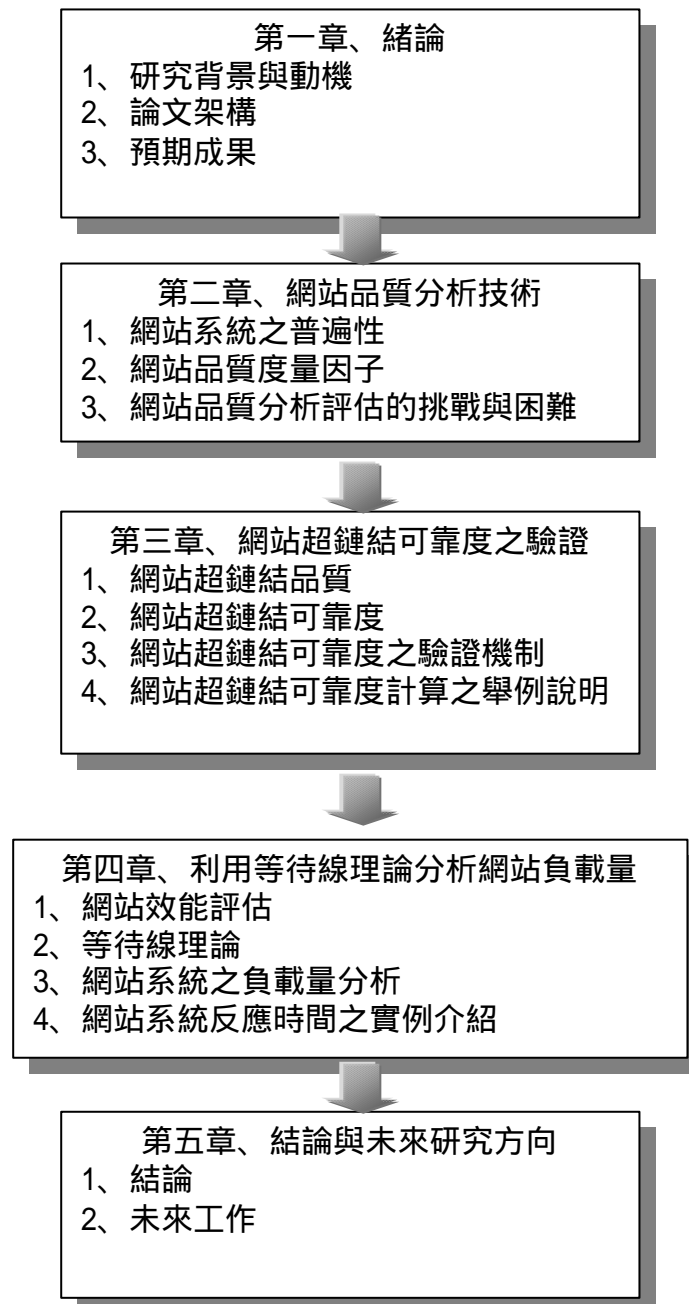
透過網路，將自己的創作與意見向全世界發表。

網路消費人口不斷的增加，凡舉線上電子交易、電子商務等也跟著孕育而生，企業公司面對網路技術快速發展的時刻，網路系統的品質日趨重要，如何針對顧客需求建立高品質的網路系統，儼然已成為今日企業用戶最關心的議題之一。

然而網路在跨越二十一世紀之前發生泡沫化的現象，許多分析師向下修正網路的前景、獲利預估，許多經營不善的公司網站紛紛倒閉，許多人開始懷疑網路是否能像當初興起時擁有蓬勃的能量，但網路泡沫化的現象僅是一時的，事實上，企業網路化的腳步仍舊持續增加，雖然網路泡沫化對於各個網路公司所造成的衝擊不小，網路泡沫化未嘗不是一件好事，可以對如雨後春筍般建立的網站進行體質上的篩檢，透過這一波市場嚴格的考驗，許多經營不善，產品不具競爭性，缺乏前瞻性策略與管理方針的網站遭到淘汰，而所遺留下的市場空缺由體質較佳的網路公司所取代，這一方面對消費者是有利的，消費者因此可獲得更多滿意的服務。

以經營的角度來看，不論是那一個市場，需在市場定位或取得消費者的信心，才可能長久經營，不管提供的服務是內容、交易平台、某種產品或是軟體之解決方案，都必須要能夠有長期的獲利模式，相對來講網站的品質變成市場競爭的絕對關鍵，因此網站品質的評估研究成為本文研究的重點與研究動機。

2、論文架構



3、預期成果

本研究依據研究目的，透過前述之研究方法與研究架構，預期完成之工作項目與成果如下：

1. 超鏈結評估流程的完整性、評估流程之自動化、網站使用模式的建立與超鏈結可靠度的評估計算。
2. 應用並修改 Little 關係式於網站效能分析，以供各公司企業主管與網管人員做為參考修正使其獲得更高之網站品質表現。
3. 提供網站品質評估驗證流程的探討，俾利國內外業界與學術界之參考與應用。

第二章、網站品質分析技術

1、網站系統之普遍性

網際網路快速興起後，在短短數年間已成為另一個獨立且擁有龐大資源的媒體，而網路所具有的即時性與超連結等獨特性，也為其在電視、報紙、廣播傳統三大媒體之外博得「第四媒體」的美稱。

網際網路的普遍性根據美國 Computer Industry Almanac 調查報告指出 [1]，2000 年全球網路使用人口，第一名為網際網路的發源地美國高居世界第一，美國上網人口為 1 億 3,500 萬，第二名則是日本，上網人口為 3,400 萬人；若以網路交易金額(含 B2B、B2C)來看，去年美國的花費達 1,425 億美元，遠高於排名第二的西歐國家。若以上網人口比例來看，北歐國家自 1992 年以後就居全球第一，美國去年排名第六，網路族佔總人口的比例為 47.8%。

隨著後期電子商務的大量興起，使得傳統產業受到前所未有的衝擊，電子化變成主要的潮流，網站變的與人們的生活息息相關，各項消費凡舉購物、理財等皆可透過網站系統的使用來完成，網路的大量使用開啟了人們另一個資訊溝通管道，也宣告了另一個新世代的開始。

2、網站品質度量因子

網站品質之評估量度因子可以分成許多類，這些品質量度因子皆用來測試網站的相關性能表現，測試前首先需要考慮網站品質因子的定義、測試者如何實際的測量網站的效能等，一般來說，網站品質測試通常是給予測試者一段時間，在這時間內評估網站，網站品質量度因子可分為下列幾項：壓力測試(Stress test)、超鏈結測試、網站內容測試、反應時間測試(Response time test)與負載量測試(Load test)等。這些測試的結果可以加以分析，若任何缺陷被找到，相關設計人員可以進行改善。

3、網站品質分析評估的挑戰與困難

網站品質分析評估之重要過程著重整個測試流程之設計，測試的困難在於伴隨著電腦科技的進步，網站系統本身的複雜性日與遽增，而網站之測試需要考慮到當時使用者端與伺服器間的測試環境，且需在真實的環境下進行測試，如此所得到的測試結果才較為精確。另一方面，網站系統的測試若需測試使用伺服器所建之資料庫，在測試前有一些基本的假設。因此，進行網站測試時，有幾項主要的困難如下：

(1) 釐定網站品質因子

網站測試的首要步驟便是釐定網站之品質因子，包括網頁內容之一致性與正確性、反應時間、可靠度等，這些品質因子在測試前就需釐定清楚，以利後期整個網站測試結果的準確。

(2) 網站的複雜度

網站由 HTML(Hyper Text Markup Language)的頁面所組成，使用者透過瀏覽器可以看到 HTML 所呈現的畫面，如文字、圖片、聲音等，瀏覽器在市場上競爭激烈，並且爭相推出功能更強大的新版軟體，以滿足使用者的需求，而這些新的功能加入使得測試的工作更加困難，測試結果的準確度也不容易掌握，另一方面，網站使用新的技術，如 Java、JavaScript 與配合資料庫的使用都會使網站測試面臨更多的挑戰。

(3) 網站驗證機制的建立

驗證機制的建立有助於測試結果的準確性，驗證機制的流程包含從開始的網站品質因子釐定、網站規格分析、測試案例的產生與執行與最後的評估報告，每個階段都需嚴謹的測試步驟，網站規格分析階段，需詳細瞭解該網站之原始規格，以利後期執行測試案例的對照，測試案例的產生與執行方面，面臨最大的挑戰在於測試案例的自動執行，因網站測試元件眾多造成產生大量的測試案例，這些測試案例若由人工一一執行將造成測試資源的浪費，因此設計測試案例的自動執行步驟是此階段的工作重點，網

站的評估報告方面，最後的評估結果與前段的計算方法與原理有關，測試理論越完整則最後的評估結果也越強而有力。

第三章、網站超鏈結可靠度之驗證

1、網站超鏈結品質

探討網站超鏈結可靠度，需先從網站超鏈結的特性來看，第三章共有四個部分，第一部份依其網站特性對現有超鏈結分析方法進行探討，包括現行之網站分析工具之介紹，第二部分，探討網站超鏈結之可靠度之定義，第三部份描述網站超鏈結可靠度之驗證機制，包括系統架構與整個評估流程，最後，用此驗證機制與其他網站分析工具進行比較，第四部份，將一個實際的範例應用在此驗證機制上加以說明，包含使用模式建構與分析與最後的評估結果。

1.1. 超鏈結特性

網際網路的出現，使得人與人之間的距離大大的縮短，網路上最普遍應用之一，便是以 HTML 為架構的網際網路 WWW (World Wide Web)。

網際網路被使用者高度的接受，其最主要的原因是瀏覽器(Browser)的大量使用，因瀏覽器具有強大的影音聲光效果與互動能力，透過超鏈結(Hyperlink)功能，讓使用者擷取想要獲得的資料、影像、聲音等，因此超鏈結被大量的使用在網際網路上。

超鏈結是線上超文件(Hypertext)鏈結方式，游標所指之處，會直接出現鏈結指向的位址，一般而言，只需點選超鏈結，就可以執行連結動作，透過使用 HTML 語言編寫的超文字文件和超鏈結服務，再加上和 Internet 上其他協定與系統，如 HTTP(Hyper Text Transport Protocol)、FTP (File Transfer Protocol)、POP3 (Post Office Protocol Version)等技術協定的整合，無論文字，聲音，圖片，影像都可以整合成完善的介面，提供使用者方便而且多媒體化的資訊，而超鏈結在 HTML 中的格式如下：

```
<a href="URL">超鏈結本文文字</a>
```

透過上述之格式，超鏈結有效的將成千上萬的網頁(Web page)進行連結，使得網際網路如同蜘蛛網般綿密，資訊也傳遞更快。

1.2. 現有超鏈結分析方法探討

通常超鏈結分析方法均透過測試分析工具來表現，而目前測試分析工具所使用的方法可以分為三類，分別是分析網站伺服器之紀錄檔、直接掃描超鏈結與掃描已瀏覽過之超鏈結。

1.2.1 分析網站伺服器之紀錄檔進行

分析網站超鏈結是否出現失效之情形，其中最節省本方法是利用現有網站伺服器所產生之紀錄檔進行分析，網站伺服器紀錄檔中紀錄使用者所有點選超鏈結的動作，若有超鏈結失效的情形，則在紀錄檔中也會記錄，分析其紀錄檔案內容便可知到哪些超鏈結失效。

分析網站伺服器紀錄檔(log)的工具如“Web Site Visitor Log”，“Web Site Visitor Log”是針對 Win95 / NT 4.0 平台所設計用來對網站紀錄檔進行分析的工具。“Web Site Visitor Log”可以接受網站的 CLF (Common Log Format)與 ECLF (Extended Common Log Format)紀錄檔[1]，藉由分析這些檔案，“Web Site Visitor Log”可以顯示整個月使用者使用該網站的活動情形。如果該網站某一超鏈結出現錯誤，透過“Web Site Visitor Log”分析網站的紀錄檔，可以知道那個超鏈結出錯，且知道有多少使用者點選此錯誤之超鏈結，但這種分析網站伺服器紀錄檔的方法也有一些缺點，例如網站使用者人數太少，造成某一失效之超鏈結沒有被點選到，從紀錄檔中便無法得知此超鏈結出現失效。因此需要大量的使用者使用該網站，相當於大量的測試案例，才能分析該網站有多少缺陷。

1.2.2 直接掃描超鏈結

掃描超鏈結是最直接的方法，由於網路上的超鏈結環環相扣，通常這

類工具的分析需使用者輸入所要測試的超鏈結數目，如 Linkbot 超鏈結分析工具[2]，Linkbot 是由 Watchfire 公司所發展，Linkbot 可以讓網站發展者在網站正式連線營業前分析並使其網頁最佳化，它可掃描測試的網站中至多 1000 個超鏈結，另外 Linkbot 也提供 HTML 的編輯器協助設計者在找到網站超鏈結錯誤時，可以迅速修復這些錯誤，以確保網站超鏈結的品質。Linkbot 總共有五大功能：基本維護、網站架構組織、修復、自動執行與測試報告，但在此法中要掃描至多少超鏈結才算足夠是一大問題。

1.2.3 掃描所瀏覽之超鏈結

根據測試者所瀏覽過的路徑再進行掃描的工具具有 Astra 與 e-TESTER，Astra 是由 Mercury Interactive 公司開發，Mercury Interactive 公司本身開發一系列之測試產品，也有進行替公司企業進行效能測試的管理等，以利維護公司企業所有的高效能應用軟體，[3]。而 e-TESTER 測試分析工具是由 Empirix 公司所開發，Empirix 公司開發相關的網路應用程式以及做語音資料網路測試與監控的應用產品[4]。

Astra 與 e-TESTER 測試分析工具皆能在 Windows 系統下執行，其分析超鏈結的方式是透過測試案例的產生，而測試案例是先記錄測試者的瀏覽超鏈結的動作，依此測試案例進行分析。此方法之缺點在於測試案例之產生需藉由測試者的瀏覽動作，因此測試案例之產生需要大量的時間，且涵蓋率也難以計算。

2、網站超鏈結可靠度

2.1. 軟體失效

軟體失效對系統的影響，小則系統當機不能執行，大則攸關生命安全的系統，例如核能電廠數位儀控系統或是醫療儀器系統等，軟體失效將會嚴重影響整個系統的任務，甚至造成無法彌補的損害。在整個軟體生命週期中，越晚發現軟體失效，修復此失效所花費的成本則越大，以軟體工程

的角度來看，從軟體設計、編碼、測試、安裝到維修的每一個步驟，都能確實執行且達到要求，軟體相關之文件或使用手冊也都要清楚正確，應用軟體品保監控整個軟體系統的發展流程。

探討軟體可靠度，需先釐清系統失效(System failure)、系統錯誤(System error)與系統缺陷(System fault)之差異[5]，其詳細描述如下表 1。

表 1、軟體失效之類別

名詞	說明
系統失效(System failure)	一個事件發生在某一時間點上，用戶察覺系統無法如預期的服務使用者
系統錯誤(System error)	不符合其原始系統規格之所有系統表現行為
系統缺陷(System fault)	非正確的系統狀態

系統失效是指系統在使用者操作期間發生非預期之事件行為，系統錯誤則是指錯誤的系統行為不符合原始系統規格，系統失效與系統錯誤不同之處在於系統錯誤所涵蓋範圍較系統失效大，系統失效僅止於特定的時間使用者操作下所表現之非預期行為，不同於系統錯誤定義之不符合系統規格的所有系統行為。而系統缺陷所定義之範圍最廣，指所有系統的非正確狀態，包含表現於外的系統錯誤行為與沒有表現出來的系統錯誤狀態。

軟體系統的失效可能會帶來不可預期的損失，因此有三種方法可以用來提高軟體系統之可靠度[5]：

A. 缺陷預防(Fault avoidance)

缺陷預防的技術用於在系統發生缺陷的情形前，將這些缺陷狀態最小化或予以誘導出來。

B. 缺陷偵測與移除(Fault detection and removal)

在系統正式運作前，使用驗證與認證的技術增加錯誤被偵測與移除的機率，這種缺陷偵測與移除技術如系統測試(system testing)與除錯(debugging)。

C. 缺陷容忍(Fault tolerance)

缺陷容忍之目的在於確定系統中的缺陷不會導致系統的錯誤或是系統的錯誤不會導致系統中的缺陷情形。

2.2. 軟體可靠度

軟體可靠度是所有軟體品質因子中重要的因子之一。軟體可靠度的重點在如何估計軟體可能發生異常行為的問題上，亦即是指系統操作時的可靠度，其定義如下[5]：

定義 1：軟體可靠度是指在一個特定操作環境下，某特定時間內不會出現系統失效的機率。

Mills et al. (1987) [5] 發現如果在軟體中移除 60% 的缺陷，則軟體可靠度會提升 3%。大部分的軟體缺陷是由系統錯誤所引起，而且是在軟體產品上市後經過使用者大量的使用後出現，因此若沒有發現這些軟體缺陷，一般使用狀況下，大部分的人還是會認為此系統是可靠的。

軟體可靠度是指在該軟體操作期間發生失效的機率，在輸入軟體系統的輸入集合中，有些輸入值會引起系統錯誤，這些系統錯誤是由系統所產生，如圖 1 之說明，一般說來，軟體可靠度是在一般使用的情況下，依據系統表現出的不符合規格之期望輸出數目機率，故系統中的任何一個失效都可能影響到整個系統的可靠度。

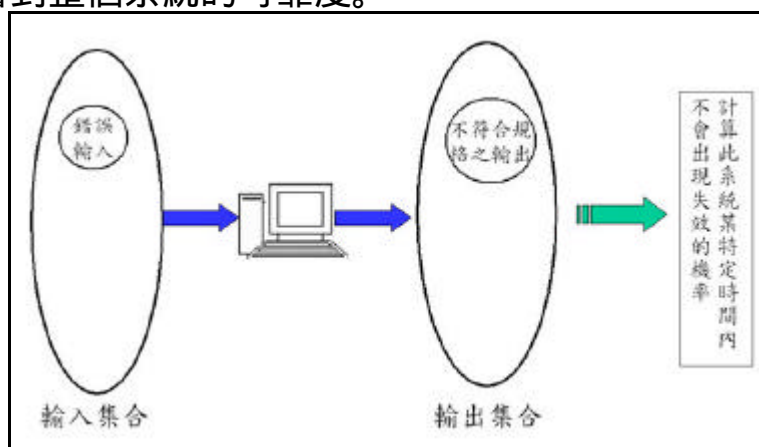


圖 1、引起系統失效之情況

可靠度驗證方面，若要驗證系統是否有依據當初開發之規格測量其系統可靠度，亦需考慮所測量之可靠度是否為使用者所接受。一般可靠度的驗證流程大致可分為系統分析、收集測試資料、產生測試案例與計算可靠度四個階段，如圖 2 之說明：

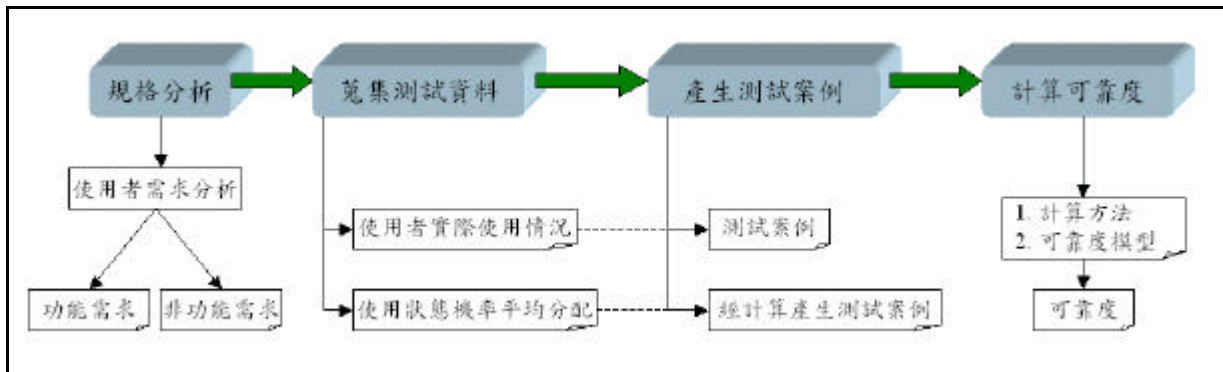


圖 2、軟體可靠度驗證流程

1. 規格分析階段

系統分析階段著重於該系統之原始使用者需求規格，系統本身需依據使用者需求規格進行開發，需求規格分為兩類[6]，一類是功能需求，界定軟體產品如何由輸入經處理轉成輸出，包括軟體功能及定義輸入資料、處理程序、及輸出資料；另一類是非功能需求，用來描述系統軟體的非功能性需求，例如，可維護度與可靠度，可維護度是說明系統軟體交付給使用者後，發展者對於系統往後系統維護度上的要求，可靠度是說明系統軟體的可靠度需求，進而達到整個系統可靠度的需求。

2. 蒐集測試資料階段

蒐集測試資料階段在於蒐集觀察使用者使用系統的狀況，其目的在於計算系統的可靠度，在使用統計測試方法中加入使用者的實際使用系統情況，所計算出的可靠度較為準確，若有一個使用者常用的功能發生失效狀態，則最後計算出的可靠度會大大的降低，反之，該功能不常使用發生失效系統可靠度不會降低太多。

若要節省測試資源，在蒐集測試資料階段可以使用平均分配機率的方法，假設系統中的所有功能使用機率相同，缺點是計算出的可靠度不如實際使用狀況之準確。

3. 產生測試案例階段

測試案例的產生與數學的計算有關，測試案例的產生其重點在於是否有效的涵蓋系統內的所有模組，如果能透過計算產生最少數量且又能涵蓋所有系統狀況的測試案例，執行測試案例時將大量減少測試的時間與資源。

4. 計算可靠度階段

執行測試案例後，若有系統缺陷被找到，此時該系統的可靠度便可以算出，計算軟體可靠度的方法有很多，例如有 Halstead 的軟體因子(Software metric)、McCabe 的複雜度因子(Complexity metric)、植錯型模式(Error seeding model)、失效率模式(Failure rate model)與可靠度成長模式(Reliability growth model)等[7]。

軟體可靠度驗證流程是一套複雜的過程，其重點在於如何測量使用者實際的操作情形，用以對映至驗證流程中，在測試案例方面，如何抽樣產生測試案例以及所產生的測試案例是否能代表整個系統，另外，驗證流程中找出來的錯誤數目多寡如何影響整個系統可靠度。

2.3. 超鏈結可靠度

超鏈結提供上網的使用者從一個大型資料庫方便的找到相關資訊，而且包含逐步的指示與清楚的資料描述，網站本身包含許多網頁，每個網頁藉由以上之超鏈結格式將相關的網頁進行連結，若其中一個超鏈結出現超鏈結錯誤，將會造成使用者使用上的困難，也無法擷取相關資料，本文將探討網站超鏈結的品質，尤其是超鏈結可靠度，相對於軟體可靠度，其基本定義如下：

定義 2：網站超鏈結可靠度(Hyperlink reliability)是指在一個網路環境下，在一特定時間內不會出現 HTTP 錯誤的機率。

顯而易見，網站超鏈結可靠度是軟體可靠度之一種特殊案例，因此統計使用測試仍適用於網站超鏈結之系統。

一般而言，超鏈結的訊息包括 HTTP 之反應狀態碼(Response status code)與描述如表 2 [8]，其中亦包含若干種之 HTTP 錯誤訊息：

表 2、超鏈結訊息種類

反應狀態碼	描述	說明
200	OK	使用者發出之需求被伺服器成功的服務，所傳的代碼是200
301	Moved Permanently	所要存取的資源已經永久的移至其他的URL
302	Moved Temporarily	所要存取的資源暫時的移至其他的URL
303	See Other	所要存取的資源已經移至其他的URL，應由使用者再自行嘗試新的URL，通常這種問題的解決方案是由CGI script將瀏覽器重新導至已經現存的檔
404	Not Found	所要存取的資源不存在於伺服器
500	Server Error	非預期的伺服器錯誤，通常引起此種錯誤的原因是伺服器端的script有語法上的錯誤造成不能正常執行
URL錯誤		超鏈結本身內容與指向的URL不同

舉例來說，HTTP 404 錯誤是一般使用者最常見到的錯誤，此錯誤訊息代表網站伺服器不能回應使用者端發出的需求，換句話說，使用者所想要擷取的檔案不在網站伺服器上；另一個錯誤是超鏈結本身的文字說明與所指向的 URL 不同，例如，網站上有「購物選擇」的超鏈結，實際點選此超鏈結卻連結到其他不相關的網頁。整個超鏈結正常與失效狀態如圖 3 所示：

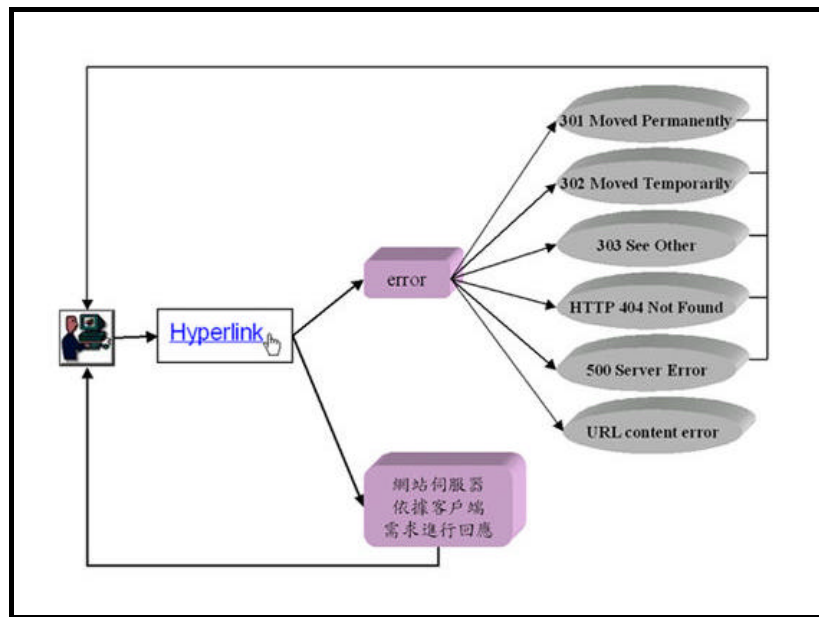


圖 3、超鏈結正常與失效狀態

若網站上的超鏈結出錯，將會大大的減低網站本身的品質，尤其對電子商務的網站而言，使用者會因為網站的服務品質不好而減低到此網站購物的意願，市場的競爭力將大打折扣；因此，如何管理與偵測這些超鏈結的錯誤變成一個重要的課題。

分析驗證超鏈結的正確性與否，可以分為兩大類，一類是網站設計者在設計建置階段，利用建構網站的工具驗證超鏈結的是否符合設計者的要求，另一類是較為常用的方法，使用超鏈結工具進行分析，使用工具驗證通常是網站已經正式連線作業階段，詳細的超鏈結分析驗證工具已在前文中描述過。

3、網站超鏈結可靠度之驗證機制

3.1. 統計使用測試

應用統計的方法於軟體測試是由 Harlan Mills、John Musa 以及分別在 IBM 與 AT & T 公司的同事所發展[9]，使用成熟的軟體工程定律於產品的測試驗證上，以前在工業界中，產品的測試是使用統計的抽樣調查，從總

產品中抽出樣本數，如果這些抽出之樣本數可以滿足驗證的條件，則代表總產品驗收通過。

除了軟體系統的統計使用測試(Statistics usage testing)用於產品與流程品質的測量，統計使用測試近年來已經被修正以及廣泛的應用在軟體品質的評估[10, 11]。觀念上，統計使用測試的基本原理在於最常發生的失效狀況會在早期的測試過程中被發現，所以統計使用測試很適合用在當測試案例母群很多的系統上。

進行統計使用測試時，需先建立使用模式用來代表此系統之所有操作狀態，此時測試案例也從使用模式中用統計的方法抽樣產生。統計的方法抽樣產生測試案例，再藉由測試中的失效資料，最後的測試資訊包括軟體系統之可靠度、平均失效時間等重要之測試評估結果[12]。統計使用測試的優點在於能夠有效的抽樣測試節省測試成本、配合馬可夫鏈的理論計算軟體可靠度等。

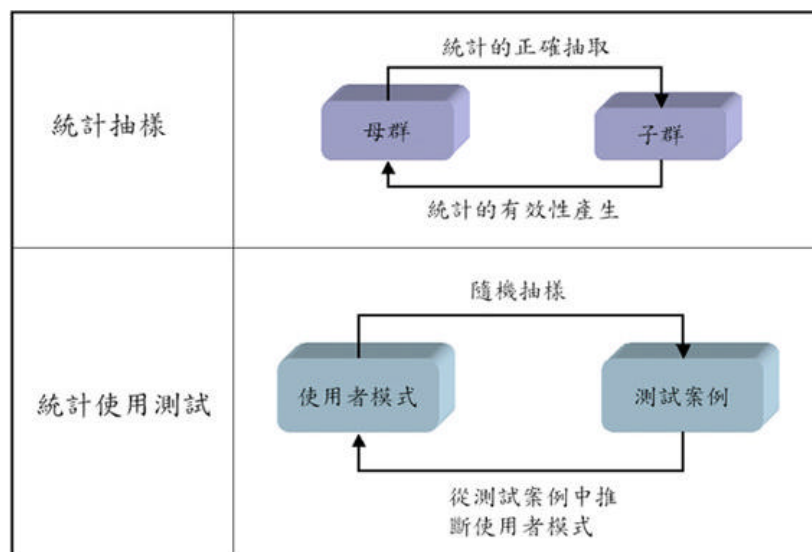


圖 4、基於使用模式的統計使用測試

統計使用測試的優點如下：

- (1) 自動產生測試案例

運用統計使用測試，不論是測試案例的最小覆蓋率或隨機的測試案例，可以透過使用模式自動產生。

(2) 有效的測試

因缺陷會導致系統的失效，只是機率的問題，使用統計測試的優點在於可以設定這些缺陷導致系統失效的機率值，如此有效的測試方法可以使可靠度更為正確。

(3) 可靠度的估算

統計使用測試配合使用模式的建立，在計算過程中加入每個狀態點的權重，可以有效計算出系統使用狀態下的可靠度。

總而言之，軟體之統計使用測試與網站使用模式的建立提供了軟體品質之定量分析，用以進行系統之評估與測試，將此理論充分的運用在網站系統上，使得網站系統之品質能更上一層樓。

3.2. 馬可夫架構模式

在馬可夫過程(Markov process)之中，假設事件未來的系統狀態，只與目前的系統狀態有關，與以前的狀態是沒有關連的，這種假設與實際情況較為相似[13]。運用馬可夫過程之基礎理論，將軟體系統之規格轉化為軟體使用模式，考慮系統內狀態之轉換機率分配，剖析使用模式之相關參數，如狀態或路徑涵蓋率等數據，以執行統計測試，達到開發高效率驗證或高可靠度與低成本軟體系統的目的[14]。

3.3. 軟體使用模式

軟體使用模式是將所有可能的操作情形予以整合後的特徵模式[15]，代表無限的所有使用情形之群體。它可在軟體程式碼撰寫前構建完成，且在建構過程中，帶動軟體規格的改善，增加軟體的可測試度。模式建構的機制是先確認系統的使用狀態(Usage state)，定義為 $s_i(i=0,1,2,n \dots,t)$ ，其中 s_0 與 s_t 分別表示模式中初始狀態與終結狀態，接著描繪狀態之間的連線並

給定其期望使用機率，然後以馬可夫鏈(Markov chain)來表示。

軟體使用模式包含三種主要的元件：狀態點(States)、連線(Arcs)、轉換機率(Transition probability)[16]。應用軟體使用模式於網站的超鏈結測試環境中，網站使用模式則包含了此網站所有的超鏈結之使用狀態與機率。將超鏈結轉化成網站使用模式的狀態點，連線會被標示著兩個數字，前一個數字代表此連線是連結到哪一個狀態點，後一個數字代表此連線到該狀態點的機率，這組數字為轉換機率。如圖 5 中由狀態起點(Software Not Invoked)所連出來的三條連線，分別連結到 A、B 與狀態終點(Software Terminated)三個連結點，因為假設使用者的點選此三點的機率相同，故每個連線的機率為 1/3。而網站系統之超鏈結亦符合一般傳統軟體之特性，將網站超鏈結轉化成軟體使用模式，則每個狀態點為超鏈結，而連線與轉換機率為每個超鏈結所連結出去之數目與機率。

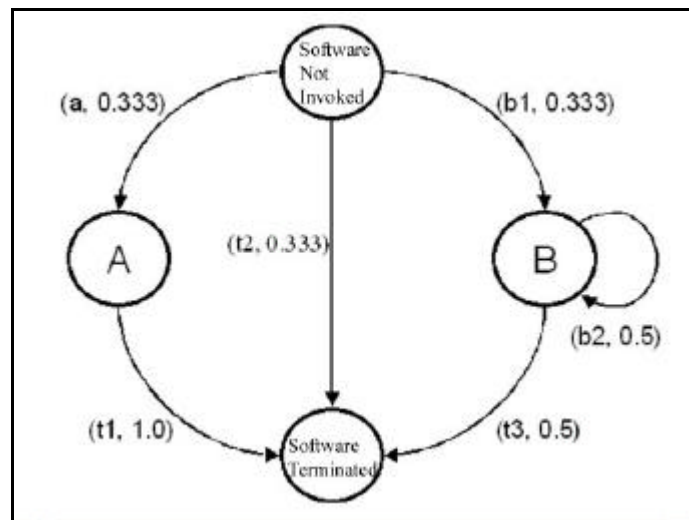


圖 5、簡單之網站使用模式範例

1993 時，Whittaker 證明使用模式可以由馬可夫鏈理論所表示[13]，在馬可夫鏈理論中，每個狀態點是由各個連線所連結，每個連線上個機率代表此一狀態點到下一個狀態點的機率。舉例來說，讓 U_s 為一個馬可夫鏈，對一個吸收鏈與一個初始的狀態點 i ，伴隨著 y_{sij} 機率下一個狀態點為

j ，其發生之單一順序為

$$y_{s_{ij}} = p_{s_{ij}}^a + \sum_{k \in t} p_{s_{ik}}^a y_{s_{kj}} \quad (1)$$

a 是為吸收模式， $P_{s_{ij}}$ 是狀態點 j 的機率， $y_{s_{ij}}$ 為非吸收狀態點直到狀態點 s_j 發生的期望順序數目是：

$$h_{s_j} = \frac{1}{y_{s_{ij}}} \quad (2)$$

從連線 $j-k$ 發生在單一順序下，其機率 $Z_{s_{jk}}$ 為：

$$Z_{s_{jk}} = y_{s_{ij}} p_{s_{jk}} \quad (3)$$

直到連線 $j-k$ 發生期望數目順序 $h_{s_{jk}}$ 是：

$$h_{s_{jk}} = \frac{1}{Z_{s_{jk}}} \quad (4)$$

基於以上的理論，對於所有可能的狀態點與超鏈結覆蓋率測量已經可以進行分析。方程式(2)與(4)可以用來評估馬可夫鏈之狀態點與連線的覆蓋率。

軟體可靠度是從使用模式中隨意抽出路徑的機率，而這些抽出的路徑是不能引起該系統的失效。而這些抽出的路徑就是測試案例，一條至少包含起始點與最終點的完整路徑就是一個測試案例，由馬可夫鏈所計算出的測試案例需基於下列三個事實：

1. 現有的狀態點的發生機率由上一個狀態點所影響
2. 所有的使用狀態是不相容
3. 每個單一的機率進行總加後等於 1

根據馬可夫理論，馬可夫使用模式產生之測試案例需包含所有的狀態點與連線。因此，如果某個狀態點造成該測試案例執行失敗，馬可夫理論可以算出產生系統失效之涵蓋率，在測試的過程中這些涵蓋率的數字可以用來決定測試是否需要終止。

3.4. 使用鏈與測試鏈

利用軟體使用測試之方法進行測試時，在規格分析階段，將先產生一個所有使用案例之母群，稱之為使用鏈，它代表此軟體所有可能的操作情形之集合，在使用鏈中，由於使用狀態間的轉換機率是經由資料分析或實際觀察等方式所取得，稱之為期望使用機率， $p_{i,j}$ 代表狀態 s_i 至狀態 s_j 的期望使用機率；而在實際測試階段所產生的鏈，包含所有已測試路徑的集合，則稱之為測試鏈，其主要目的則是用來追蹤觀察實際每個狀態的執行成功與否[15]。在測試鏈中，由於使用狀態間的轉換機率是經由不斷測試累積所取得，稱之為預估使用機率， $\hat{p}_{i,j}$ 代表狀態 s_i 至狀態 s_j 的預估使用機率，或以矩陣 \hat{P} 代表使用鏈中所有的預估使用機率。

隨著測試案例的增加，測試過的使用狀態在使用鏈所有使用狀態中佔有的百分比，稱之使用狀態點涵蓋率(State coverage)，而測試過的連線在使用鏈中所有連線中所佔有的百分比，則稱之連線涵蓋率(Arc coverage)，此兩項數據可用來決定軟體測試驗證何時可以終止。

3.5. 超鏈結可靠度計算說明

超鏈結可靠度之計算基礎在於建構網站超鏈結之使用模式，依據測試鏈進行評估分析，網站超鏈結之使用模式之起始點為該網站之首頁(Homepage)，終點為 Terminated，原使用模式定義之可靠度計量方程式如下[17]，in 為初始狀態，term 為結束狀態， S 為轉換（非吸收）的狀態集合：

$$R_{in,term} = \hat{p}_{s_{in,term}} + \sum_{j \in \tau} \hat{p}_{s_{in,j}} \times R_{j,term} \quad (5)$$

例 1、以一個簡單的網站為範例，套用原使用模式定義之可靠度計量方程式，假設有一網站，共有 6 個網頁，透過 5 個超鏈結相互的連結，其架構如圖 6 所示，連結 2 與 4 因為沒有其他的連結點連結出去，故需連結至 Terminated 狀態點。

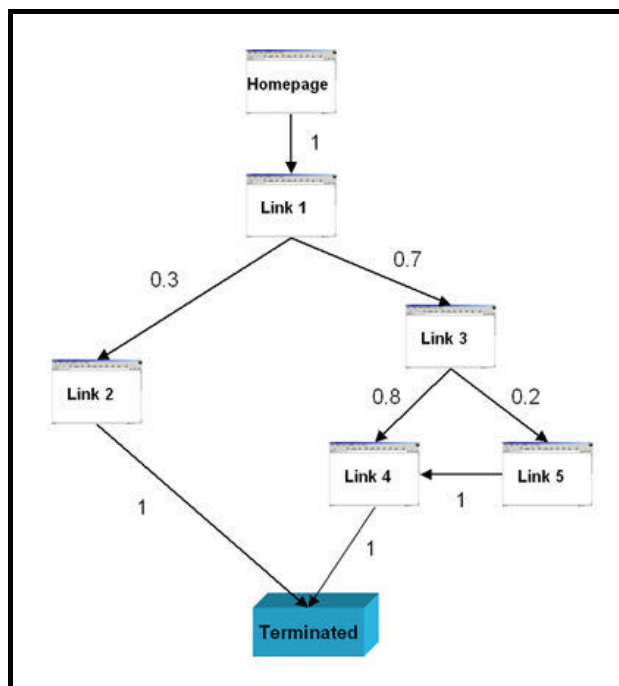


圖 6、超鏈結可靠度計算範例

從 Homepage 狀態點至 Terminated 狀態點共有三條路徑，整個網站超鏈結的可靠度如下：

路徑 1：Homepage → Link1 → Link2 → Terminated

路徑 1 之可靠度： $1 \times 0.3 \times 1 = 0.3$

路徑 2 : Homepage → Link1 → Link3 → Link4 → Terminated

路徑 2 之可靠度 : $1 \times 0.7 \times 0.8 \times 1 = 0.56$

路徑 3 : Homepage → Link1 → Link3 → Link5 → Link4 → Terminated

路徑 3 之可靠度 : $0.7 \times 0.2 \times 1 \times 1 = 0.14$

假設此網站超鏈結無任何錯誤，全部之超鏈結可靠度為路徑 1、2 與 3 的可靠度總加，如下：

$$0.3 + 0.56 + 0.14 = 1$$

假設 Link5 出現錯誤，因路徑 3 通過 Link5，路徑 3 之可靠度不納入計算範圍，整個網站可靠度如下：

$$0.3 + 0.56 = 0.86$$

3.6. 驗證機制

為了證明此網站超鏈結驗證機制的實用性，我們設計一驗證架構名為「Dr. Hyperlink」，用意為像醫生看診找出超鏈結之 HTTP 錯誤，此機制共有五個分系統分別是「網站使用模式建構」、「網站使用模式分析」、「測試案例產生」、「測試案例執行」與「評估結果」；「網站使用模式建構」先給此分系統該網站之首頁位址以及所要評估超鏈結之階層數，分系統會逐一掃描所需之超鏈結階層，並建構網站使用模式，以利後分系統「網站使用模式分析」之輸入分析此使用模式之結構，再由分系統「測試案例產生」能夠涵蓋網站超鏈結所有操作狀況之測試案例以工具針對這些測試案例加以執行已產生失效訊息，透過這些失效資訊得以計算產生網站之超鏈結評估結果，如下圖 7。

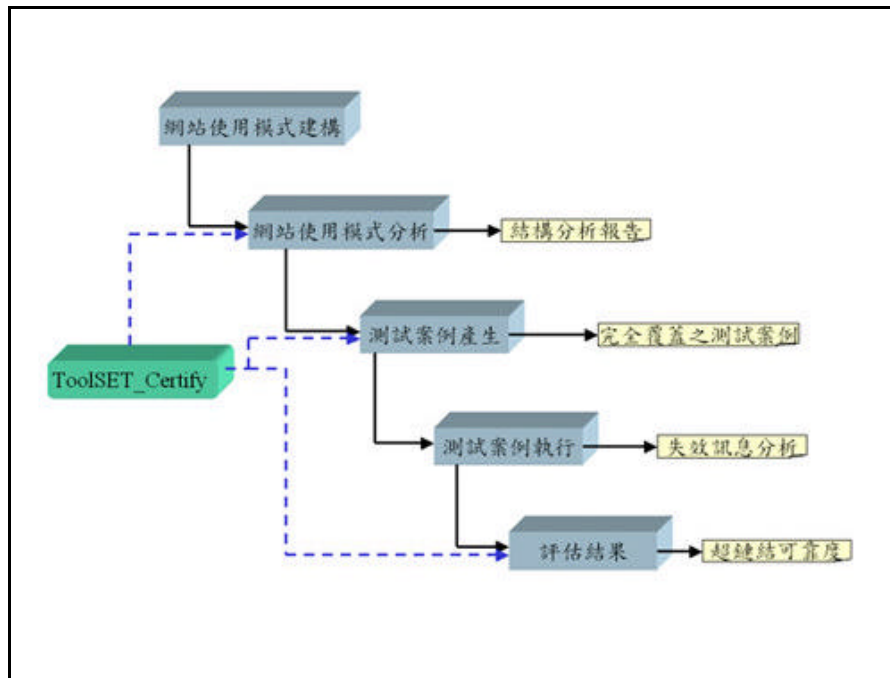


圖 7、網站超鏈結驗證系統架構 - Dr. Hyperlink

3.7. 評估流程

3.7.1 網站使用模式模式建立

評估流程的第一步是根據所蒐集的網站超鏈結以建立使用模式，利用 HTML 的內容使用工具進行分析超鏈結的工作，但因為網站上的超鏈結都是環環相扣，如果用工具分析會到無限層，所以此時測試者就必須判斷此次的測試需要評估至第幾層的超鏈結。

使用模式的建立是整個測試評估系統的第一步驟，建立的程序為假設將某個網站架構視為一個軟體系統，使用模式中定義一個完整的模式必須至少包含狀態起點與終點如圖 8。

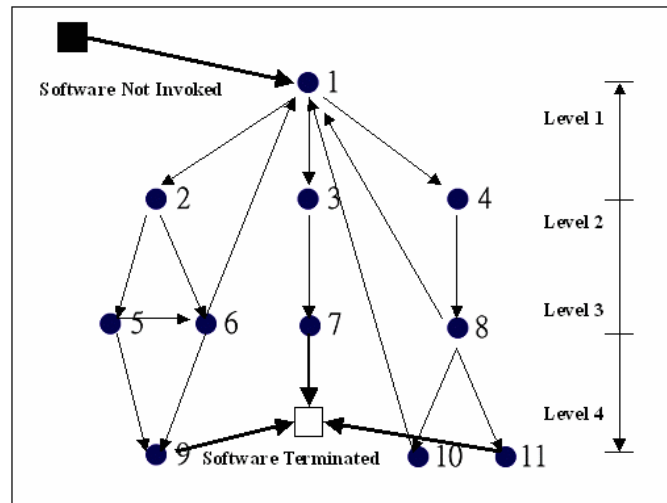


圖 8、完整使用模式架構圖

上圖中，編號為 1 的網頁是該網站之 root URL，通常是此網站的首頁，所以設定在 Software Not Invoked 之下的第一個狀態點，編號 7、9、11 的網頁因為沒有連出去的超連結，所以這三個網頁均要各別設定一連線到 Software Terminated，如此的模式才是一個完整的網站使用模式圖。

每個不同的網站都可以依據相同的方式來建制完整的使用模式，因為有著相同的建置方式，故此測試評估系統便可以自動化地產生網站之使用模式。

3.7.2 使用模式分析

網站使用模式模式建立後，分析其結構資訊，包括狀態點數、連線數、期望測試案例長度以及預計的測試案例數目。網站使用模式之狀態點數與連線數可以用人工進行分析，期望測試案例長度與預計的測試案例數目使用數學的方法加以推算，但通常網站使用模式相當的複雜，若要使用人工分析其複雜之結構資訊會耗時耗力，因此在本研究我們藉用 ToolSET_Certify 工具[18]進行分析。

3.7.3 執行測試案例

產生測試案例後，若此網站之超鏈結有任何 HTTP 錯誤則出現失效訊息的測試案例會被記錄下來。當測試者根據所產生的測試案例進行測試時，如果遇到失效的狀況，則分別記錄執行失效的測試案例所執行失效的步驟，以利後期網站超鏈結可靠度的計算。

3.7.4 評估結果

根據每個測試案例的通過/失效結果，從馬可夫鏈理論中可以計算出測試評估之數據，介於信賴區間 95% 與 99% 下的超鏈結可靠度以及所產生的測試案例下連結與狀態點涵蓋率。隨著產生的測試案例增加，連結與狀態點的涵蓋率也會隨著增加，重要的評估特性如表 3。

表 3、評估摘要

項目	內容
Script #	該測試案例的編號
Status	該測試案例的測試結果：通過/失效
R	超鏈結可靠度：理論上該網站系統超鏈結會正常工作的機率值(數值在0到1之間)
C=95%	對可靠度(R值)95%的可信度誤差評估：這個值在輔助R值的正確性。如果這個值等於0.02，而R=0.92，表示系統在95%的工作時間中，擁有0.92-0.02至0.92+0.02的可靠度
C=99%	對可靠度(R值)99%的可信度誤差評估：這個值在輔助R值的正確性。如果這個值等於0.02，而R=0.92，表示系統在99%的工作時間中，擁有0.92-0.02至0.92+0.02的可靠度

3.8. 系統架構

為了能將馬可夫模式計算網站之超鏈結可靠度的實際應用，我們設計一系統架構，其主要的實作及運算部分皆在伺服器端進行，為標準的

Client-Sever 的架構，使用者介面採用 Java AWT 元件建立出來的 Applet。

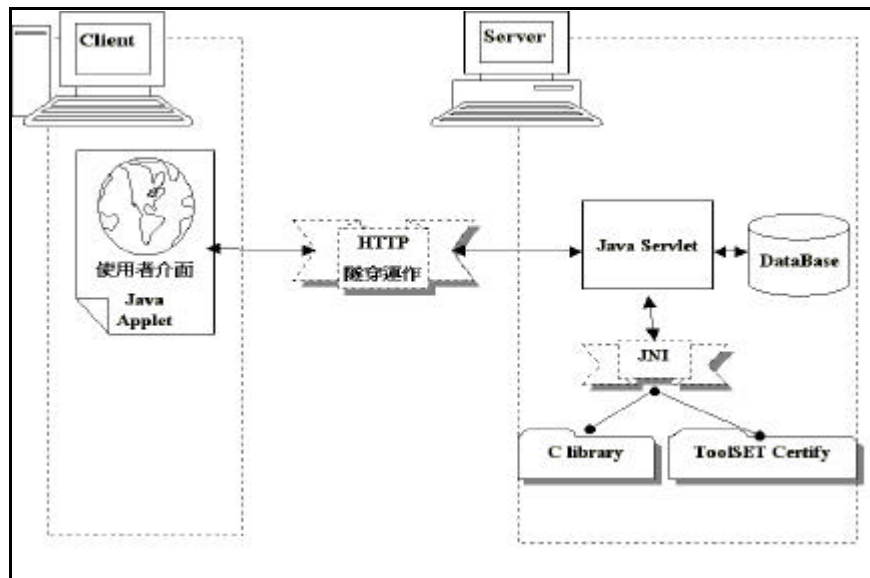


圖 9、系統自動測試架構圖

3.8.1 Client-Sever 間訊息及參數傳遞的機制

此評估測試系統架構配合使用者介面，採用 Java 技術中 HTTP 的隧穿運作(Tunnelin)從使用者介面傳遞訊息及參數呼叫相對之 Servlet，Java Servlet 是昇陽公司發展用以開發伺服器端程式，與 ASP、CGI 等較有效率且跨平台及亦可結合資料庫、RMI 等，之後再將 Server 端 Servlet 實作的結果及訊息回傳到 Client 使用者介面。

3.8.2 系統實作部份

採用 Java servlet 的方式來進行系統所有的實作，包括找尋連結點，資料庫連結，操作 ToolSET_Certify 工具之 API，實作測試案例，產生於網頁上的報告等。

3.8.3 資料庫

資料庫用 HypersonicSQL database , 用以紀錄所有的連結點的 URL 及其相關的資訊。

3.8.4 硬體架構

Client 端為任一擁有可執行 Java Applet 瀏覽器之平台 , Server 端之平台為 SUN SPARC5 Workstation, 安裝 SUN Solaris7 作業系統 Apache HTTP sever 、 Resin Java Servlet Engine 、 HypersonicSQL Database 以及 ToolSET_Certify 工具。

3.8.5 應用軟體技術

軟體應用技術使用 Java Servlet , Servlet 程式為 Server-side-Java , 在網站伺服器上執行 , 與 applet 或 JavaScript 不同在於當客戶端對 servlet 提出需求後 , 伺服器端會將 servlet class 及其所需相關的 classes 載入記憶體中 , 由伺服器端的 JVM(Java Virtual Machine)執行此 servlet 程式對客戶端進行服務。當完成服務後 , 此 servlet 仍會留在記憶體中 , 以供後續相同的需求做服務 , 所以在第一次呼叫 servlet 時 , 因需做 load classes 動作 , 速度較慢 , 爾後在對相同的 servlet 送出需求時 , 速度將會提升。

另一個使用的技術是 Java 的隧穿運作 , 隧穿運作建制在 Java RMI(Remote Method Invocation) , RMI 規格定義如何在網路上安全地使用 TCP/IP 通訊協定來實作遠端的 Method 的叫用 , 然後再加上 HTTP 的通訊協定 request-response 的方式 , 來作為訊息傳遞之用 , 但為了能指定遠端伺服器特定的 servlet 工作 , 而在其中產生子協定 , 子協定所包括的資訊 , 在伺服器端所應產生的物件及叫用物件的方法 , 其運作的流程如下 :

1. 開啟 HTTP 連線，HTTP 為無狀態協定，所以必須為每一個需求產生新的連線
2. 把需求予以格式化，包含 indicator，用來描述叫用哪個 method 以及所需的參數
3. 設定 HTTP 的 request header。包含傳送的資料型態及資料總長度
4. 傳送需求，將資料流寫入伺服器
5. 讀取需求。叫用指定的servlet, 並給予HTTP請求的資料, 讓servlet 取得欲叫用的 method 及所需的參數，如果為第一次呼叫 servlet，伺服器端會產生該 servlet 的物件的 instance
6. 叫用 method
7. 將 response 格式化
8. 設定 HTTP response header
9. 傳回 response
10. 關閉 HTTP 連線

3.9. 比較分析

在此我們以提出之驗證機制 Dr. Hyperlink 與其他工具方法進行比較，這些工具方法如前所述共分為三類，分別是分析網站伺服器之紀錄檔、直接掃描超鏈結及掃描已瀏覽過之超鏈結，比較的工具則有 Web Site Visitor Log, Linkbot, Astra 與 e-TESTER 四種，以不同的考慮因素進行比較如表 4

表 4、網站品質之評估工具比較

工具 比較因子	<i>Web Site Visitor Log</i>	<i>Linkbot</i>	<i>Astra</i>	<i>e-TESTER</i>	<i>Dr. Hyperlink</i>
測試案例產生方法	依網站伺服器之記錄檔	直接掃描超鏈結	記錄使用者的操作行為	記錄使用者的操作行為	系統化的使用模式建立
超鏈結涵蓋率	無	無	無	無	有
測試時間	長	短	長	長	短
測試結果	錯誤數目	錯誤數目	錯誤數目	錯誤數目	錯誤數目 瀏覽路徑涵蓋率 超鏈結可靠度
缺點	需要大量的紀錄檔 難以決定測試何時終止	難以決定測試何時終止	測試案例之產生耗時耗力 涵蓋率難以計算	測試案例之產生耗時耗力 涵蓋率難以計算	計算稍複雜
作業系統	Windows 95 / 98 / ME / NT / 2000	Windows 98 / NT / 2000	Windows 98	Windows 98	Windows 98 / Sun Solaris
發展機構	Bacus Laboratories 公司	Watchfire 公司	Mercury公司	Empirix Software公司	本研究

4、網站超鏈結可靠度計算之舉例說明

4.1. 應用範例介紹

由於網路的技術多元性，理論上網站超鏈結問題可以透過代理伺服器來解決，而且實際上目前也有部份的使用者使用代理伺服器瀏覽網站；然而，為說明簡單方便起見，本研究之網站超鏈結可靠度驗證方法將以不採用代理伺服器的情況下進行超鏈結之可靠度評估。

AutoFEA 網站 [19] 是 AutoFEA 工程軟體技術公司 (AutoFEA Engineering Software Technology Inc) 所有，公司總部設在美國加州。

AutoFEA 公司開發許多用於資訊應用的分析工具，這些分析工具用來幫助各個領域的業界。網站超鏈結的測試將選用此網站，其首頁如圖 10 所示。



圖 10、AutoFEA 網站之首頁

4.2. 建構使用模式

為了能套用統計使用測試，網站的超鏈結使用模式必須先建立起來。不過，網站的超鏈結幾乎都是環環相扣，永無止盡，沒有其終點，所以當建立網站的使用模式時，需要考慮到所要建立的使用模式的層級(Level)。如此一來，使用模式才不會無止盡的擴大。在這次的範例應用中，使用 4 個階層的使用模式。因此使用工具自動掃描該網站下的超鏈結是相當重要的步驟，圖 11 是 AutoFEA 之 4 個階層的網站超鏈結架構圖，其中 A 與 B 兩個圓圈表示其內還有數個連結點，而完整的網站使用模式至少包含起始與結束狀態點，每個路徑最後需連至結束狀態，為了使圖能簡單的表達，因此在圖 11 中省略結束狀態點。

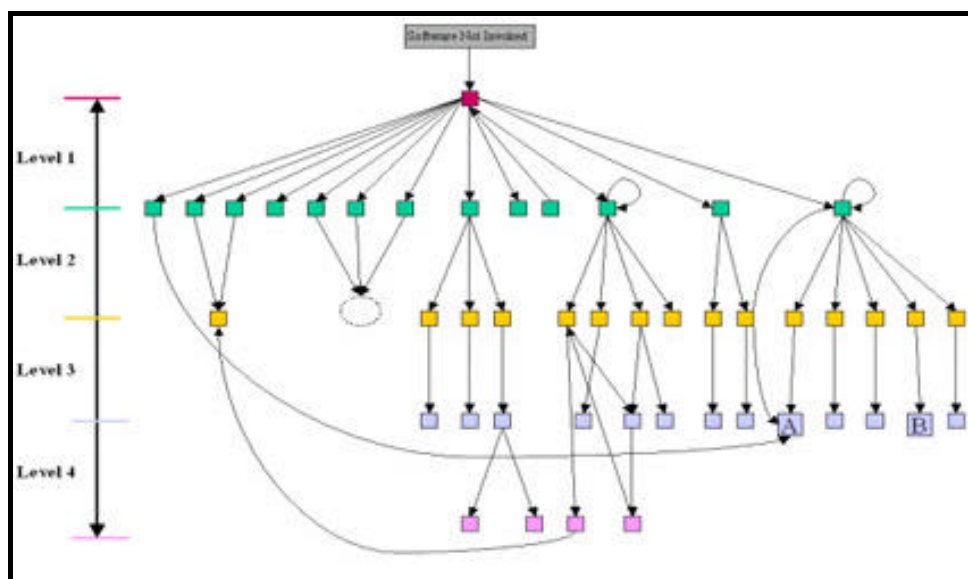


圖 11、AutoFEA 網站超鏈結架構圖

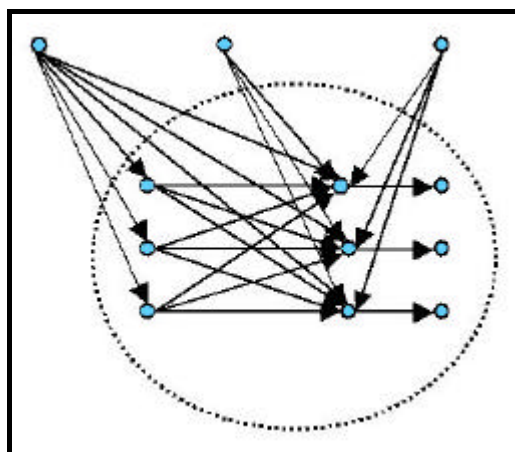


圖 12、A 狀態點之放大圖

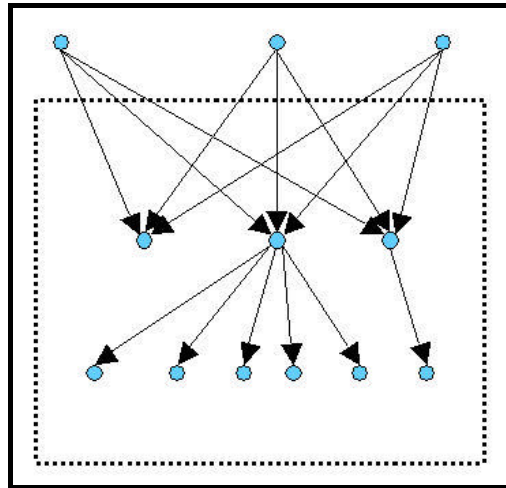


圖 13、B 狀態點之放大圖

4.3. 使用模式分析

建立 AutoFEA 網站使用模式後，經分析其模式結構，結果顯示共有 66 個狀態點數、連線數共有 123 個、期望的測試案例長度為 5.944、預測狀態點與連線完全涵蓋為 432(431.999)個測試案例，如表 5：

表 5、 AutoFEA 使用模式分析報告表

狀態點數	66
連線數	123
期望測試案例長度	5.944
狀態點完全涵蓋時需要案例數	431.999
連線完全涵蓋時需要案例數	431.999

4.4. 系統評估

4.4.1 測試案例

為說明方便，表 6 標示評估測試中之第六個測試案例，在此表中第一個測試步驟通常是連到該網站的 root 連結，也就是網站的 Homepage，第

二個測試步驟則是連結至網站中任一個超鏈結，以此類推，測試案例中的最後一個步驟需連結至網站使用模式的終點-Software Terminated。每個測試案例的長度不盡相同，內容也不一樣，越複雜或測試超鏈結階層越多的網站，所產生的測試案例則越多。

表 6、AutoFEA 網站之測試案例之一

測試步驟	期望連結點
Software Not Invoked	(起始狀態)
1. click to Homepage	Homepage
2. click to SHOP NOW	SHOP NOW
3. click to Version -\$2495	Version -\$2495
4. click to BUY NOW \$4995	BUY NOW \$4995
5. click to Web Store \$4995	Web Store \$4995
6. click to Software Terminated	Software Terminated (結束狀態)

4.4.2 失效分析

經過此次的測試案例執行，我們發現有一個超鏈結錯誤，此失效點發生在”with FEMAP for Windows”的超鏈結上，造成在 507 測試案例中共有 17 個測試案例包含錯誤此 HTTP 錯誤之超鏈結。因為在跑測試案例的過程中，只有一個錯誤被找到，所以在測試案例中每個錯誤的發生機率是一樣的。如表 7 之失效報告表，其中平均第一通道(Mean First Passage)是指該失效狀態點在測試案例中位於第一個轉換期望值，發生機率(Probability of Occurrence)是指該失效狀態點在某一測試案例中出現的機率。

表 7、AutoFEA 網站之失效分析報告表

失效編號	平均第一通道	發生機率
1	3056	0.001969
2	3056	0.001969
3	3056	0.001969

4	3056.000244	0.001969
5	3056.000244	0.001969
6	3056.000488	0.001969
7	3056.000488	0.001969
8	3056.000732	0.001969
9	3056.000732	0.001969
10	3056.000732	0.001969
11	3056.000732	0.001969
12	3056.000977	0.001969
13	3056.000977	0.001969
14	3056.001221	0.001969
15	3056.001221	0.001969
16	3056.001221	0.001969
17	3056.001221	0.001969

4.4.3 評估結果

建立馬可夫使用模式後，透過 *Dr. Hyperlink* 之輔助便能計算在 95% 與 99% 信心水準下的超鏈結可靠度，由於執行測試案例後發現一個錯誤，因此在執行 507 個測試案例後，狀態涵蓋率可以達到 100%，但是因為只有一個超鏈結錯誤，連線涵蓋率只能達到 99.187%；但若執行測試案例後沒有發現任何一個超鏈結錯誤，則狀態點與連線涵蓋率將可達到 100%，表 8 是 AutoFEA 網站超鏈結之最後評估結果。

表 8、AutoFEA 網站超鏈結之評估結果

Script #	Result	R	C=95%	C=99%	% States Certified	% Arcs Certified
496	失效	0.969758	0.001215	0.001596	100	99.187
497	失效	0.967807	0.001224	0.001609	100	99.187
498	正常	0.967871	0.001222	0.001606	100	99.187
499	正常	0.967936	0.001219	0.001602	100	99.187
500	正常	0.968	0.001217	0.001599	100	99.187
501	失效	0.966068	0.001226	0.001611	100	99.187
502	正常	0.966135	0.001224	0.001608	100	99.187
503	正常	0.966203	0.001221	0.001605	100	99.187

504	正常	0.96627	0.001219	0.001602	100	99.187
505	正常	0.966337	0.001216	0.001599	100	99.187
506	正常	0.966403	0.001214	0.001596	100	99.187
507	正常	0.966469	0.001212	0.001592	100	99.187

圖 14 是表示分佈在 507 個測試案例中，超鏈結的可靠度成長曲線，由圖中可清楚看到，在第 180 測試案例超鏈結的可靠度是不斷的下降，過了 180 個測試後，可靠度便穩定的成長，但到了約第 420 個測試案例後，可靠度再度降低，這個結果顯示，錯誤的超鏈結分佈在各測試案例的情形，因為包含錯誤的測試案例會降低可靠度的估算，這種現象說明錯誤的超鏈結大部分在 180 個測試案例以前以及 420 個測試案例後被瀏覽到。

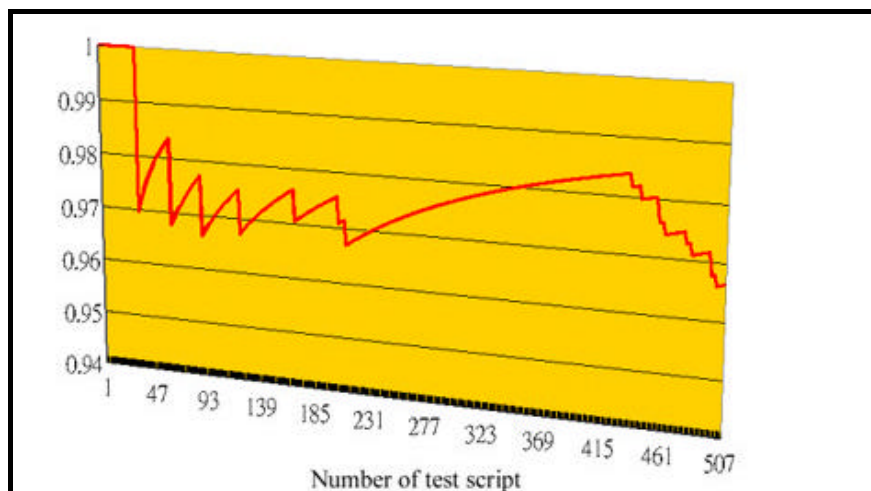


圖 14、分佈於 507 個測試案例中的可靠度曲線

第四章、利用等待線理論分析網站負載量

1、網站效能評估

1.1. 網站性能評估

網站性能的評估因網路環境之各種因素而有所變化，在此以其重要性與複雜性作為說明。

1.1.1 重要性

在分秒必爭的網路世界中，公司企業須應用最新網路科技，加速超越其他競爭者，否則終會被市場淘汰，而網站效能成了該網站成功與否的重要關鍵之一，而確保網站能迅速回應、易於瀏覽為網站設計的主要目標，為了使網站得到充分的利用，其方式便是進行網站的效能分析。

網站效能分析能監控並回報網站的使用情況，讓網管人員更瞭解使用者的各項瀏覽動作使網站得到充分的利用[20]，以增加使用者數目，除此之外網站效能分析並可確保網站的反應狀態正常，網站超鏈結無任何失效，瞭解網站內容的受歡迎程度以及新舊使用者的特性等。

1.1.2 複雜性

網際網路從 90 年代快速竄紅，隨著網路科技日新月異，目前的網路發展得比以往任何時候都要更複雜，網站的內容呈爆炸式增長，各樣的應用，如同伺服器應用程式、圖片、聲音、動畫、瀏覽器附加程式、XML[21]和 script，這些新技術的運用除了更能讓使用者體會聲光奪目外，在成本、頻寬、程式開發、可靠度及安全性的複雜度上也漸漸的增加。

因此進行網站效能分析時在各項因素的複雜度提高後使得驗證評估的工作難度更為困難，尤其近來上網人口的不斷增加，各個既有之線路頻寬

在某時段中出現壅塞的現象，網路頻寬的超載現象除了影響資料流的傳輸效能導致客戶端的需求輸入延遲外，亦造成測試工作上的困難，這也是本文研究之主要原因之一。

1.2. 通訊協定

電腦網路是由許多電子設備透過特定的傳輸媒介、通訊協定連結，使其能夠相互溝通，達到交換資訊、資源分享的目的[22]，構成構成網路的基本網路三要素為傳輸媒介、網路服務與通訊協定。

一般而言，網路傳輸媒介可分為多種，如雙絞線、同軸電纜、光纖電纜、無線技術、紅外線傳輸等，不論傳輸媒介分為幾種，其目的為進行網路資料的傳輸。網路服務(Internet service)亦分為多種，視伺服器所提供之服務而定包括網頁瀏覽服務、檔案下載服務以及電子郵件服務等，這些不同的網路服務有不同的通訊協定，網頁瀏覽服務使用 HTTP 通訊協定、檔案下載服務使用 FTP 通訊協定，而電子郵件服務使用 POP3、IMAP(Internet Message Access Protocol)與 SMTP(Simple Mail Transfer Protocol)[23]不同的通訊協定。

通訊協定代表各式協定技術與標準的統稱，每一種通訊協定都有一定規則，以解決傳送資料、識別目的地、處理錯誤情況、如何壓縮資料等等傳輸的步驟與方法。網際網路發展至今，瀏覽器擁有閱讀 HTML 文件的能力普遍為大眾接受，由於瀏覽器使用 HTTP 通訊協定，因此本文之網站效能分析研究以 HTTP 通訊協定下之網頁瀏覽服務為主要方向，HTTP 稱為超文件傳輸通訊協定，網站本身就是一個執行 HTTP 伺服器程式，而想要瀏覽網站內容的使用者，必須執行瀏覽器程式，瀏覽器程式會利用 HTTP 通訊協定跟網路上的 HTTP 伺服器程式要求網頁資料，將資料讀回，再按照網頁的 HTML 排版格式，將網頁呈現在使用者面前。有些網站檔案的傳輸使用 HTTP 通訊協定是為了讓更多的使用者同時連線，相較於 FTP 通訊協定，由於 FTP 通訊協定需消耗較大的頻寬，多人同時連線時網路容易阻塞，而每個使用者的等待時間與網站的反應時間也會隨著增加。

1.3. 網站效能測試

網站效能測試分成許多類，用以預測網站的相關性能表現，測試前首先需要考慮網站品質測試的定義、測試者如何實際的測量網站的效能等，一般來說，網站效能測試通常是給予測試者一段時間，在這時間內評估網站，效能測試可分為下列幾項：壓力測試、反應時間測試與負載量測試等[24]。這些測量的結果可以加以分析，若任何缺陷被找到，相關設計人員可以進行改善。

1.3.1 現有評估因子

網站效能測試的衡量標準及項目非常多，市售相關網站測試軟體也提供類似的測試。網站效能測試皆是一系列的測試，例如評估網站伺服器之負載量、效能瓶頸、模擬線上交易、伺服器反應時間、傳輸速度及頻寬流量等等。

效能測試的結果能提供衡量準則，管理人員可以針對這些測試結果在不變更軟硬體架構的情況下進行伺服器的最佳化，使其得到最大的效能表現，若要變更軟硬體架構的情況下，在變更後可以與之前未變更前的情況加以比較，評估變更後是否達到預期之效能輸出。以電子商務來看，網站效能測試更形重要，顧客消費與瀏覽行為的改變都可藉由網站的效能測試而得知。

1.3.2 網站效能測試種類

網站效能測試分為多種，包括壓力測試、反應時間測試、負載量測試與前測測試(Smoke test)等，在本文中主要說明壓力測試、反應時間測試、負載量測試。

[1] 壓力測試

當網站達到負載顛峰時時會發生什麼事情？絕大多數網站經營層、管

理者最想要知道的事，網站負載時應該採取什麼樣的行動對公司、網站本身以及顧客是最好的。壓力測試有兩個主要的目標，第一、分析網站何時出現超載，第二、預測網站出現壓力負載後出現什麼情形。壓力測試通常包含了需要大量計算或模擬大量使用者的方法。當網站超載時可能發生不可預期的後果，壓力測試可以防止一些潛在性的危險[24]，另一方面，網站的軟硬體的升級後，亦需要壓力測試來評估升級後的效能。

通常網站出現超載的情形時解決之方法可分為兩類，一類是假設網站只能提供 1000 位使用者同時連線，那當第 1001 個使用者使用者連線時，網站伺服器向第 1001 位使用者發出”server not available”訊息，另一類作法是讓此使用者進入伺服器的等待線(Queue)中，一旦有其他使用者離線，立即的服務在等待線中的使用者。

測試的時間主要由測試的方法所決定，若不是使用工具模擬使用者使用網站，則壓力測試的時間則是該網站使用的尖峰時段，若使用工具模擬使用者使用網站，為了讓測試結果更正確、則壓力測試的時間必須選擇人數最少的時候，通常是凌晨使用者最少的時候進行測試。

[2] 反應時間測試

反應時間是指系統或具有功能之元件受到輸入之訊號產生反應所需要的時間。以網站的反應時間來看，指介於客戶端電腦從使用者需入一個鍵發出需求的時間與伺服器端電腦接受到需求第一個字元的這兩段時間為反應時間。

網站反應時間的重要性在於反應時間越快，客戶所得到的回應則越快，通常使用者等待網站的回應只有數秒鐘之久，在競爭的激烈網際網路市場中，若網站的反應時間超過一般使用者所能接受的等待時間，所造成的結果將是顧客的流失以及網站本身的競爭力降低。

[3] 負載量測試

網站負載量測試是評估短時間之內網站的性能表現，有助於找到該網

站之效能瓶頸，在發展的早期，公司企業可以參考負載量評估的結果用以決定該網站硬體等級，網站尚未連線前，若能適時的找出該網站之效能瓶頸，亦夠有效的預防網站超載。

負載測試的項目包含瞭解網站系統的能力極限與管理其網站性能與成長。Newport Group 公司就預測負載測試工具與相關的服務市場，在 2000 年時到達 3 億 8 百萬美金，到了 2003 年更會到達 7 億 4 千 6 百萬美金[25]，由此可見負載測試評估對許多公司企業來講是非常重要的。

1.3.3 現有網站評估技術與工具

目前現有的網站效能評估技術、工具與服務有許多種，本文中主要介紹三個網站效能之測試工具，分別是 eValid、WCAT 與 OpenLoad，在本文後段我們將以這三個測試評估工具與本研究提出之方法作為比較。

[1] eValid

eValid 產品是位於美國舊金山 eValid 公司開發[26]，eValid 的測試引擎為 Test Enabled Web Browser™，提供瀏覽器端的界面讓測試者進行網站的品質確認、動態測試、網站內容的驗證、效能調校、網站伺服器負載與能力測試等功能。eValid 在網站測試上提供提供獨特的方法能夠以簡單、快速與有效性的進行網站的效能測試評估。網站負載測試方面，eValid 可以產生模擬 100 個或 1000 個使用者的測試案例，執行這些測試案例以評估網站的負載能力。網站正確性測試方面，eValid 亦可模擬使用者的操作動作，以驗證網站的超鏈結正確與否、HTML 語法是否有錯。此方法之缺點在於測試案例之產生需藉由測試者連上伺服器的動作，因此測試案例之產生需要大量的時間，且這些測試案例所記錄的使用者動作是否能代表所有使用者的使用行為也難以評估。

[2] Microsoft WCAT load test tool

WCAT 負載測試工具是由微軟所開發[27]，針對微軟系統本身的

IIS(Internet Information Services)與 NT 進行測試，WCAT 發出模擬的數百個使用者需求後，在這一段的尖峰時間 WCAT 可監控網站伺服器的狀況，亦有報表的產生，不管是簡單型的網站伺服器，還是網站搭配資料庫的使用都可以進行網站效能的評估。此方法之缺點在於使用模擬的方式是模擬多個需求進行測試評估，測試結果與真實的環境仍有些出入。

[3] OpenLoad

OpenLoad 是由 OpenDemand 公司所研發[28]，它是一套完整的網路系統負載測試工具，在負載測試方面，測試案例的產生可以藉由模擬上百或上千人數的使用者每天進入網站伺服器的使用狀況，這些測試案例可以被重複的使用也能自動的產生，且可以透過程式將負載程度設定為某一等級，若超過此一等級，測試者便可與前次的測試進行比較。此方法之缺點在於使用模擬的方式是模擬多個需求進行測試評估，與分析的方法比較會消耗較多的測試資源。

2、等待線理論

2.1. 等待線模型

等待線理論源自於等待線的系列問題，如飛機航線的排班與顧客排隊買票等，進一步的等待線理論研究最早於 1940 年代末期，第一本介紹等待線理論的是由 Morse 於 1958 年所寫的”Queues, Inventories, and Maintenance”[29]，在經歷了近 30 年的演進，等待線理論從一開始初淺研究演變成今日在各大期刊上發表的論文研究，其論文數已經超過 1000 篇以上。

基本上，等待線模型的特徵是由三個主要元件所組成，分別是使用者(Customer)、伺服器與等待線 [30]，其表示如圖 15：

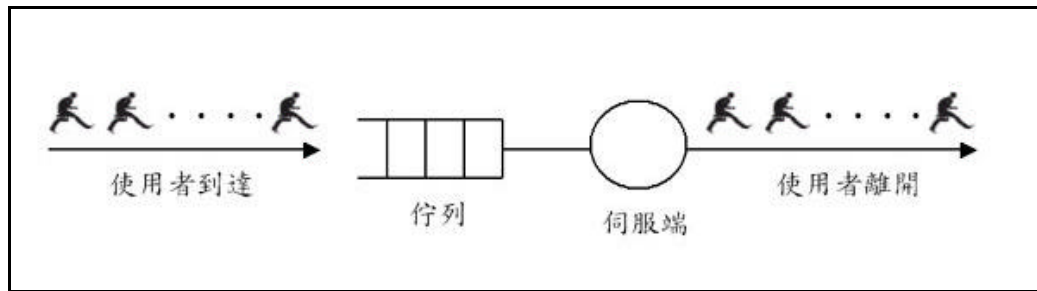


圖 15、基本等待線模型

使用者、伺服器端與等待線之描述分別如下：

2.1.1 使用者

使用者進入等待線模型後需考慮到許多因素，如主動離開(Reneging)、止步(Balking)與轉換(Jockeying)，主動離開是指使用者進入等待線後是否有主動離開而不再等待的情形，止步為使用者在到達之前不進入等待線中等待，轉換之動作為使用者從一等待線轉換至另一等待線上，這些使用者非預期的動作將會造成評估上的困難。

2.1.2 伺服器端

伺服器端為一實體系統用以提供各項服務予各個使用者，而服務使用者的其間稱為服務時間，評估伺服器端的系統方面也需考慮各項因素，如同伺服器對每位使用者的服務時間是否一致？以及當使用者為無限大時伺服器是否還可以繼續提供服務等。另一方面，伺服器的組態(Configuration)設定亦是相當重要，因與該伺服器系統是否能提供平行處理、為使用者服務時使用什麼策略等有相當重要之關連。

2.1.3 等待線

等待線中有優先權之排列順序，如使用先進先出排序(First come first served)與後進先出排序(Last come first served)等。不同的排序對使用者來說

有不同的服務表現，而評估等待線模型時等待線本身也佔了很重要的因素之一，有些等待線模型採用單一之等待線服務使用者，有些等待線模型使用雙等待線服務使用者，等待線的數目亦會影響評估系統時的困難度。

2.2. 需求到達率

網站尖峰期間需求來到某個網站時，並不是分散到達，而是一大群的需求一起連到該網站。設計如何進行效能測試時，應考慮在測試階段中該網站是否會出現超載之現象。需求到達率是指在一固定時間內有多少個需求進入系統被服務，此數據配合等待線理論技術之應用達到適合於評估某個網路效能的目的。

3、網站系統之負載量分析

3.1. 反應時間

反應時間在目前分秒必爭的社會中相當的重要，尤其進入 e 時代的電腦網路境界中，以使用者的觀點來看 0.1 秒的反應時間，這段時間可以讓使用者認為網路系統的反應迅速，若反應時間為 1 秒相當於使用者思考的瞬間，若網站系統能在 1 秒內做出回應，則不需任何的回饋機制，若反應時間超過 10 秒，在這段等待時間中使用者可能失去集中的注意力而轉移至其他軟體程式或網站，所以有些網站考慮這點因素在使用者等待的其間給予使用者回饋的機制。在本研究中我們定義網站反應時間如下：

定義 1：網站反應時間(Response time of Website)是指從使用者發出服務需求到伺服器發出回應時間之總和，其中包含使用者的等待時間及伺服器的服務時間。

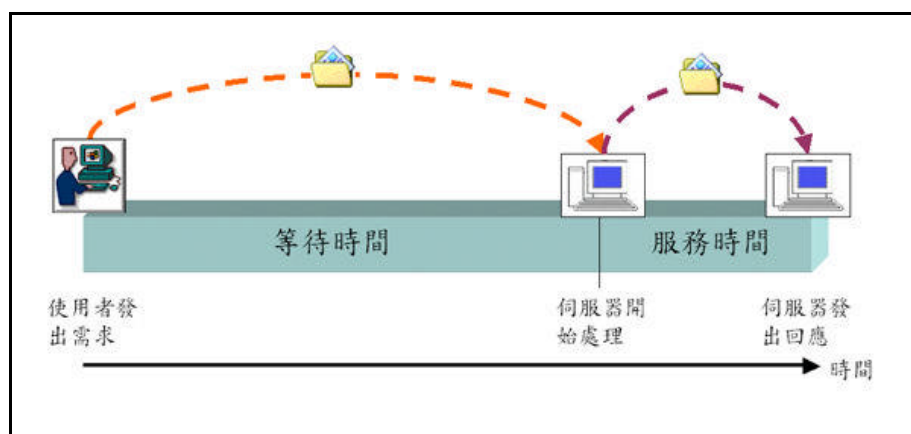


圖 16、網站反應時間示意圖

假設等待時間為 T_w ，服務時間為 T_s ，前述之網站反應時間為 T ，則：

$$T = T_w + T_s$$

3.2. 服務率

網站系統的服務率越高，被服務的需求數目則越多，在網路的環境中，服務率與全部進入系統的需求數目有著特定的關係。本研究中假設所有的需求在被系統服務前，先進入等待線中等待服務，由於網站系統的能力一定，因此進入的需求數越多則服務率會越低，但藉由調整網站伺服器的連線逾時時間(Connect time)可以讓服務率提高，藉由服務率的提高，每個需求的網站反應時間又會增加，因此如何在這些效能數目上取求一個平衡點為本研究之重點方向。在本研究中我們定義網站服務率如下：

定義 2：網站服務率(Service rate of Websites)是指被系統服務之需求數目佔進入系統等待線中等待需求數目的百分比。

3.3. Little 關係式

Little 關係式是個非常普遍化的分析公式，當兩個值成已知的狀態可求出第三個值，用於評估電腦系統之效能值[31]，Little 關係式適合用在所有的等待線原理，不過 Little 關係式在評估之前需要相當的假設才能使所求的評估值變的有效。

原有的 Little 關係式中，有兩種方法可以計算使用者的等待時間。第一個方法是將每個需求的等待時間進行總加，所獲得的便是全部需求的等待時間。例如有五個需求分別是 A、B、C、D 與 E，在不同的時間內進入系統，此時全部的需求等待時間則為 A+B+C+D+E，其計算之關係式如下， $W_q(i)$ 為第 i 個需求在等待線中的時間。

$$W = \sum_{i=1}^N W_q(i) \quad (6)$$

另一個計算全部等待時間的方法是以積分的方法計算，求出時間 a 與 b 期間每一時間點上之等待時間總和，如下關係式 2， $L_q(t)$ 為第 t 時間點的等待時間，a 為時間起點，b 為時間終點[32]。

表 8、關係式 1、2 與 3 之符號表

符號	解釋
(t)	t 時間進入系統的需求數目
(t)	t 時間離開系統的需求數目
$L_q(t)$	$(t) - (t)$
L_q	服務需求數
$W_q(i)$	第 i 個需求在等待線中的時間
W_q	需求在等待線中之等待時間
a	需求進入系統時間起點
b	需求離開系統時間終點

$$W = \int_a^b L_q(t) dt \quad (7)$$

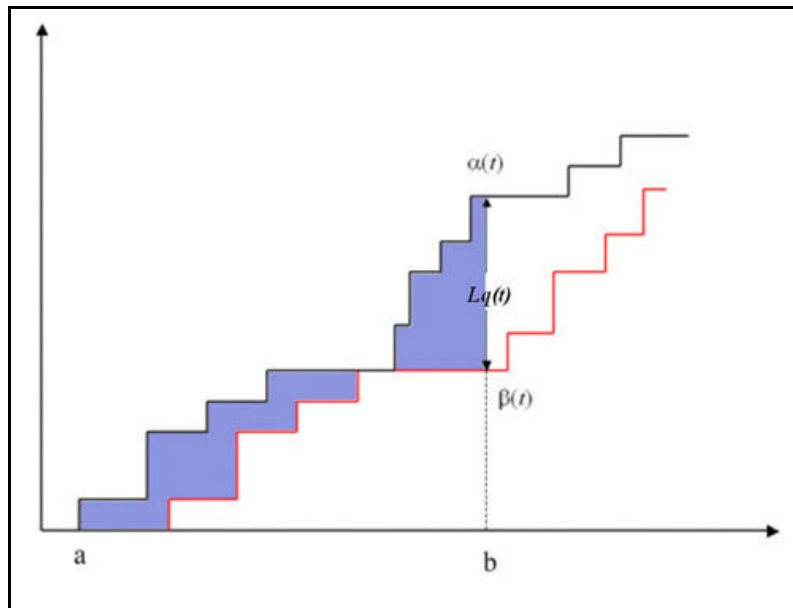


圖17、t時間系統需求數計算圖

$$W = \int_a^b Lq(t) dt \quad (8)$$

而 Little 關係式的整個表達式如下：

$$L_q = W_q \quad (9)$$

為了符合網路的特性，我們修改 Little 關係式以求得更精確的評估結果，在實際的網路環境下，當大量的使用者連線至網站超過其負荷時，一定會有一些使用者沒有辦法接受服務，而每個網站伺服器皆設定連線逾時時間，一旦這些使用者超過連線逾時時間則被網站伺服器踢除，換句話說從使用者發出需求至網站伺服器的這段期間沒有被服務，這段期間稱為無效等待時間(Idle waiting time)，這個觀念是我們修改 Little 關係式最主要的出發點。

3.4. 修正後 Little 關係式

下圖 18 用以描述修正後 Little 關係式的觀念，圖中假設有五個需求分別是 A、B、C、D 與 E，同時連線至網站伺服器，D 與 E 需求的等待時間等於網站伺服器的連線逾時時間，一旦超過等待時間超過伺服器連線逾時時間則需求被伺服器踢除，故 D 與 E 需求缺少服務時間。D 與 E 的時間總加為”無效等待時間”，真正全部等待時間為 A、B 與 C 的總加，因此等待時間為連線逾時時間與服務時間的總加。

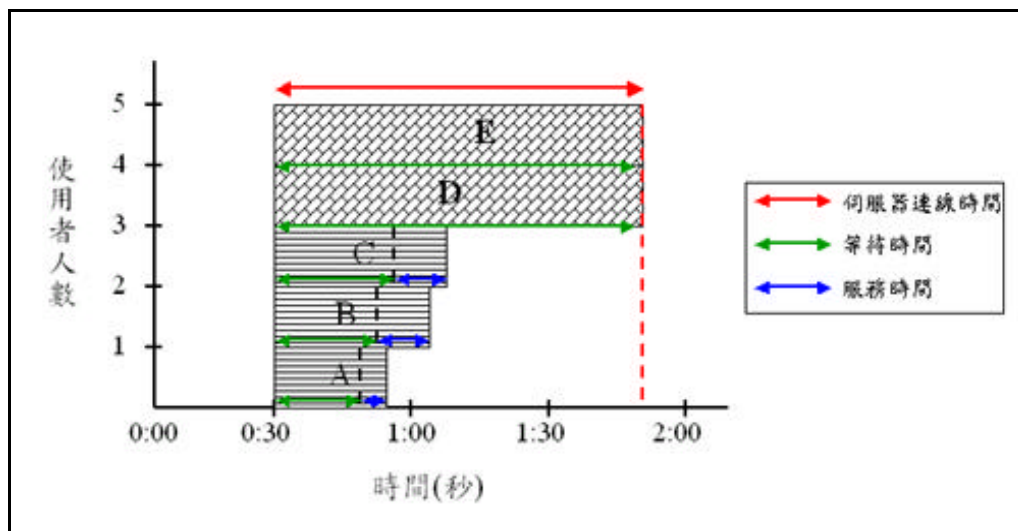


圖 18、需求全部等待時間 W_q 表示圖

由於網際網路實際的情況相當的複雜，因此要評估網站所有情況之性能表現將會相當困難，以下我們列出一些基本的假設，這些假設規範網路複雜情況之簡化狀況，其修正後 Little 之基本假設如下：

1. 所有的需求在等待期間中是在唯一之等待線中等待。
2. 當其需求之等待時間超過網站連線逾時時間，即稱為無效等待時間。
3. 大量需求進入伺服器時，假設幾乎在同一時間內同時進入等待線。

4. 每一需求進入等待線後幾乎持續等待，即無主動離開、止步與轉換之情形發生。

介紹修正後 Little 關係式前，首先介紹於文中使用的等待線符號表如下：

表 9、等待線符號表

符號	解釋
\bar{R}_s	被服務需求之平均網站反應時間
R_t	被服務需求之總網站反應時間
	需求到達率
L_n	未被服務之需求數目
c	網站連線逾時時間
n	被服務之需求數目
N	全部需求數目
$W_q(i)$	第 <i>i</i> 個需求在等待線中的時間(秒)
t_1	第一個需求到達等待線的時間(秒)
t_2	最後一個需求到達等待線的時間(秒)

定理 1、未被服務之需求數可依下式計算：

$$L_n = [(t_2 - t_1)] - \frac{R_t}{R_s} \quad (10)$$

證明：

為了使修正後 Little 關係式符合網際網路環境之特性，我們定義 $W_q(i)$ 有兩種情況，第一種為 $W_q(i)$ 小於連線逾時時間，而有被服務的需求平均等待時間如下關係式：

$$\overline{R_s} = \frac{Rt}{n}, \text{ 當 } W_q(i) < c$$

另一種情況是當 $W_q(i)$ 大於連線逾時時間時，這些需求將無法被網站伺服器所服務，因此，未被服務的需求數目可以如下關係式所代表：

$$L_n = N - n, \text{ 當 } W_q(i) \geq c$$

$$\text{因 } \frac{N}{t_1 - t_2}$$

$$\text{故 } N = (t_2 - t_1) \cdot \overline{R_s} = t_2 \cdot \overline{R_s} - t_1 \cdot \overline{R_s}$$

最後修正後之 Little 關係式如關係式 3 所表示：

$$L_n = N - n = (t_2 - t_1) \cdot \overline{R_s} = [(t_2 - t_1)] \cdot \overline{R_s}, \text{ 故得證}$$

例 1、為說明方便，在此舉一小例說明：

假設在 $t_2 - t_1 = 1$ 秒內有 100 個需求數(N)進入網站伺服器的等待線中

$$\text{則需求到達率} = \frac{N}{t_1 - t_2} = \frac{100}{1} = 100$$

假設網站伺服器的連線逾時時間為 10 秒($c=10$)，被服務的需求總網站反應時間(R_t)為 490 秒，被服務的需求平均網站反應時間($\overline{R_s}$)為 7 秒

故修正後之 Little 關係式 $L_n = [(t_2-t_1)] \cdot \frac{Rt}{Rs} = [100 \times 1] \cdot \frac{490}{7} = 30$

網站服務率為 $\frac{70}{100} = 70\%$

由結果可得知當網站之需求到達率為 100 且連線逾時時間設定為 10 秒時，其網站的服務率為 70%。本研究之目的為在不同連線逾時時間下，當不同的需求數進入網站後，可協助求出該網站服務率，以供相關人員參考。

3.5. 比較分析

在此我們與三個其他的網站效能測試工具進行各項功能特徵的比較，分別是”eValid”、” Microsoft WCAT load test tool”與” OpenLoad”效能測試工具，目前效能測試的方法主要分為兩種，一種是以產生測試案例的方法進行測試，另一種是以模擬多位使用者使用網站伺服器的狀況進行測試評估，這些工具進行負載量測試時，皆只考慮被服務之需求，而求得之反應時間亦是被服務需求之反應時間，與本研究之分析方法比較如下表 10。

表 10、分析比較表

方法 考慮因子	eValid	WCAT	OpenLoad	本研究
被服務的需求	V	V	V	V
未被服務的需求	X	X	X	V
網站伺服器連線逾時時間	X	X	X	V
被服務之需求反應時間	V	V	V	V
未被服務之需求反應時間	X	X	X	V

4、網站系統反應時間之實例介紹

4.1. 應用實例介紹

為了能更確切的說明修正後 Little 關係式，我們以 3D spotlight 網站作為範例，其首頁如下圖 19，3D spotlight 網站成立於 1998 年[33]，在這幾年當中已經服務過成千上萬的使用者，每天的更新內容為介紹 PC 相關的軟硬體、遊戲最新消息以及報導文章等，選擇 3D spotlight 網站為實例是因為此網站行之有年且為各大媒體雜誌所報導，如 ZDNet、Computer Gaming World、PCWorld、MaximumPC、PC Magazine、The Register 等。



圖 19、3Dspotlight 網站

4.2. 資料蒐集

資料蒐集的目的是讓修正後之 Little 關係式能夠計算出沒有被服務的需求數目。在此測試環境中，我們設定全部的需求在一秒內進入網站伺服器中，進入需求的數目分別是 500、1000 與 1500。

在資料設定後，使用工具分析其 R_s 與 L 資料， N 為全部需求數目，

t_1 第一個需求到達等待線的時間， t_1 為最後一個需求到達等待線的時間， R_s 為需求到達率， R_s 為被服務需求之總網站反應時間， \bar{R}_s 為被服務需求之平均網站反應時間， L_n 未被服務之需求數目，如表 11：

表 11、資料蒐集與設定

N	t_2-t_1		R_s	\bar{R}_s	L_n	$N-L_n$
500	1	500	1358	3	47.334(47)	453 (90.6)
1000	1	1000	15412	17	93.412(93)	907 (90.7)
1500	1	1500	21634	20	118.3(118)	1081 (72)

4.3. 評估結果

根據修正後 Little 關係式以及表 11 的資料，我們計算出在不同連線逾時時間下與需求數下的服務率與等待時間，如表 12。

表 12、不同需求數下的網站服務率(%) 與需求等待時間

全部需求	C=30		C=40		C=50	
	網站服務率 (%)	網站反應時間(s)	網站服務率 (%)	網站反應時間(s)	網站服務率 (%)	網站反應時間(s)
500	90.6	3	95.5	11	97.4	19
1000	90.7	17	94.7	26	92.3	34
1500	72	20	79.8	28	86.1	39

圖 20 是 3Dspotlight 網站最後的評估結果，圖中我們發現當網站伺服器連線逾時時間設定的越長，則服務率越高，但是平均等待時間也會跟著變長，在最佳的情況下，要讓服務率越高，平均等待時間越短，但是這種最佳情況幾乎是不可能的，因為每個網站伺服器的能力皆為有限，本研究

之目的在於將評估結果給予網管人員做為參考，使得在不變更軟硬體的環境下可以使網站的效能達到最佳的效率。

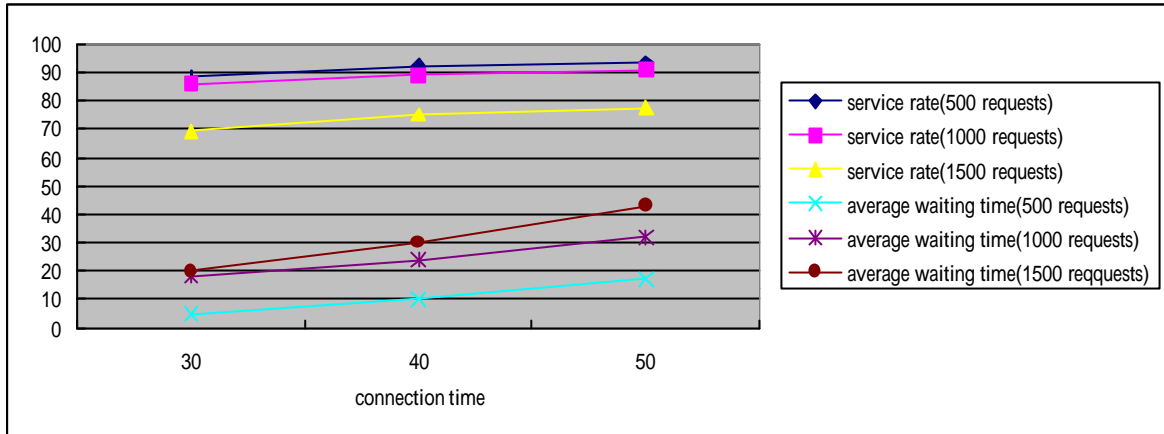


圖 20、3Dspotlight 網站最後評估結果

第五章、結論與未來研究方向

1、結論

無可諱言，對於一個經營健全的網站而言，不論是在市場定位或獲利能力方面，其網站的品質水準已成為關鍵因素之一，若網站本身有任何的超鏈結錯誤，將造成使用者使用的意願降低，網站的競爭力相對地也會隨之降低。

然而，網站是由許多元件組成、如文字、圖片、聲音、超鏈結、影像等，由於這些因素增加網站品質評估測試的困難，因此在本研究中首先提出相較於軟體可靠度之超鏈結可靠度定義，再提出網站品質分析的架構，分析流程中應用統計使用測試的方法用以評估網站之超鏈結可靠度，其優點在於可產生最少數目的測試案例，相較於其他的分析工具與方法，這些測試案例更可以系統化百分之百的涵蓋整個網站所有超鏈結操作模式，最後提出驗證超鏈結可靠度之驗證機制，在這個驗證機制裡有完整的測試步驟，用以品質分析結果的正確性。

網站效能分析方面由於網際網路的特性，當大量的使用者同時連線至網站超過網站的負荷會造成使用者無法連線之問題，為了評估此問題之影響程度，本研究配合網際網路之特性修改 Little 關係式評估網站效能表現，測試的結果可提供各界參考。

本研究於提昇網站系統之品質方面，其助益有下列四點：

1.1 軟體使用測試之機制可應用於驗證網站超鏈結之可靠度評估

本研究將軟體使用測試應用於驗證網站超鏈結可靠度之評估上，並提出測試驗證機制，包括網站使用模式之建立與分析，用以統計測試方法產生與執行測試案例，最後求出網站超鏈結之可靠度，俾利國內外業界與學術界之參考與應用。

1.2 網站超鏈結評估流程之自動化

本研究提供網站超鏈結驗證流程的探討，其中一項優點為自動化流程之設計，從網站使用模式之建立、測試案例之產生、執行以及最後的評估結果皆為自動化之設計，使得網站品質的評估能更為快速，減少測試資源之浪費。

1.3 修改應用 Little 關係式以獲得網站效能分析結果

以原先用於評估等待線理論之 Little 關係式加以修改並應用至網際網路環境，配合網際網路之特性以求得網站系統之效能分析，最後獲得網站之負載量等效能評估數據，提供網站相關人員參考使網站達到效能之最佳化。

2、未來工作

本研究針對超鏈結可靠度與網站效能進行分析，除了獲得一些結論可供參考運用外，仍有許多後續研究可補充加強的地方，在超鏈結可靠度的未來研究方向上，我們期望運用此驗證機制流程評估其他的網站元件，例如應用 Java 相關技術網站的品質評估、分析使用資料庫的網站與其他新的網站技術，以使此驗證機制流程更加完整；在長遠目標上，更期望應用此機制至其他更複雜之網站，如使用 ASP、Active X 等技術，且能有更精確的評估分析。

在網站效能分析的未來工作方向上，由於網站系統的複雜性與網路環境之不穩定性造成測試時的困難度提高，雖然本研究提出網站效能之分析評估方法能夠以節省測試資源的方法進行評估，但在測試結果的正確性上期望在未來能更為精進，所評估之數據可以更上一層樓。在未來亦期望發展完整之測試機制，以其發揮其最大效益。

參考文獻

- [1] Bacus Laboratories INC Web site, <http://www.bacuslabs.com/WsvlCore.html>, (April 2002).
- [2] Watchfire Inc Website, <http://www.watchfire.com>, (April 2002).
- [3] Mercury Interactive Inc Website, <http://www-svca.mercuryinteractive.com/>, (April 2002).
- [4] Empirix, <http://www.empirix.com/>, (April 2002).
- [5] Lan. Sommerville, “Software Engineering”, Sixth Edition, Pearson Education Limited, 359-363, (2001).
- [6] 經濟部工業局, 「軟體技術文件手冊」, 軟體工業五年發展推動計畫, (2001)。
- [7] Hoan Pham, “Software reliability,” Springer, (2000).
- [8] Stephane Gigandet, Ashok Sudarsanam, and Anshu Aggarwal, “The Inktomi Climate Lab: An Integrated Environment for Analyzing and Simulating Customer Network Traffic”, (2001).
- [9] Stacy J.Prowell, Carmen J. Trammell, Richard C. Linger and Jesse H. Poore: “Cleanroom Software Engineering – Technology and Process”, Addison Wesley Longman Inc, 91-92, (1999).
- [10] G. H., Walton, J. H. Poore and C. J. Trammell: "Statistical Testing of Software Based on a Usage Model," Software Practice and Experience, vol. 25(1), 97-108, (January 1995).
- [11] Wen-Kui Chang, S. Twu and W. Teng: “Ensuring Functional Test Coverage For Avionics Control Applications Through Statistical Usage Testing,” FESMA'99 - 2nd European Software Measurement Conference 4-8 Amsterdam, the Netherlands, 261-268, (October 1999).
- [12] Wen-Kui Chang, Shing-Kai Hon, “A practice on performance testing for web-based systems”, The Asian Journal on Quality / Vo1.1 No. 1, (December 2000).
- [13] J.A. Whittaker, K. Rekab, M.G. Thomason, “A Markov chain model for predicting the reliability of multi-build software”, Information and Software

- Technology 42, 889-894, (2000).
- [14] 王咨博,「分散式系統之軟體可靠度的度量研究」,東海大學資訊工程與科學研究所,碩士學位論文,(2001)。
- [15] James A. Whittaker, Jeffrey Voas, “Toward a More Reliable Theory of Software Reliability”, IEEE Computer, 36-42, (December 2000).
- [16] 張文貴、王咨博,「失效加權軟體可靠度模式化之評估系統」,中華民國電腦學會電腦學刊, 12-29, (December 2001).
- [17] K. Sayre, J.H. Poore, “Stopping criteria for statistical testing”, Information and Software Technology 42 851-857, (2000).
- [18] Q-Labs, ToolCertify User Guide, Version 4.0 (1999).
- [19] AutoFEA Website, <http://www.autofea.com>, (April 2002).
- [20] Wen-Kui Chang, Shing-Kai Hon. “A PRACTICAL ANALYSIS OF STRESS TESTING FOR A WEB-EXAM SERVER”, 11 ICSQ International Conference on Software Quality, (2000).
- [21] Norman Walsh, “A Technical Introduction to XML”, XML white paper, (1999).
- [22] Ed Wilson, James Naramore, “Network Monitoring and Analysis: A Protocol Approach to Troubleshooting”, (2000).
- [23] 朱正忠、張景勳, ”網際網路與電子商務”, 全華科技圖書股份有限公司, (2000).
- [24] Daniel J. Mosley, “Client-Server Testing on the Desktop and the Web”, Prentice Hall PTR, (2000).
- [25] Steven Splaine, Stefan P. Jaskiel, Alberto Savoia, “The Web Testing Handbook”, Software Quality Engineering Pub, (2001).
- [26] eValid Web site, <http://www.soft.com/eValid/evindex.html>, (May 2002).
- [27] Microsoft WCAT load test tool Web site, http://www.microsoft.com/windows2000/techinfo/reskit/en/iisbook/c05_web_application_stress_tool_and_wcat.htm, (May 2002).
- [28] Opendemand Web site, <http://www.opendemand.com/openload/features.shtml>, (May 2002).

- [29] Philip Morse, "Queues Inventories and Maintenance", (1958).
- [30] Donald Gross, Carl M. Harris, "Fundamentals of Queueing Theory ", Wiley-Interscience , (1997).
- [31] Avi Silberschatz, Peter B. Galvin, Abraham Silberschatz, Greg Gagne, "Applied Operating System Concepts", first edition, (1999).
- [32] Randolph W.Hall, "Queueing Methods For Services and Manufacturing", Prentice-Hall International Editions, (1999).
- [33] 3Dspotlight Web site, "<http://www.3Dspotlight.com>", (May 2002).