

東海大學資訊工程研究所

碩士論文

指導教授：朱正忠 教授

多重服務品質考量的雲端服務選擇基
於基因演算法

Multiple QoS-aware Selection of Cloud
Service based on Genetic Algorithm

研究生：唐東暘

中華民國一〇二年七月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 唐 東 暘 所提之論文

多重服務品質考量的雲端服務選擇基於基因
演算法

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召集人

盧若偉

簽章

委

員

楊朝棟

薛念林

張志宏

指導教授

朱正忠

簽章

中華民國 102 年 7 月 3 日

摘要

近年來雲端計算是以服務導向架構的方式提供服務給使用者使用，在這個架構下，軟體經由網路提供功能性的服務給其他軟體使用，為了實現更多包含許多個別服務的複雜工作或商業程序，我們必須使用服務組合(service composition)的概念去幫助我們完成這個工作，但是如何在眾多功能相近的服務中挑選出最適當的服務組合以維持良好的服務品質 (QoS, Quality of Service) 則是一個挑戰，要達到良好的服務品質必須考慮包含花費(cost)、回應時間(Response time)等多種因素，因此基於 QoS 的服務組合選擇問題是一個 NP-Hard 問題。

本研究探討如何解決服務選擇優化的問題，並提出一個基於基因演算法(GA)的方法，藉由基因不斷的演化過程，找出較優的服務 QoS 組合，以確保服務品質。

關鍵字：服務品質、服務選擇、基因演算法

Abstract

Cloud computing is providing services based on service-oriented architecture in recent years. The cloud application can be implemented by service composition in this architecture. An important challenge is how to select a service for each task involved in a composite service such that the overall QoS for the composite service is optimal. This includes customer focused elements such as response time, cost, reliability and availability. QoS-based service selection is a combination problem and is a kind of NP-Hard problems.

This study investigated how to optimize the selection problems and propose a genetic algorithm-based method which solving the problem by considering many QoS attributes in the gene. By the evolution process, the gene will be more suitable solution for services to ensure service quality.

Keywords: Quality of Service, Service Selection, Genetic Algorithm

致謝

首先要感謝我家人的支持與栽培,讓我能無後顧之憂地完成我的學業,還有感謝張志宏學長對我論文內容的建議與指導、盧志偉學長對我的鼓勵和握手、楊朝棟老師對我的論文鞭辟入裡的想法與建議,最後要感謝我的指導教授 朱正忠教授,謝謝朱老師不厭其煩地教導我做研究的方法與態度,訓練我如何找到問題並解決它,讓我了解研究所與大學的差別;也謝謝朱老師不斷地帶領實驗室參與許多研究計畫,讓我能夠在進入社會前能夠磨練自己的技術以及與其他團隊之間的分工合作,以完成較大的案子,更謝謝朱老師給了我不一樣的視野,讓我更加體認到資訊產業的深度與廣度,最要感謝的是研究所同學和朋友的支持和鼓勵,才能讓我有動力完成這篇論文,研究所畢業只是人生路途小小的一步,未來的路還很長,還有很多需要學習的知識與要做的事,畢業後即將面臨人生下一個轉捩點,我必須更加的謹慎去面對未來的挑戰,把每件事都做到最好。

目錄

摘要	I
Abstract	II
致謝	III
目錄	IV
表目錄	VI
圖目錄	VII
第一章 緒論	1
1.1. 前言	1
1.2. 研究動機	1
1.3. 研究目的	3
1.4. 章節安排	4
第二章 背景知識與相關研究	5
2.1. 雲端服務	5
2.2. 服務組合	8
2.3. 雲端服務的服務品質	10
2.3.1. 服務品質的屬性	10
2.3.2. 服務品質管理	12

2.4. 基因演算法	14
第三章 研究方法	18
3.1. 問題定義	18
3.2. Qos 服務品質指標	19
3.3. 服務選擇	20
3.4. 演算法流程	21
3.5. 染色體編碼	22
3.6. 適應性函數	24
3.7. 族群數	25
3.8. 適應性選擇與世代的再生	26
第四章 實驗與分析	27
4.1. 實驗參數	27
4.2. 突變機率差異設定分析	28
4.3. 服務數量對計算時間的影響分析	29
4.4. 實驗結果	31
第五章 結論與未來工作	33
參考文獻	34

表目錄

表 2-1 雲端服務與商業模式[32]	6
表 2-2 各種 QoS 屬性的定義規範模型	11

圖目錄

圖 1-1 研究流程	4
圖 2-1 BPEL-WS 服務程序流程[25].....	9
圖 2-2 雲端服務架構的 QOS 代理	13
圖 2-3 基因演算法流程(GENERATION-BASED).....	15
圖 3-1 基因編碼	23
圖 3-2 族群數分佈.....	25
圖 4-1 服務實例	27
圖 4-2 最佳 FITNESS SCORE	28
圖 4-3 平均 FITNESS SCORE	29
圖 4-4 不同數量的抽象服務的計算時間.....	30
圖 4-5 產生初始族群.....	31
圖 4-6 演算法結果.....	32

第一章 緒論

1.1. 前言

本論文主要著重於如何將基因演算法(Genetic Algorithm)應用於雲端服務組合的選擇上，並且考慮多重的服務品質(Quality of Service)的限制，基於 QoS 的服務選擇產生了許多問題，這些問題中有許多都是優化問題，本研究利用基因演算法來處理服務選擇的問題

基因演算法是一個模擬自然界生物演化的啟發式演算法，用於解決最優化問題的一種方法，這種啟發式演算法通常用來生成有用的解決方案來優化和搜索問題。

1.2. 研究動機

目前雲端運算偏向以服務導向(Service oriented)架構的方式提供雲端服務給使用者使用，而一套完整的雲端服務系統各個元件可能是由多個分散的雲端服務所構成，這些原本分散的服務藉由介面的溝通與綁定組合成了一個新的服務。

同一功能類型的服務可能有多個選擇並且彼此可以互相取代，在眾多服務中挑選符合需求的服務可以根據服務品質(QoS)的屬性去選擇，有些人可能要的是價錢最便宜的服務，有些人要的是最快的服務，有些則是要介於兩者之間的，因此 QoS 的屬性必須根據使用者的非功能性需求去制定服務指標，服務提供商可根據這些 QoS 屬性制訂服務等級協議(Service Level Agreement, SLA)，保證服務的品質可以滿足使用者的需求。

然而在現實中成功的雲端服務必須去面對一些關鍵的問題，其中一個問題是如何去管理服務組合的服務品質(QoS)問題，因為當提交服務不僅要滿足客戶的功能

性需求也要滿足客戶的 QoS 需求，才能達到客戶的期望與滿足。

最常見的問題是客戶期望他們可以消費使用一個擁有最好 QoS 的服務，例如：客戶希望使用的服務可以有最短的回應時間、最低的成本、最高的可靠性、最高的傳輸流量，也有可能客戶需要多重的服務品質最佳化，多重服務考量的服務組合問題是一個研究如何選擇最佳的服務滿足客戶需求的服務，以得到最好的使用者體驗為目標。

基於 QoS 的服務選擇問題是一個研究如何分配適當的外部服務進而讓整個服務 workflow 達到最好的結果，本質上這是一個規劃問題，這個問題在以下三個方面非常具有挑戰性：

1. 首先，這個問題是具有大型規模的，如何在每個服務組合任務挑選一個適合的服務，假設一個複合服務擁有 20 個任務，而每個任務都有 30 個候選服務，然後就會有 30^{20} 種可能的選擇組合，導致從這些大量的可能種找出最佳的服務選擇決策是非常棘手的。
2. 其次，這個問題面臨了許多限制條件，例如：服務之間有許多服務是必須互相依賴的，也有某些服務是彼此衝突的，為了保證服務可以正常運作，這些限制是必須被考慮進去的，然而也會增加問題的複雜度。
3. 最後，問題可能會有非常多的 QoS 衝突必須處理，例如：客戶希望服務的回應時間越短越好，花費的金額越少越好，然而這兩個期望可能是衝突的，要得到較快的回應時間、較好的效能，相對的花費的成本自然也會提高。

1.3. 研究目的

如何在眾多服務組合間挑選出符合 QoS 服務品質的組合是一個組合優化的 NP-Hard 問題，必須還要考慮各種 QoS 限制條件，本論文提出一個基於基因演算法的方法，藉由基因不斷演化的過程，淘汰不好的基因留下好的基因組合，再進行交配、突變等過程，不斷的重複直到找到最好的基因。

在解決 NP-Hard 等困難複雜的問題方面要求得準確的解答一直缺乏有效率的方法，而啟發式演算法（包含基因演算法 genetic algorithm、模擬退火演算法 simulated annealing、禁忌搜尋演算法 tabu search 等）是目前較為流行的解決辦法，其中基因演算法 GA 由於它本質上的特性是可以較容易進行平行運算的，GA 以字串編碼問題的可能性解答染色體集合，然後模擬達爾文進化論中生物染色體世代的選擇、交配、突變等過程，以適應性函數(fitness function)對染色體的優劣程度進行評分，將較優良的基因留下來繁殖產生下個子世代染色體集合，在這個機制下經過多次的世代後會產生適合的解答。

基於 QoS 的服務組合問題被證實為 NP-Hard 問題[1]，需要有效率的演算法來處理才能保證令人滿意，也許不一定是最佳的解決方案，但是至少是可以接受的。

本研究探討了如何使用基因演算法(GA)，以解決基於 QoS 的服務組合問題，GA 是來自自然演化過程的啟發，演算法的工作原理是模擬演化的過程，如基因的繼承，變異，選擇，交配，最後產生可能的解答，但以優化和搜索問題來說，並不總是能提供最佳的解決方案。GA 已成功地應用在許多問題領域（如電腦自動化設計，電子電路設計，無線傳感器網絡等），用於解決複雜的、大規模的、約束和多目標優化的問題，這些成功的經驗，也鼓勵我們選擇 GA 以解決基於 QoS 的 Web 服務組合具有複雜，規模大，約束和多目標優化問題。本研究的目的是開發有效和可擴展的演算法，基於 QoS 的 Web 服務組合優化，找出可行、可靠和高品質的複

合 Web 服務，以完成客戶交付的目標。

1.4. 章節安排

本論文之架構分為五章，包含「緒論」、「背景知識與相關研究」、「研究方法」、「實驗與分析」以及「結論與未來工作」等部分，其架構流程與大綱如圖 1-1。

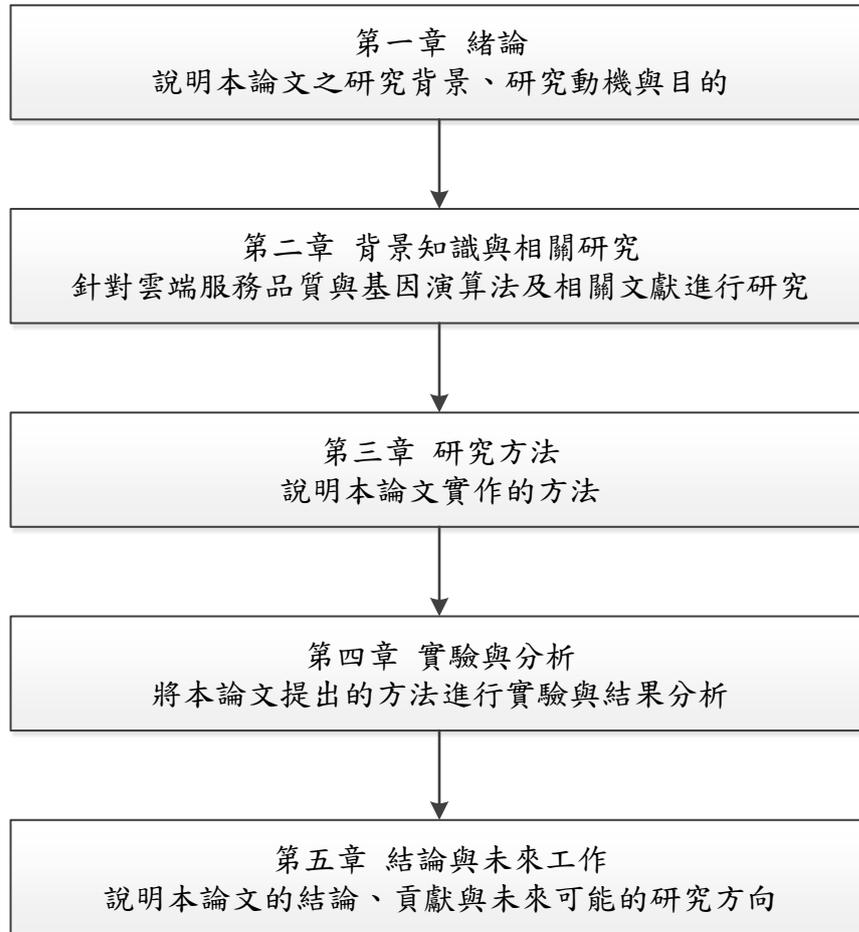


圖 1-1 研究流程

第二章 背景知識與相關研究

2.1. 雲端服務

根據 Gartner 所公布的雲端市場預測報告[2]，2012 年全球公有雲端服務市場產值達 1,090 億美元，較 2011 年成長 19.6%，到了 2016 年預估可成長達 2,006 億美元。其中，基礎架構即服務(Infrastructure as a Service；IaaS)持續成長，預計市場規模將成長 45.4%。此外，IDC 研究報告[3]也顯示企業雲市場持續深化，SaaS 市場升溫，伴隨著電信營運商、國內外軟硬體業者與諮詢服務業者先後投入雲端市場，並透過策略聯盟等方式提供更多元的服務模式，企業對雲端服務的接受度隨之升高。

雖然雲端市場成長可期，有越來越多的企業願意改買為租、依據所需向雲服務運營商(Cloud Service Provider)動態租用虛擬資源，並將公司資訊系統從原先建置於實體機之方式改建置於雲端，以更有效率、更節能的方式提升運算資源使用率並降低設備建置與維護成本。一家成功的雲服務運營商必須先具備完善的雲端服務營運維運管控技術與系統解決方案，才能夠提供給客戶良好的雲端服務使用經驗。因此，舉凡國內外雲服務運營商如 Amazon 或 Savvis，在提供公雲或私雲服務予客戶時，皆會考量以自行發展或外購引進的方式，打造出一套架構參照國際標準且具備擴充能力之營運維運管控系統，來快速、彈性地提供穩定、高品質的雲端服務，及時滿足客戶所需。除此之外，該管控系統亦須能夠充分支持美國國家標準與技術研究院(NIST)所揭示的雲端運算關鍵特性，包括：支援多樣化網路終端接取(Broad Network Access)、將運算與儲存資源予以集中管理與調度(Resource Pooling)、服務應用快速且具備彈性(Rapid Elasticity)、服務可量測管理(Measured Service)、客戶可自主服務並隨需即用各式雲端服務(On-demand)。

美國國家標準和技術研究院的雲端運算定義中明確了三種服務模式：[4]

軟體即服務 (SaaS)：消費者使用應用程式，但並不掌控作業系統、硬體或運作的網路基礎架構。是一種服務觀念的基礎，軟體服務供應商，以租賃的概念提供客戶服務，而非購買，比較常見的模式是提供一組帳號密碼。例如：Microsoft CRM 與 Salesforce.com。

平台即服務 (PaaS)：消費者使用主機操作應用程式。消費者掌控運作應用程式的環境（也擁有主機部分掌控權），但並不掌控作業系統、硬體或運作的網路基礎架構。平台通常是應用程式基礎架構。例如：Google App Engine。

基礎架構即服務 (IaaS)：消費者使用「基礎運算資源」，如處理能力、儲存空間、網路元件或中介軟體。消費者能掌控作業系統、儲存空間、已部署的應用程式及網路元件（如防火牆、負載平衡器等），但並不掌控雲端基礎架構。例如：Amazon AWS、Rackspace。

由於雲端運算的主要精神為服務，故三大服務模式下紛紛衍生諸多商業模式，例：儲存即服務 (Storage-as-a-service)、運算即服務 (Computing as a Service)、資料庫即服務(Database-as-a-service)、測試即服務 (Testing as a Service)、整合即服務 (Integration-as-a-service)、桌面即服務 (Desktop-as-a-service)、流程即服務 (Process as a Service)等等，甚至還出現 XaaS (Everything as a Service) 的說法，相關商業模式內涵說明請參閱表 2-1：

表 2-1 雲端服務與商業模式[32]

資源配置層次	服務名稱	定義及介紹
IaaS	運算即服務 (Computing as	為具快速部署、彈性、按需求擴展的運算服務。實例：目前國內相關產品如國網中心與華碩打造的新一代高

	a Service)	效能叢集運算系統 Formosa 4。
	儲存即服務 (Storage-as-a-Service)	讓使用者在網路上可以擁有一個自助式管理和存取的儲存空間。 實例：中華電信推出仿 Dropbox 的雲端資料儲存「CloudBox」服務，除提供資料備份外，更特別強化安全性能。
PaaS	資料庫即服務 (Database-as-a-Service)	提供遠端託管的資料庫服務，與其他用戶共同使用，但在邏輯上就像是本地的資料庫一般。 實例：微軟的 SQL AZURE、VMware 的 vFabric Data Director 資料庫即服務平台。
	整合即服務 (Integration-as-a-service)	整合即服務整合其他資源形成解決方案，從本質上說，其包括了傳統企業應用整合(EAI)技術中的大部分特性和功能，不同的是它以服務的形式提供。 實例：HP CloudSystem 的整合式系統外，此外另有法國開放源碼整合公司 Talend 所發表的整合統一平台的第 5 版 Talend v5；而國內的整合服務多以垂直性軟體整合為主，如：華碩推出的「雲端平台+應用軟體+伺服器」整合方案。
SaaS	桌面即服務 (Desktop-as-a-service)	是將個人電腦的桌面從實體機器中切分開來，並將此虛擬桌面儲存在遠端電腦中。 實例：資策會與廣達電腦、迎廣合作推出的「桌面雲」服務。

	流程即服務 (Process-as-a Service)	流程即服務為讓用戶為自己的商業流程建立模型，少數業者甚至提供完整的流程開發與執行功能。 實例：Salesforce 的 CRM 軟體，國內產品如五十五資訊網路科技股份有限公司推出的 GROUPs Cloud EIP。
--	------------------------------------	---

除了上述雲端服務外，另外，尚有醫療雲、教育雲、行動雲、家庭雲以及交通雲等等以行業別為基礎或具特定目的而建構的雲端服務，以雲端服務為架構的應用越來越多，因此如何維持良好的服務品質，讓操作時感受與桌面應用無太大的差別，帶來良好的使用者體驗(User Experience)也是相當重要的。

2.2. 服務組合

雲端服務的一個吸引人的特性是現有的服務可以整合在一起，創建新的價值服務以滿足使用者的需求，整合建立新的服務的過程稱為服務組合[28]。

服務組合的用意是取得現有的服務，並且結合這些服務形成新的服務能力，也就是將現有的服務加以組合，可以解決更複雜問題的服務這也讓服務可以應用在企業流程。以流程為基礎的網路服務組合，是一種新興的服務方式，可運用在組織內，或跨組織的自動化企業流程方法。

服務組合對於系統開發來說，是很重要的變革。傳統的系統開發是從無到有，必須依據系統開發流程，從需求訪談、系統分析與設計開始，一個元件一個功能辛苦地開發，而現今的方式是藉由現有服務的搜尋，配合特殊需求功能的服務開發，共同組合成一個新的系統。這樣的開發方式不僅可以減少系統開發的時間，並且可以增加網路服務的可再利用性。

依據使用者的需求,產生和執行組合計劃的能力,這樣的系統所提供的價值，明顯地超過個別網路服務所提供的價值，因此網路服務組合的必要性，在於它不僅可

以將企業的流程，依據新的需求自動化產生，並且能快速地提供服務，一個服務組合的程序流程如圖 2-1 所示，其中的規範定義了結構化的活動來管理整個流程以及基本的活動，包括本身與服務互動的過程。

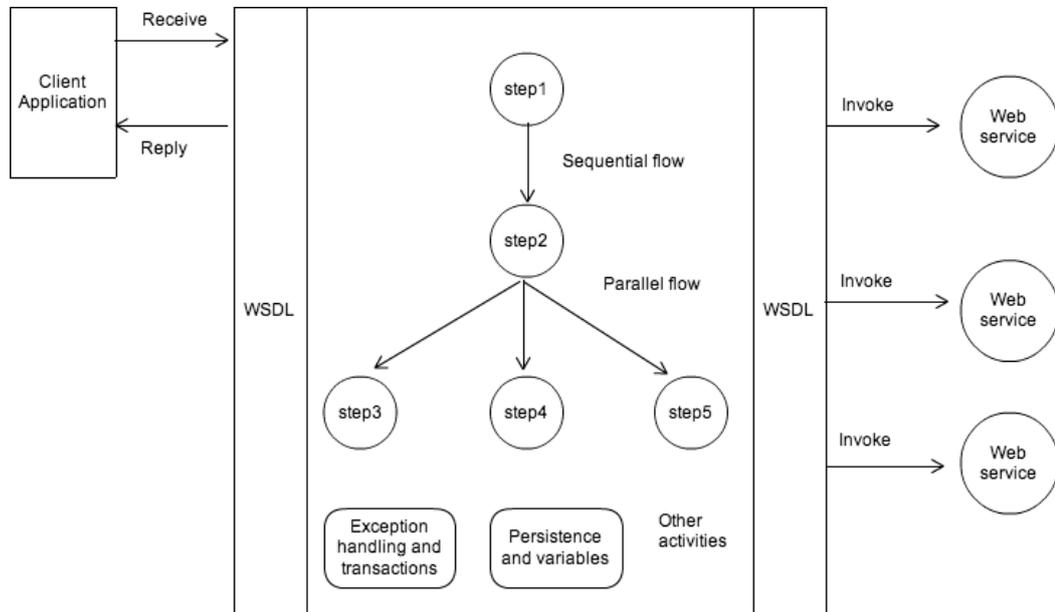


圖 2-1 BPEL-WS 服務程序流程[25]

大多數基於 Web 的雲端服務應用必須透過服務組合來建立，在開發雲端服務的應用程式的時候，考量 QoS 的服務組合是必須的，Jager[24]在他的論文中已經確定基於 QoS 的服務組合主要的步驟如下：

1. 服務組合的設計：工作流程的設計者使用服務組合定義語言(如：WS-BPEL[25])定義服務組合的工作流程，在這個步驟產生的結果是產生一個使用特定複合服務的服務組合程序，一個簡單的例子如圖 2-1 所示，其中一個組合服務包含由特定的商業邏輯所定義排序後的抽象服務集合。
2. 服務的發現：一個服務代理處理在這個服務組合中每個抽象服務集合具體服務的發現。
3. 服務品質考量的服務選擇：根據客戶的 QoS 需求，這個步驟包含對於每個抽

象服務集合根據設定的規格選擇具體的服務，在這個階段將產生可執行的服務組合規劃。

4. 綁定服務並執行：根據產生的服務執行規劃，這個步驟包含綁定所選擇的具體服務並且執行服務組合，傳統的服務綁定發生在設計階段，這並不利於進行 QoS 考量服務組合，客戶的需求通常會在提出服務請求的時候才會知道，如果的設計階段就綁定服務的話並不一定可以保證服務品質的需求，拜最新的服務綁定技術[26]所賜，服務的綁定可以推遲到執行階段，這使得服務組合可以更有有彈性，更可以滿足客戶的需求。

雲端服務的組合與選擇是一個建立基於服務導向的應用程式必要的過程，導入 QoS 的雲端服務組合面臨了許多挑戰，例如：如何建立服務的 QoS 模型和設計 QoS 的即時監控(Monitoring)框架，進一步的挑戰是如何選擇服務去組成完整的服務，並考慮最佳的 QoS 因子，後者面臨的挑戰是所謂的基於 QoS 的服務選擇問題，在現實中，基於 QoS 的服務選擇通常不是一個單機操作，選擇應該考慮服務之間的倚賴性和衝突性等種種限制條件以保證其正確性，Zeng [8]在研究中提出了使用整數線性規劃的方法去解決這個問題，Yu 和 Lin[27]則提出了基於動態規劃的演算法去解決這個問題，也有學者用基因演算法(GA)[29][30]去克服這個問題。

2.3. 雲端服務的服務品質

2.3.1. 服務品質的屬性

網路上的雲端應用程式必須藉由服務組合的方式來達到某些功能性的需求，相反地這些應用程式也必須達到某些非功能性的需求，這種需求稱之為服務品質(Quality of Service)。舉例來說：對於每個使用者呼叫(Request)，一個應用會產生回應時間(Response Time)，並且每個呼叫都會花費一些成本(cost)，該回應時間的長

短與花費的金額與其他服務品質的屬性的綜合結果直接地反映了使用者會對於該服務的滿意程度，這些因素將會影響雲端服務是否成功，因此當遞送服務給使用者時，我們必須考慮的不僅是服務的功能性，也要考慮服務品質。為了達到此目的，一個重要的需求是要有良好的服務品質管理，因此必須使用適合且有效率的方法從眾多服務中挑選出適合的服務組合以滿足使用者的需求與體驗[5]。

第一件必須做的事是定義各種服務品質的屬性，雲端服務品質的概念主要是在網路、即時性、中介軟體等方面的屬性，在過去的研究中多注重如何定義服務品質的模型，目前還沒有任何標準技術去定義服務品質模型，不同的組織和研究者使用不同的服務品質分類去定義，三種常見的服務品質定義模型如表 2-2 所示。

表 2-2 各種 QoS 屬性的定義規範模型

UML QoS-Profile[5]	Ran's QoS Model[7]	Zeng's QoS Model[8]
throughput	throughput	throughput
latency	response time	response time
efficiency	cost	cost
availability	availability	availability
reliability	reliability	reliability
security	security	reputation

雖然三種定義模型中任何一個都提供了服務品質屬性的綜合性描述，但是只有 Zeng[8]所提出的模型涵蓋了如何從服務品質加積公式去計算服務組合的品質，這是在討論服務品質時最基本的，目前 Zeng 提出的加積公式已經廣泛地應用於注

重服務品質的服務組合的領域。

在服務品質的屬性的含義包含以下幾點：

- 反應時間－反應時間表示量測的預期延遲，也就是發送請求與接收到回傳結果的經過時間。
- 花費－花費表示呼叫使用該服務所需要的金額。
- 可靠性－可靠性表示服務功能正確與一致的能力並且提供同樣的服務儘管系統或網路故障的時候。
- 可用性－可用性表示服務可以使用的比率。
- 聲譽－聲譽表示一個服務的可信賴程度，主要來自最終用戶使用該服務的體驗。
- 吞吐量－吞吐量表示某段時間服務可被提出呼叫的承載量。

本論文綜合以上三種 QoS 模型，制定符合大部分雲端服務所需要的 QoS 服務指標，將於第三章中詳細說明。

2.3.2. 服務品質管理

當已經選擇了服務品質的屬性後，現在的問題是當遞送服務的時候如何去保證這些服務指標，所以需要一個有效的服務管理方法去幫助管理，過去的研究主要範圍是在支援服務品質服務提供的服務探索框架[9][10][11]，並以此發展服務品質相關的技術[12][13][14][19][20]。

關於服務品質的框架在先前的研究中就已經被提出來，最常見的是基於服務代理的架構如圖 2-2 所示，在這個架構中，在服務可以被使用之前，服務的提供者需要先發佈服務，當服務提供者送出一個註冊的請求(如圖 2-2 中的 P1 所示)時，

也必須提供服務品質的資訊，服務代理才能允許這個註冊請求。

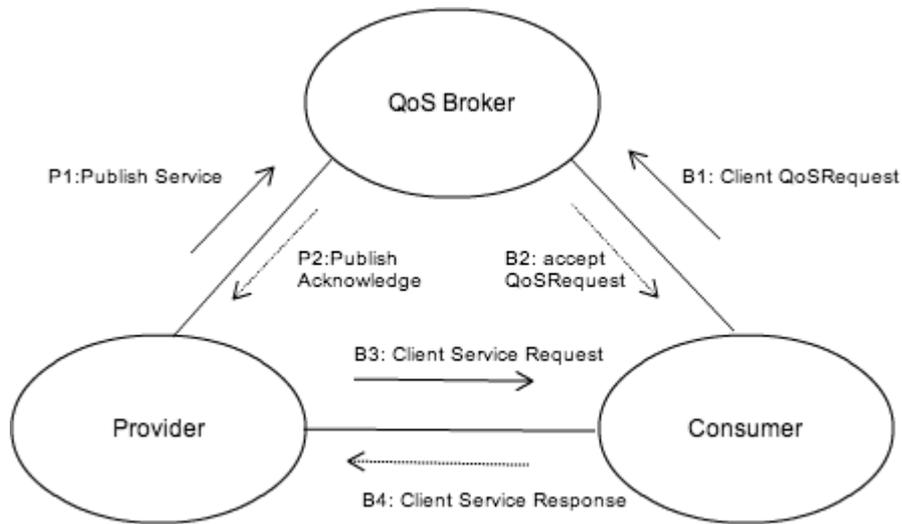


圖 2-2 雲端服務架構的 QoS 代理

整個服務的執行過程包含下列四個步驟[21]:

1. 一個客戶發出一個請求，包含服務品質需求資訊
2. 一個 QoS 代理代表服務提供者與顧客協商 QoS，這個協商可能導致 QoS 請求被接收、拒絕、或採取折衷方案
3. 當服務協議被接受後，客戶可以開始使用該服務
4. 服務提供者以保證服務品質的方式提供服務

這樣的一個 QoS 框架提出了許多技術問題，以下總結了主要的研究問題和這些問題的解決方案。

第一個問題必須加以解決的是如何讓 QoS 的資訊可以在各方交換(即服務提供者、QoS 代理和客戶)，許多的 QoS 規範語言被提出來解決這個，其中典型的應用包括 IBM 的 Web 服務登記協議(WSLA)語言[12]，HP 的 Web 服務管理語言(WSML)[22]等，藉由這些 QoS 規範語言，客戶、服務代理商、服務提供商可以溝

通交換彼此的 QoS 需求。

服務的註冊和發現也是必要的，許多研究者已經開發出 QoS 代理去擔任 UDDI 資源庫，以支持基礎服務的註冊與發現[14][23]，這件事可以進一步分為發展一個與 UDDI 元件庫整合的 QoS 代理以及一個與 UDDI 元件庫分離的 QoS 代理，無論什麼類型的 QoS 代理，都負責建立客戶與服務提供商之間的 QoS 互動。

為了保證組合後的服務交付給用戶後有當初承諾的 QoS 品質，有效率的服務配置機制也是必要的，不過這個問題目前還沒有得到很好的解決，也是研究的重點之一，QoS 的監控機制也是必要的，必須去監控所有可用的資源提供以及客戶端的需求。

2.4. 基因演算法

基因演算法最原始的概念源自於 1859 年達爾文(Charles Darwin)在物種源起(The Origin of Speices)一書中對於演化的理論的「物競天擇，適者生存」觀念，而在 1975 年美國密西根大學 J. H. Holland 與他的同事及學生發表的研究[15]中首次被提出，其精神是以模擬生物演化過程中擇優汰劣的自然法則，保留母代的優良特性，一代一代找出適應性最強的子代，從而得出最佳解。

基因演算法適合用來解決全域最佳化問題(Global Optimization Problem)，例如時間表排程的問題等。基因演算法的基本流程如圖 2-3[16]，步驟如下：

- STEP 1 產生初代族群(Population)
- STEP 2 針對初代族群的每一個體計算其適應性
- STEP 3 根據交配(Crossover)或稱交叉操作產生下一代
- STEP 4 根據突變(Mutation)機率產生突變的個體

STEP 5 計算此世代每一個體的適應性

STEP 6 測試是否滿足終止條件，若滿足則結束，否則回到 STEP 3

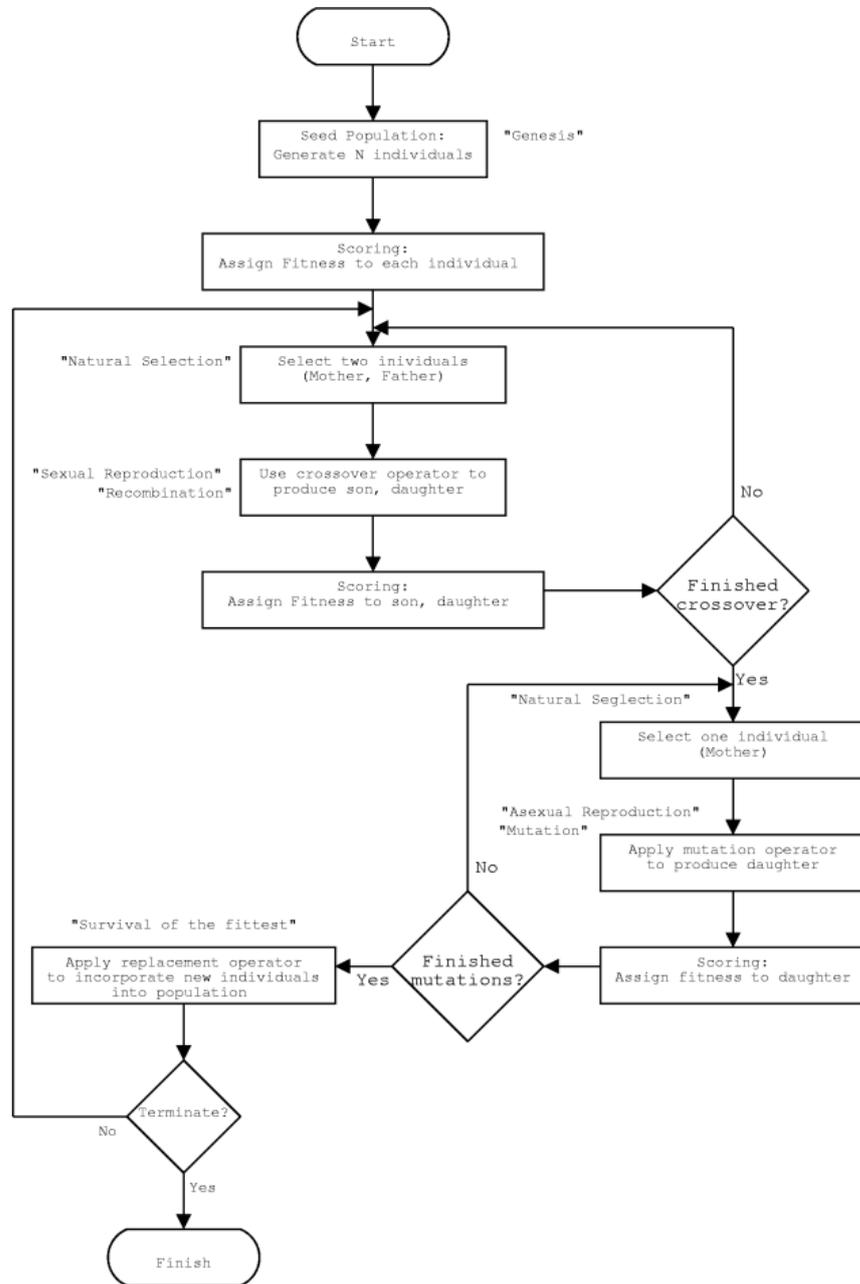


圖 2-3 基因演算法流程(Generation-based)

族群是染色體(Chromosome)的集合，而染色體是由基因所組成，針對問題設定一適性函數，用來評估各個體對於問題的適應性，也就是每個解對於該問題適

合的程度，並且對染色體做編碼產生母群體，接著進行複製、交配與突變等操作產生新的族群，並對每個個體評估其適應性與測試是否達到終止條件。

基因演算法的參數說明如下：

- 編碼(Encoding)：編碼型式依不同問題而有所不同，可依問題的類型與結構做選擇，常見的編法方式有二進位法、整數編碼、實數編碼等。
- 族群數(Population Size)：族群數會影響結果的產生，若設定一較小的值，雖然搜尋速度快，但會導致過早收斂，可能只得出區域最佳化的解而忽略了全域更好的解；反之，若設定較大可以在更大的物種多樣性中尋找解答，但會需要更多的計算時間。
- 交配機率(Crossover Probability)：介於 0 到 1 之間，用以控制不同基因組合的個體產生的速度，若太大會收斂過快，陷入區域最佳解；反之，則需要更長久的繁衍才會收斂。
- 突變機率(Mutation Probability)：介於 0 到 1 之間，但一般來說都設得很小，因為設太大與隨機選取無異，用以防止可能更優良的基因組合無法進入族群。
- 適性函數(Fitness Function)：用來評估各個體的適應程度，透過適應程度的差異淘汰次佳個體，加速搜尋最佳解的依據。在不同的問題結構與特性下，可能需要尋找在某一範圍內之最大化或最小化當成解的適應性佳的趨勢。
- 終止條件(Termination Condition)：用以結束演化循環的控制條件，實作中，繁衍終止的條件很多，例如達到繁衍世代次數限制、繁衍的後代適應性達到事先訂定的標準或是繁衍出來的後代適應性皆收斂至同一水

準而得到最佳解等。

而產生新的子代族群時有「整代取代」或是利用「菁英政策[17]」挑選，菁英政策基因演算法[18]是在產生子代時，挑選出適應性較佳的個體組成新的子代族群，加速搜尋最佳解。

本研究將探討如何使用基因演算法(GA)，以解決基於 QoS 的服務組合問題，基於 QoS 的 Web 服務組合優化，找出可行、可靠和高品質的複合 Web 服務，以完成客戶交付的目標。

第三章 研究方法

3.1. 問題定義

如上面 2.2 章節中所提到的，多重服務品質的服務選擇關係到組合後是否能保證服務品質的成功與否，選擇的結果直接決定了組合後的服務的 QoS，因此如何選擇適合的組合是非常重要的。

這個問題是典型的組合優化問題，本研究將問題定義為：

- (1) 一組服務組合內的抽象服務或工作的集合 $A\{A_1, A_2, \dots, A_n\}$ ， n 為在這個組合內的抽象服務數量。
- (2) 在抽象服務 A_i 中擁有不同數量的具體服務 $C_i\{C_{i1}, C_{i2}, \dots, C_{im}\}$ ， m 為在抽象服務 A_i 中擁有的具體服務數量。
- (3) 每個具體服務擁有自己的 QoS 指標數值，包含回應時間、花費價格、可靠性、可用性等。
- (4) 每個 QoS 指標的權重數值不同，
- (5) 組合後的服務可以根據這些 QoS 指標數值與 QoS 指標權重以及使用者的需求求出較優的服務組合

3.2. QoS 服務品質指標

QoS 品質指標代表服務某方面非功能性的質量屬性，本研究參考表 2-2 各種 QoS 屬性的定義規範模型，在 UML 的 QoS profile 中缺少了花費(cost)這項指標，但是考慮到在雲端服務模式下通常是以量計費或者用租賃服務的方式，因此 cost 是必要的，而安全性(security)也是雲端的必須考慮的一個重要的因素，因為客戶會非常在意機敏資料是否遭到洩漏，但是吞吐量(throughput)通常是特定的應用必須同時容納大量使用者的同時存取才必須考慮，綜合各點決定使用以下六個服務指標定義服務品質：

1. 費用(Cost, C)

費用反映服務使用的經濟成本，為提供商給出的使用該服務所需的價格，單位為元，負指標。

2. 反應時間(Response Time, RT)

反應時間反映服務的運作速度和需要使用者等待的時間，包括網路延遲和執行時間，透過呼叫服務的歷史資料所得到結果的平均時間計算，單位為秒，負指標。

3. 可靠性(Reliability, R)

可靠性反映服務的可靠程度，透過呼叫服務後在規定的時間內成功執行的次數佔總呼叫次數的比例計算，無單位，正指標。

4. 可用性(Availability, A)

可用性反映服務能夠被正常呼叫的機率，透過在過去一段時間內服務可用的時間佔總時間的比例計算，無單位，正指標。

5. 安全性(Security, S)

安全性反映服務的安全程度，透過安全級別（0~10）去衡量，無單位，正指標。

6. 企業名聲(Corporate Reputation, CR)

反映使用者對該服務提供商的滿意程度，透過用戶對服務的評分的平均值計算，無單位，正指標。

3.3. 服務選擇

一個任務需要多個不同服務類型去組成，我們必須考慮服務的 QoS 屬性，從候選的服務清單中找出最符合需求的選擇，如果只有一個單一的條件必須被考慮，那麼很簡單的只要找出條件最好的成員即可；而當有多重條件必須被考慮時，這個問題變成了多重背包問題(multiple choice knapsack problem)，這是一個 NP-Hard 問題，Zeng [8]在研究中提出了使用整數線性規劃的方法去解決這個問題，Yu 和 Lin[27]則提出了基於動態規劃的演算法去解決這個問題，而本研究將使用基因演算法(GA)來解決這個問題。

3.4. 演算法流程

本研究中基因演算法應用於組合選擇優化的概念主要如下：將個別可能會使用服務類別以字串組合的方式表示成基因編碼，基因的族群表示一群候選服務組合的集合，適應性代表對於基因優劣量測後的屬性，用於之後的演化過程篩選，主要是以服務組合的 QoS 指標條件屬性計算適應性，以 QoS 為準則的服務選擇演算法過程如下：

1. 產生初始族群：以亂數的方式產生隨機編碼的基因初始族群，基因編碼為可能的服務組合索引字串形式，根據編碼的組合 QoS 屬性對每個基因計算適應性。
2. 選擇：根據基因適應性的分數對族群排序，把分數較低的基因移除，留下適應性分數較高的基因。
3. 交配：從經過選擇後適應性較高的族群中，隨機挑選任意兩個不同的父母基因，進行基因的交配，產生新的基因組合直到達到需要的族群數量為止。
4. 突變：將交配產生的新基因進行突變，根據設定的突變機率隨機找尋要突變的部分進行基因突變。
5. 疊代：反覆的進行以上選擇、交配、突變的動作產生新的子代，直到滿足收斂條件為止。

演算法流程如下所示：

Algorithm : A QoS-based genetic algorithm for the service selection problem

1. initialize population with random candidate solutions
2. evaluate the fitness of each individual based on the fitness function
defined in Equation 3.6 which contains QoS attributes
infeasible individuals
3. **while** termination condition is not true **do**
4. select fit individuals for reproduction
5. probabilistically apply the crossover operator to generate offspring
6. probabilistically apply the mutation operator to offspring
7. evaluate the fitness of each individual based on the fitness function
defined in Equation 3.6
8. **end**

3.5. 染色體編碼

為了將染色體表示成一組問題的解，首先必須將問題編碼為染色體基因組合，本研究中以整數陣列編碼代表染色體組合，以 0-based 陣列的索引代表服務種類的順序，內容的值代表該項服務在該服務種類的索引編號，根據此索引編碼規則對應服務組合。

以下列圖 3-1 基因編碼 chromosome 為例子：

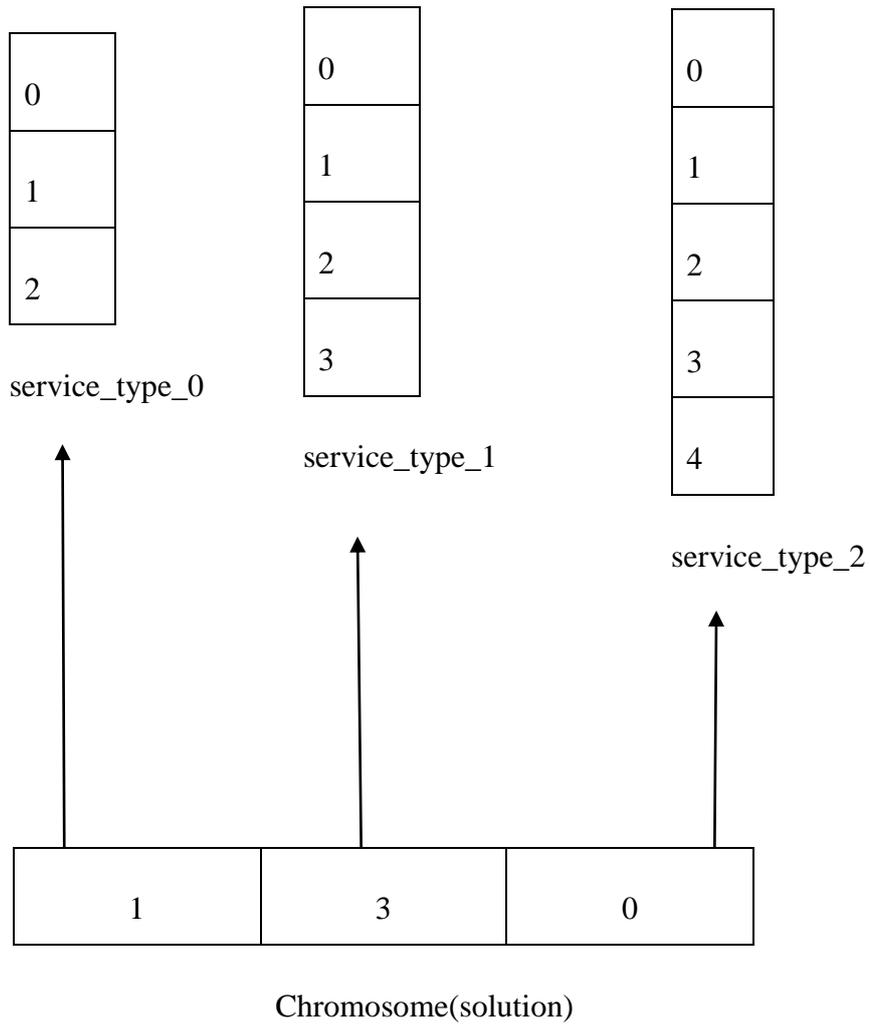


圖 3-1 基因編碼

有三個服務類型(service_type0,service_type1,service_type_2)，分別擁有不同個數的服務，染色體編碼為： $\{1,3,0\}$ ，index 0 的數值對應到 service_type_0 編號 1 的成員，以此類推。

3.6. 適應性函數

根據前面定義的 QoS 指標定義 fitness function，用來評估解的優劣，並以此為依據判斷個體是否被淘汰，而當問題存在某些限制條件、或者服務存在相依性或者衝突性時，必須加入新的 fitness function 判斷是否符合或者不符合限制條件，若不符合條件，則從族群中移除該個體。

本研究單純以 QoS 指標條件進行尋找全域式的最佳化，計算出來的值越大，代表該個體的適應性越好，服務組合擁有更好的服務品質。

$$f(c) = \frac{\omega_A \prod_{i=0}^{n-1} A_i + \omega_R \prod_{i=0}^{n-1} R_i + \omega_S \sum_{i=0}^{n-1} S_i + \omega_{CR} \sum_{i=0}^{n-1} CR_i}{\omega_C \sum_{i=0}^{n-1} C_i + \omega_{RT} \sum_{i=0}^{n-1} RT_i} \quad (3.6)$$

上列式子是對於有 N 種服務的服務組合進行適應性函數計算，對於可用性(A_i)與可靠性(R_i)屬性做乘積，安全性(S_i)、名聲(CR_i)、花費(C_i)、回應時間(RT_i)等屬性做加總，暫時不考慮服務之間的相依性或衝突性的計算方式，以正指標的屬性放在分子，負指標的屬性放在分母，並乘上使用對於不同 QoS 的權重(Weight) ω_i ($\omega_i \geq 0$)，求出來的值越高代表 QoS 品質越好。

3.7. 族群數

族群數會影響尋找最佳解的搜尋或是否因為太小而陷入區域最佳解，因此必須考慮族群數對於運算時間所造成的影響，規模較小、較簡單的問題，族群數若設得太大與窮舉法無異，雖然可能在較少的演化世代數得到最佳解，但會造成搜尋長度太大而降低演算法效能，若設得太小會造成個體多樣性不足，容易陷入區域最佳解；而規模較大、較複雜的問題，族群數若設得太小，也需要演化更多世代才可能找到最佳解，而設得太大更加使得演算法效能低落。假設有 N 種服務類型，每種服務類型擁有的服務數量不一定，而數量總共有 M 個，最佳的族群數設定並非為一個定值，因此本研究中的族群數設定為：

$$\text{Population Size} = \lceil \sqrt{N \cdot M} \rceil \quad (3.7)$$

此式用以反應問題的規模大小與複雜程度，使得基因演算法可以不失其精神且不會造成演算法效能的過度影響，族群數大小分佈如圖 3-2 所示。

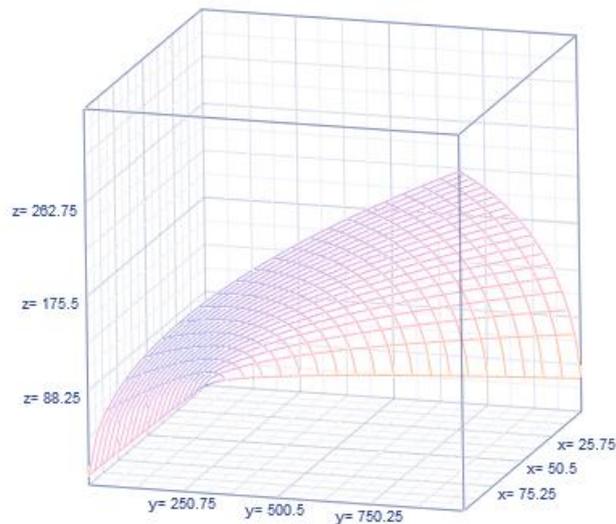


圖 3-2 族群數分佈

3.8. 適應性選擇與世代的再生

在適應性比例選擇 (fitness proportionate selection) 的第一個步驟是根據染色體的優劣分數 (score) 去做降冪排序，之後根據分數比重 (weight) 大於某個值進行決定此染色體組合是否存活下來。

存活下來的染色體組合經過交配和突變後再達到相同的族群大小，並反覆的執行產生新子代，直到產生一定數量的子代 (達到收斂條件) 才停止。

經過不斷的將染色體族群進行分群篩選較優良的存活下來，並進行交配、突變等重複動作，當達到設定的收斂條件 (產生第幾個世代、達到某個適應性分數) 後，程式才停止運行，最後會得到一組適應性最高的服務組合的基因編碼，根據編碼的值對應到具體的服務。

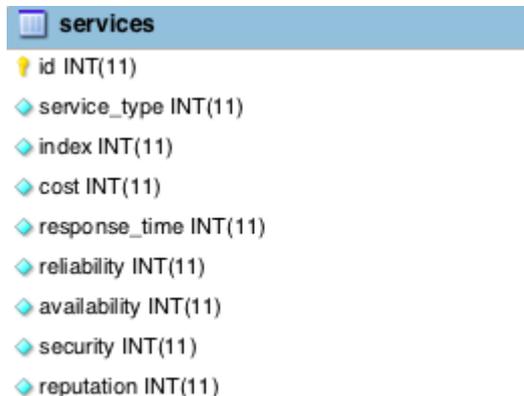
第四章 實驗與分析

由於一般在做基因演算法求解時，族群數、交配機率與突變機率等參數的設定上沒有一定規則，即使在設定上沒有特別針對個別問題做設定，依然可以找到所需要的解，故於第四章中針對本研究之問題設定族群數以及交配機率，以期能夠有更好的效能。

本研究將不同種類的服務定義反應時間、可用性、花費等 QoS 指標數據，可透過此演算法計算服務等級，找出最佳的服務組合，針對不同的實驗參數進行演算法的效能評估。

4.1. 實驗參數

在模擬實驗中，首先根據不同的服務類型隨機生成的許多服務實例，實例中包含了服務的各種 QoS 屬性如圖 4-1 所示。



services	
id	INT(11)
service_type	INT(11)
index	INT(11)
cost	INT(11)
response_time	INT(11)
reliability	INT(11)
availability	INT(11)
security	INT(11)
reputation	INT(11)

圖 4-1 服務實例

基因演算法中定義的參數包含下列幾個參數：

- 服務種類(variety of services)-服務種類代表抽象化服務的數量，每個抽象化服務中擁有不同數量的具體服務與 QoS 屬性。
- 族群大小(Population size)-族群大小代表每次世代產生的族群數量
- 染色體存活率(Proportion of survivors in each generation)-染色體存活率代表每

次進行適應性選擇時存活下來的個體佔原本總個體數量的比例。

- 交配機率(Crossover chance)-交配機率代表有多少個體會進行交配。
- 突變機率(Chance of random mutation)-突變機率代表個體基因中某段基因可能會突變的機率。
- 世代數(Number of generations)-世代數代表總共要產生的世代數量。

4.2. 突變機率差異設定分析

基因演算法在運算時間和產生的解答好壞取決於服務選擇問題的大小與複雜度，而問題的大小是由兩個參數決定：(1)在工作流中的抽象服務的數量(2)每個抽象服務所擁有的具體服務數量，當數量越大，問題的複雜度也越大。

實驗中設定 10 組不同的抽象服務種類，每個抽象服務擁有不同數量的實體服務，而每個實體服務也具有各種不同的 QoS 屬性，根據不同的突變機率執行後，如圖 4-2、圖 4-3 所示。

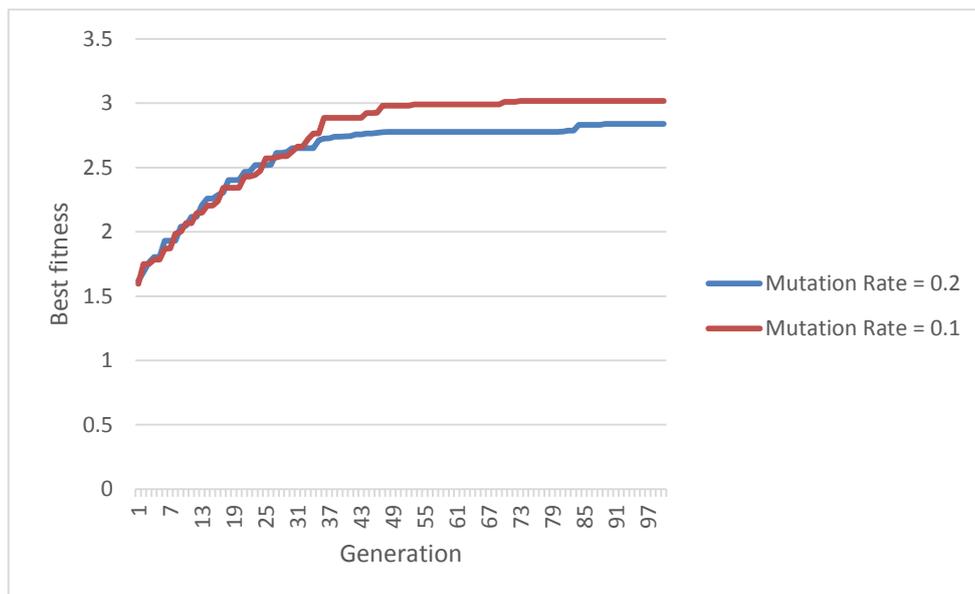


圖 4-2 最佳 fitness score

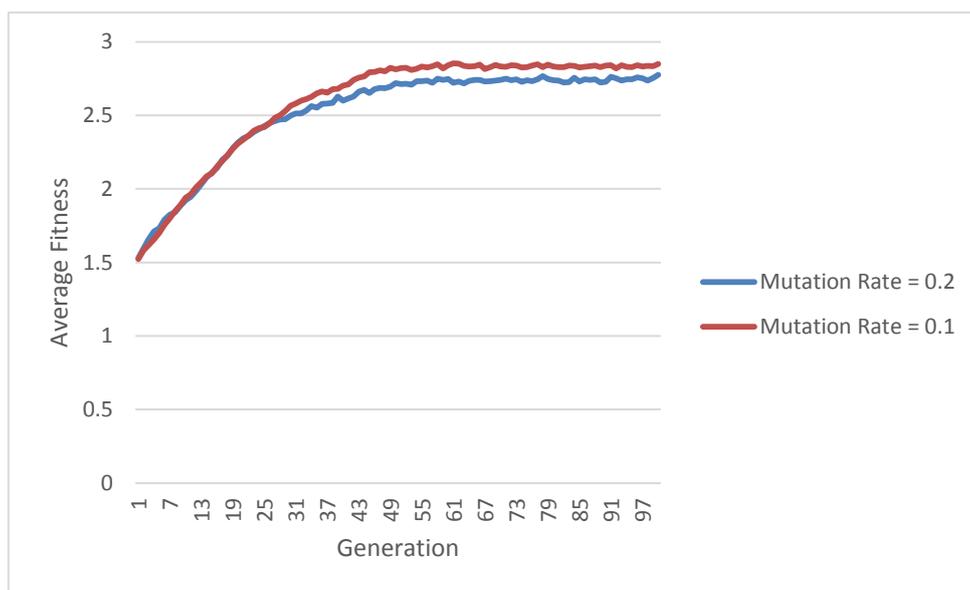


圖 4-3 平均 fitness score

從實驗數據中可以看出經過世代的演化後，族群組合的 fitness score 呈現上升的趨勢，代表路徑解答逐漸呈現最佳化，曲線逐漸收斂的範圍值與設定條件的最佳解的區間符合，而在本演算法在計算服務選擇的問題時，突變機率 0.1 明顯效率優於突變機率 0.2 的設定，是由於突變機率越大時，可能會改變的基因片段越多，帶來較差的結果導致，但是針對不同的案例或問題規模，比較小的突變機率不一定是最好，必須針對案例去調整適合的值。

4.3. 服務數量對計算時間的影響分析

本實驗中設定 10 組不同的抽象服務類型，分別有 10、20、30...100 種服務類型，每種服務類型擁有不同數量的實體服務與 QoS 屬性，測試不同數量的服務對於演算法效能的影響，詳細運算時間如圖 4-4 所示。

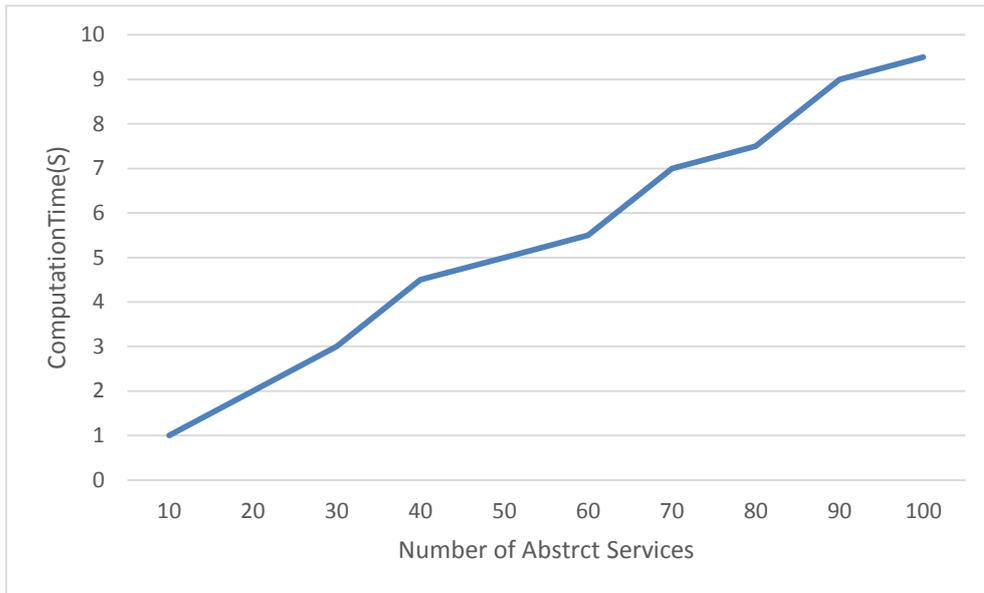
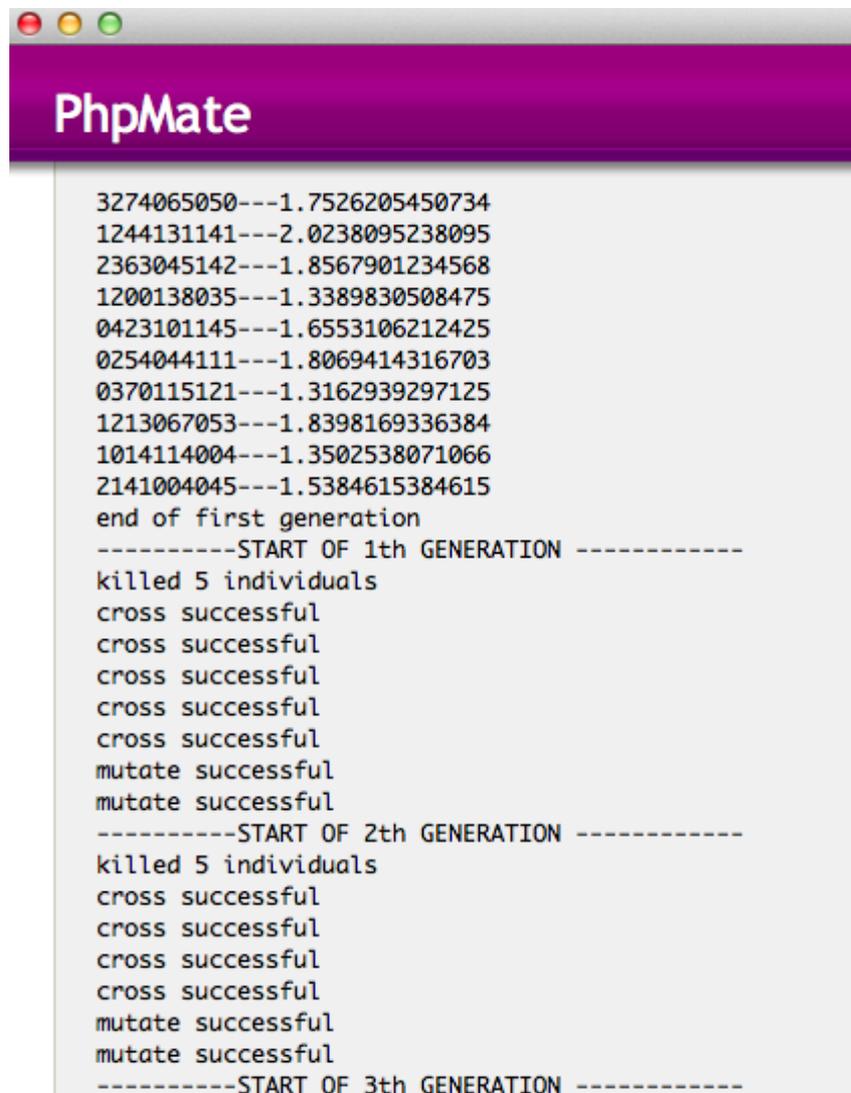


圖 4-4 不同數量的抽象服務的計算時間

當抽象服務類型越多，或者每一個抽象服務所擁有的具體服務數量越多時，所花費的計算時間相對地也越長，為線性成長的曲線分佈，在基因演算法的族群數設定上，如 3.7 節中所述是以 $\sqrt{N \cdot M}$ 根據不同數量的服務動態地調整，不會發生問題規模不大，但是卻因為產生的族群數太多，導致運算時間變長的情形發生。

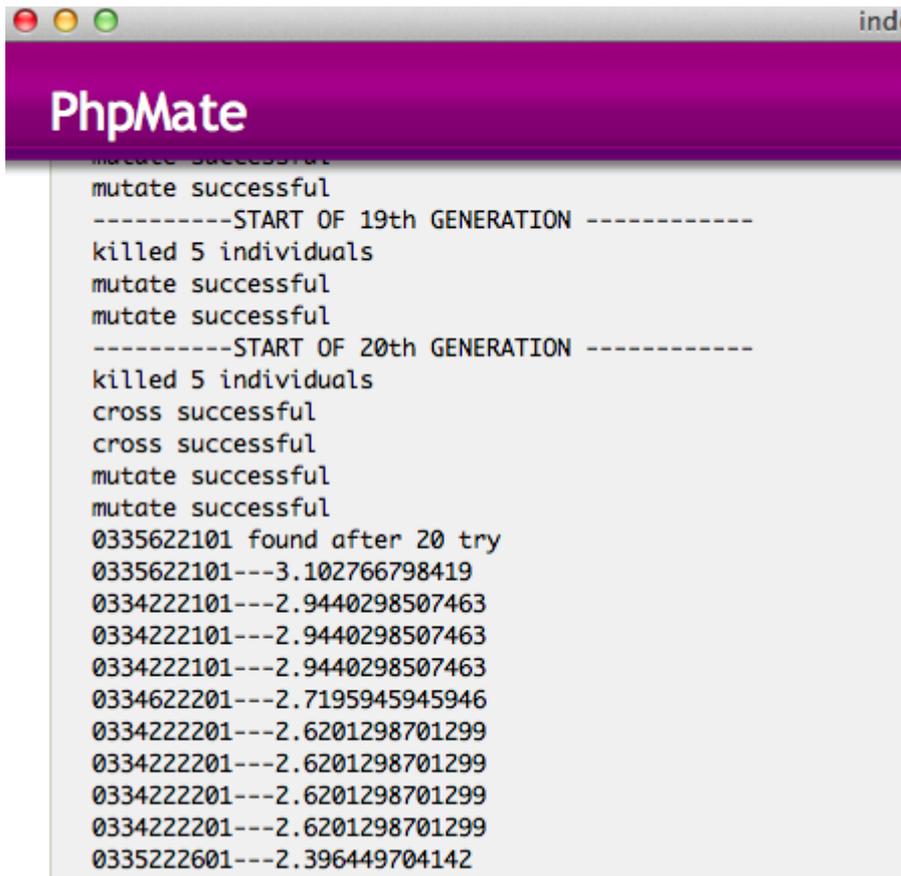
4.4. 實驗結果

經過了多次的世代交替，重複地進行選擇、交配、突變等過程，找到符合 QoS 需求的服務組合，以 10 個抽象服務類型為例子，首先會亂數產生初始族群如圖 4-5 所示並計算其適應性分數，排序後將後面半數的成員從族群中移除，選擇留下前半段的優良成員，從留下的成員中挑選出某幾個成員進行交配及突變，產生新的子代，一直重複到滿足收斂條件為止，如圖 4-6 演算法結果所示，本例子中是設定演算法找到 fitness score 大於某個值後才停止。



```
3274065050---1.7526205450734
1244131141---2.0238095238095
2363045142---1.8567901234568
1200138035---1.3389830508475
0423101145---1.6553106212425
0254044111---1.8069414316703
0370115121---1.3162939297125
1213067053---1.8398169336384
1014114004---1.3502538071066
2141004045---1.5384615384615
end of first generation
-----START OF 1th GENERATION -----
killed 5 individuals
cross successful
cross successful
cross successful
cross successful
cross successful
mutate successful
mutate successful
-----START OF 2th GENERATION -----
killed 5 individuals
cross successful
cross successful
cross successful
cross successful
mutate successful
mutate successful
-----START OF 3th GENERATION -----
```

圖 4-5 產生初始族群



The image shows a screenshot of a PhpMate application window. The window title bar includes standard macOS window controls (red, yellow, green buttons) and the text "ind". The main content area has a purple header with "PhpMate" in white. Below the header, the application displays a list of text-based output from an algorithm, including generation markers and numerical data points.

```
mutate successful  
mutate successful  
-----START OF 19th GENERATION -----  
killed 5 individuals  
mutate successful  
mutate successful  
-----START OF 20th GENERATION -----  
killed 5 individuals  
cross successful  
cross successful  
mutate successful  
mutate successful  
0335622101 found after 20 try  
0335622101---3.102766798419  
0334222101---2.9440298507463  
0334222101---2.9440298507463  
0334222101---2.9440298507463  
0334622201---2.7195945945946  
0334222201---2.6201298701299  
0334222201---2.6201298701299  
0334222201---2.6201298701299  
0334222201---2.6201298701299  
0335222601---2.396449704142
```

圖 4-6 演算法結果

第五章 結論與未來工作

本論文探討了多重服務考量的雲端服務選擇問題，根據具體服務本身的 QoS 屬性，透過基因演算法找出符合使用者需求的服務組合，主要的貢獻是設計了一個有效率的方法，可以幫助使用者在眾多的服務中，找到滿足客戶需求的服務組合，由於有些人可能要的是價錢最便宜的服務，有些人要的是最快的服務，有些則是要介於兩者之間的，因此 QoS 的屬性必須根據使用者的非功能性需求去制定服務指標，根據服務指標的屬性找出最適合的服務組合，由於服務的種類數量龐大與種種 QoS 條件的限制根據，如果要找出符合服務品質的服務組合是相當困難的，本研究使用 GA 幫助我們更容易的去選擇符合限制條件的組合，確實可以幫助在多個抽象服務中找出適合的服務，以保證服務的品質。

本研究僅針對特定服務選擇問題去設計與實作，未來可針對不同的條件限制去設計不同的演算法，由於服務選擇有不同的策略因素需要考慮，有時候要考慮到服務的相依性與衝突性，或許可以將演算法以 API/Library 的方式方便使用者針對不同的情況使用。

參考文獻

- [1] Michael R. Garey and David S. Johnson., "Computers and intractability: a guide to the theory of NP-completeness," W. H. Freeman, 1979.
- [2] Cloud Computing | Technology Research | Gartner Inc.,
<http://www.gartner.com/technology/topics/cloud-computing.jsp>
- [3] IDC Cloud – An IDC Four Pillar Research Area,
<http://www.idc.com/prodserv/FourPillars/Cloud/index.jsp>
- [4] Cloud Computing, http://en.wikipedia.org/wiki/Cloud_computing
- [5] Jorge Cardoso, Amit Sheth, John Miller, Jonathan Arnold, and Krys Kochut, "Quality of service for workflows and web service processes," Web Semantics: Science, Services and Agents on the World Wide Web, Volume 1, Issue3, pp.281 – 308, 2004.
- [6] Object Management Group. UML Profile for Modeling QoS and Fault Tolerance Characteristics and Mechanisms., <http://www.omg.org/spec/>
- [7] Shuping Ran, "A Model for Web Services Discovery with QoS," SIGecom Exch., Volume 4 Issue1, 1–10, 2003.
- [8] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, and Marlon Dumas, "QoS-aware middleware for web services composition," IEEE Transactions on Software Engineering, Volume 30 Issue5, pp.311–327, 2004.
- [9] Daniel A. Menasce', Honglei Ruan, and Hassan Gomaa, "QoS management in service-oriented architectures," Performance Evaluation, Volume 64 Issue7-8,

- pp.646 – 663, 2007.
- [10] M. Tian, A. Gramm, H. Ritter, and J. Schiller, "Efficient Selection and Monitoring of QoS-Aware Web Services with the WS-QoS Framework," 2004 IEEE/WIC/ACM International Conference on Web Intelligence, pp.152–158, 2004.
- [11] Tao Yu and K.-J. Lin, "A broker-based framework for QoS-aware Web service composition," 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, pp.22 – 29, 2005.
- [12] Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, and Richard Franck, "Web service level agreement language specification,"
<http://www.research.ibm.com/people/a/akeller/Data/WSLASpecV1-20030128.pdf>
- [13] Vladimir Tasic, Kruti Patel, and Bernard Paturek, "WSOL - Web Service Offerings Language," Web Services, E-Business, and the Semantic Web Lecture Notes in Computer Science, Volume 2512, pp.57–67. Springer Berlin / Heidelberg, 2002.
- [14] Shuping Ran, "A Model for Web Services Discovery with QoS," SIGecom Exch., Volume 4 Issue1, pp.1–10, 2003.
- [15] Holland, J. H., "Adaption in natural and artificial systems," The University Michigan Press, Ann Arbor , 1975.
- [16] GAUL: Genetic Algorithm Utility Library,
<http://gaul.sourceforge.net/intro.html>(July 1st)
- [17] Srinivas, M., "Genetic algorithms: a survey," *Computer*, Volume 27, Issue 6, pp.17-26, JUNE 1994.

- [18] Chang W. A., "Elitism-based compact genetic algorithms," *IEEE Transactions on Evolutionary Computation*, Volume 7, Issue 4, pp.367-385, AUG 2003.
- [19] V. Deora, J. Shao, G. Shercliff, P.J. Stockreisser, W.A. Gray, and N.J. Fiddian, "Incorporating QoS Specifications in Service Discovery," *Web Information Systems WISE 2004 Workshops*, Volume 3307 of *Lecture Notes in Computer Science*, pp.252–263, Springer Berlin / Heidelberg, 2004.
- [20] Liangzhao Zeng, Hui Lei, and Henry Chang, "Monitoring the QoS for Web Services," *Service-Oriented Computing ICSOC 2007*, volume 4749 of *Lecture Notes in Computer Science*, pp.132–144, Springer Berlin / Heidelberg, 2010.
- [21] Daniel A. Menascé', Honglei Ruan, and Hassan Gomaa, "QoS management in service-oriented architectures," *Performance Evaluation*, Volume 64, Issue 7-8, pp.646 – 663, 2007.
- [22] Vijay Machiraju Akhil Sahai, Anna Durante, "Towards automated SLA management for web services," <http://www.hpl.hp.com/techreports/2001/HPL-2001-310R1.pdf>
- [23] Seema Degwekar, Stanley Y. W. Su, and Herman Lam, "Constraint specification and processing in web services publication and discovery," *2004 IEEE International Conference on Web Services*, pp.210, 2004.
- [24] Michael C. Jager, "Optimising Quality-of-Service for the Composition of Electronic Services," PhD thesis, Berlin Univeristy of Technology, 2006.
- [25] Poornachandra Sarang Matjaz B. Juric, Benny Mathew, editor, "Business Process Execution Language for Web Services," Packt Publishing, 2006.

- [26] Nanda Mangala Gowri, Chandra Satish, and Sarkar Vivek, "Decentralizing execution of composite web services," SIGPLAN Not., Volume 39 Issue10 pp.170–187, 2004.
- [27] T. Yu and K. Lin, "Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints," LECTURE NOTES IN COMPUTER SCIENCE, Volume 3826, pp.130, 2005.
- [28] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, and Marlon Dumas, "QoS-aware middleware for web services composition," IEEE Transactions on Software Engineering, Volume 30, Issue5, pp.311–327, 2004.
- [29] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani, "An approach for QoS-aware service composition based on genetic algorithms," conference on Genetic and evolutionary computation, pp.1069-1075, 2005.
- [30] Zhen Ye, Xiaofang Zhou, Athman Bouguettaya, "Genetic Algorithm Based QoS-Aware Service Compositions in Cloud Computing," Database Systems for Advanced Applications Lecture Notes in Computer Science, Volume 6588, pp.321-334, 2011.
- [31] Aphrodite Tsalgaidou and Thomi Pilioura, "An overview of standards and related technology in web services," Distributed and Parallel Databases, Volume 12, Issue 2-3, pp.135–162, 2002.
- [32] 繽紛多樣的雲端服務，政府機關資訊通報第 297 期 中華民國 101 年 7 月
<http://www.dgbas.gov.tw/public/Data/262910255771.pdf>