私立東海大學

資訊工程研究所

碩士論文

指導教授：陳隆彬　博士

階層式內容遞送網路之改良型服務部署策略

Enhanced Service Deployment for Hierarchical Content Delivery Networks

研究生：蔡培民

中華民國九十九年四月

摘要

隨著雲端與格網計算的興起，網路服務架構(SOA)的技術逐漸受到重視，它運用網路服務架構的快速與彈性的特性，發展全面性的網路服務系統。本篇論文中，我們以數位內容服務的遞送為應用課題，來探討如何部署內容服務元件以加強內容遞送的效能。本篇論文將對於階層式內容遞送網路之服務部署策略，使用 OMNet++軟體來進行模擬，驗證該演算法對分散式服務部署伺服器的效能影響。

關鍵詞：服務部署、Web-service、OMNet++

頁次

摘要……I

頁次……..…II

圖表頁次……..…III

壹、介紹……..…1

貳、內容服務部署的問題模式……3

一、Object Dependency Graph(ODG)……3

二、部署項目……..5

三、階層式伺服器網路的部署策略……..…..5

四、應用程式部署策略演算法……..…..7

(一) 部署演算法的基本模組…….,,.…7

(二) 階層式部署……..….9

參、實驗方法與結果……..….12

一、OMNet++…….……12

二、OMNet++程式開發…….……15

三、實驗結果…….……..17

(一)主從式架構……...……17

(二) 階層式網路……...…..18

肆、結論……..…20

參考文獻…….……21

圖表頁次

圖 1: SOA（Service Oriented Architecture）：服務導向架構……..1

圖 2：程式語言電子書的 ODG ……4

圖 3: (a)最佳部署……7

圖 3 : (b)對應於(a)之網路切割……9

圖 4:物件與物件行成的 ODG 圖形……10

圖 5(a):內容伺服器網路架構 S=(S0,S1,S2)……10

圖 5(b): ODG 部署在階層式網路的一組部署策略……11

圖 6:OMNet++ 模型……13

圖 7: (a) OMNeT++程式編輯與執行 GUI 畫面……..14

圖 7: (b) OMNeT++模擬分析工具 GUI 畫面……..15

圖 8:OMNet++運作流程……16

圖 9:主從式架構……...17

表一：主從式架構部署時間……20

圖 10:階層式網路架構……19

表二：階層式網路架構部署時間……19

壹、介紹

網路服務架構(SOA)的概念是將單一軟體分割成許多被稱為服務的元件，服務是發佈在 Internet 的元件，它不一定由企業的 IT 部門自行開發，提供服務元件的開發商稱為服務提供者，而建構應用程式的企業就是服務消費者，其精神是資訊模組能被快速與彈性的解構與重組[8]如圖 1，而構成一個具有特定功能的企業應用程式。企業的客戶透過調用企業的應用程式而得到企業的服務。

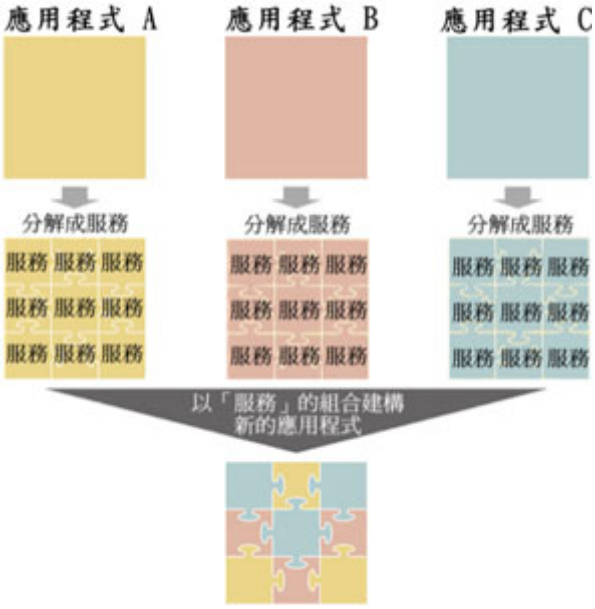


圖 1: SOA（Service Oriented Architecture：服務導向架構

在許多的網際網路應用中，當一個內容物件 b 要從主機 A 遞送到主機 B 時有兩種可能的傳送方式，第一種方式為主機 A 直接經由網路傳送。第二種方式為主機 A 將所需前行者物件送至主機 B，再由

主機 B 以計算方式來產生 b。例如，web 伺服器可將 XML 文件傳給用戶端的 web browser 進行 XSLT 轉換而產生 HTML 網頁來呈現，經由使用者端的計算能夠加強內容遞送的執行，使用者端的計算基於他的前行者節點執行轉化運作來建立物件[4],[12]，如何針對每個物件選擇上述兩種遞送方式，使得一個內容應用程式中內容物件的整體派送成本最少。本論文將內容遞送的問題轉換成網路流量的問題[1][2][3][4][6][11]來建立最佳的部署策略，在此我們針對階層式網路，利用最佳服務部署演算法，使分散式內容應用程式有效地運作。本篇論文將對於階層式內容遞送網路之服務部署，使用 OMNet++軟體來進行模擬，驗證該演算法對分散式服務部署伺服器的效能影響。本篇論文的章節安排如下，第二節介紹內容服務部署的問題模式，第三節為實驗方法與結果，第四結為結論與未來工作。

貳、內容服務部署的問題模式

一、Object Dependency Graph(ODG)

一個 SOA 系統管理許多功能各異的內容應用程式，而一個內容應用程式又由許多服務組成。本研究採用有向圖 Object dependency graph (ODG)[2]來代表內容應用程式。ODG 中圖中每一個節點代表一個服務，而服務就是可執行的軟體元件。一個服務必須被部署到某一主機才能被該主機執行，而節點 a 到節點 b 之間的線條代表服務 a 與服務 b 之間的 dependency 關係。每個服務都是可執行物件，執行服務可得到執行結果，而這個結果又成為後續服務的輸入資料。我們假設 ODG 是無迴圈的，因此依照 ODG 中的 dependency 關係，服務的執行順序必定是 ODG 圖的節點的 topological order。在 ODG 中，沒有前行著的節點稱 initial node 且沒有後行著的節點稱為 target node。

以圖 2 為例，程式語言電子書包含了三個主題：Data type、Flow-control、Function，而每個主題又有 **CONTENT**、**DISPLAY**、與 **QUIZ** 等三類物件。一個 content 所表示的是每個主題的知識內容且該內容物件無法由其他物件導出。因此將 **CONTENT** 的 comp 成本設為∞，如圖 2 所示。此外，每個物件的 net 成本的設定需以物件大小與頻寬成比例以符合遞送成本。

ODG 提供一個通用框架的來說明網際網路上各種不同的應用，

以圖 2 程式語言電子書來說明。伺服器擁有所有以 XML 格式儲存的電子書內容物件的正本。根據 ODG 的問題模型，一個目標物件 X 可以經由伺服器直接遞送到用戶端，其產生的成本為 net(X)。同樣的該目標物件也可以經由用戶端自行運算所產生，以圖 2 為例，若是要得到 Display2 文件，使用者可以用 net(Display2)=120 的成本得到 Display2 文件。根據 SDG 的語意，使用者還可以用另一種方式得到 Display2，就是先取得 Content2，然後自行計算 Display2，這個方案的成本是 net(Content2)+comp(Display2)=30+30=60，顯然比第一種方案更佳。上述兩種可能的部署方案的 trade-off 問題的常見的例子就是大型文件的 patch 更新。直接傳送大型文件往往比傳送 patch 檔由使用者自行計算最新版本來得費時。

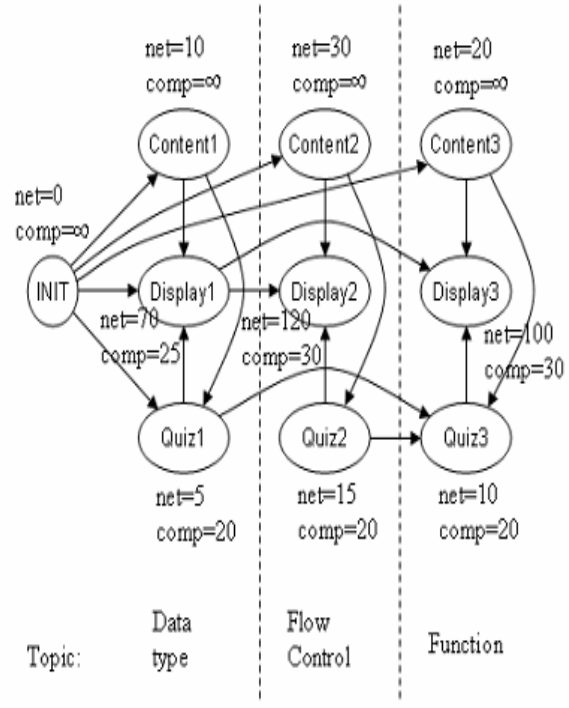


圖 2：程式語言電子書的 ODG。

二、部署項目

接下來我們描述服務部署問題的定義。部署策略是在CDN系統中執行服務與傳遞執行結果的策略。對每個網路主機S_k，一個服務的部署策略共有三種：

- O-node：ODG的節點a 是O-node 表示S_k表示不部署服務a。
- N-node：ODG 的節點 a 是 N-node 表示服務 a 是由其他主機計算後，傳送計算結果至主機S_k。
- C-node：ODG的節點a是C-node 表示服務a是由主機S_k自行計算。

對主機S_k，一對集合(N_k , C_k) 表示一個部署項目，其中

- N_k 表示主機S_k 的N-node的集合。
- C_k 表示主機 S_k 的C-node的集合。
- 不屬於 N_k 或 C_k 的節點就是主機 S_k 的O-nodes。

由於一個節點對一個主機S_k只能有一種部署方式，因此，$N_k \cap C_k = \phi$。

三、階層式伺服器網路的部署策略

在本篇研究中將採client-server的網路模式，假設網路S有M+1 個主機 $S_0,S_1,\cdots,S_M$，而且這M+1 台主機被組織為樹狀的階層式架構，以

S_0 為根節點主機，位於樹葉位置的主機稱為終端主機，那麼對這M+1個主機，我們會有M+1 個部署項目。這些項目集合起來就形成應用程式ODG圖形在網路S上的部署策略。但並不是所有部署項目都是“可行的”。只有“可行的”部署項目集合才能稱為部署策略。定義如下：

對於部署項目集合 $((N_0,C_0),(N_1,C_1),\cdots(N_M,C_M))$，以下條件限制為真：

- Computable：對所有 (Nx,Cx), $0 \leq X \leq M$,一個C-node in C_x 的前行者節點必定在N-node N_x 或是C-node C_x 之中。也就是說，C-node的前行者不可為O-node。所以， $p(Cx) \subseteq (Nx \cup Cx)$ ，其中 $p(Cx)$ 為Cx的節點的前行者集合
- Supply-Demand：若主機S_x是parent，而主機S_y是child，則主機S_y 的N-node由主機S_x來提供。所以， $Ny \subseteq (Nx \cup Cx)$
- Fulfillment：令T為ODG的target node則對所有終端主機S_x，條件 $T \subseteq (Nx \cup Cx)$ 成立。

例如圖 3(a)的 (Nx,Cx)， $Nx=(2,3,5)$ ， $Cx=(4,6,7,8)$ 滿足Computable的條件因為 $p(Cx)= p(\{4,6,7,8\}) = \{2,3,5,6,7\} \subseteq \{2,3,4,5,6,7\} = Nx \cup Cx$ 。又，(Nx,Cx) 滿足Fulfillment條件因為 $T = \{8\} \subseteq (Nx \cup Cx)$ 。圖 3(a)只顯示主機S_k 的策略 (Nx,Cx)， 若server的策略 (N_0,C_0) 包含 Nk 則Supply-Demand條件亦被滿

足。

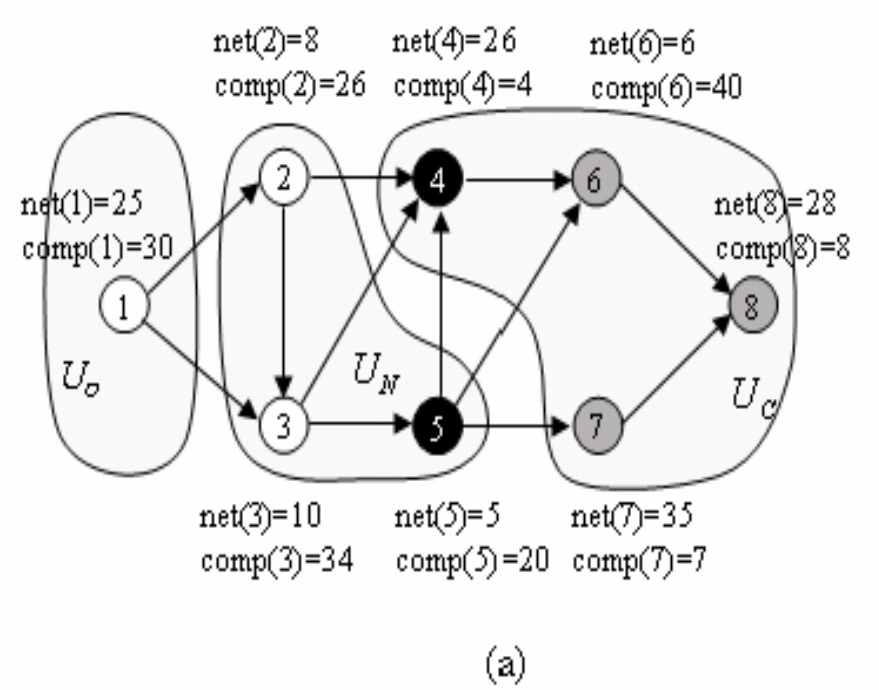


圖 3: (a)最佳部署

四、應用程式部署策略演算法

本篇論文將運用雲端計算概念以及 SOA 的快速與彈性的特性，發展高效能的內容服務網路。我們針對階層式網路，利用部署策略演算法，使分散式內容應用程式有效地運作。

(一) 部署演算法的基本模組

根據[2]，client-server網路的部署策略問題，可以轉換成圖形最小切割問題(或是最大流量問題)來解決。圖 3(a)顯示以最小切割來求出最佳部署的方法的簡化概念 。首先 ， 每一個ODG節點i (圖 3(a)) 被

轉成流量圖的i_b和i_e兩點 (圖 3(b))。流量圖中加了許多成本為∞的線，強迫流量圖的切割與ODG的部署策略之間存在著一對一的對應關係：

流量圖的切割 C 與 ODG 的部署策略對應關係：

- ODG的節點i為O-node，流量圖的edge的i_b和i_e皆在切割C的左方。
- ODG的節點i為N-node，流量圖的edge的i_b在切割C的左方而且和i_e在切割C的右方。
- ODG的節點i為C-node，流量圖的edge的i_b和i_e皆在切割C的右方。

例如，圖 3(a)的部署項目({2,3,5}，{4,6,7,8})對應到圖 3(b)的切割C_1，而且它們的成本相同。因此，求出流量圖的最小切割，再依照上述對應關係轉換，即可求出最佳部署策略。

所謂“最佳部署策略”就是所有合乎上述定義的部署策略中成本最低的那一組。

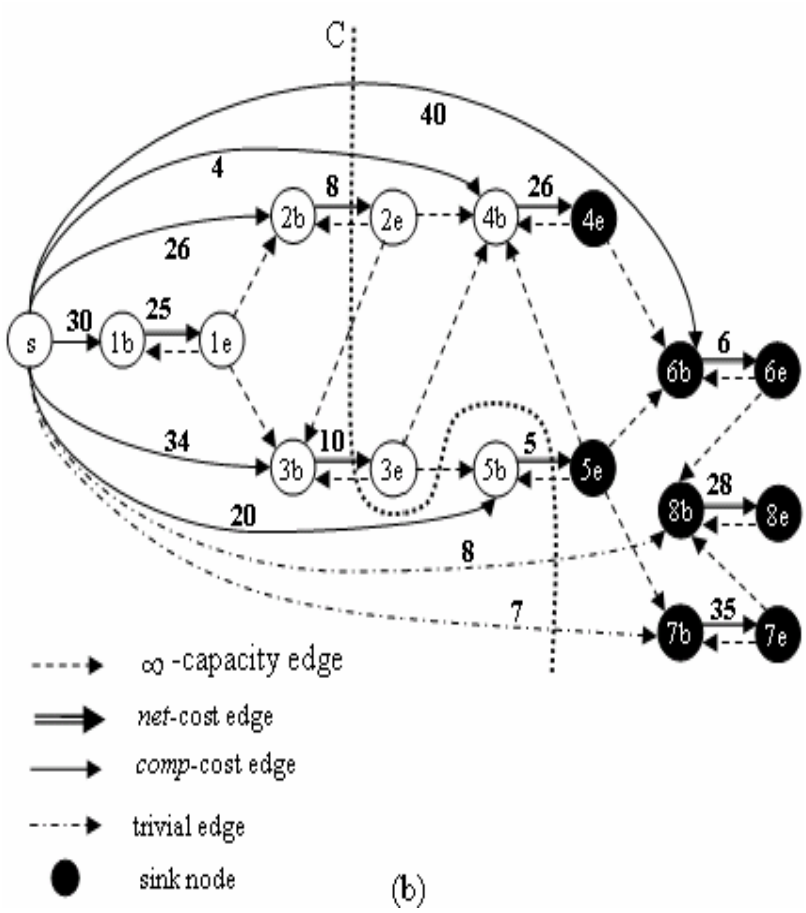


圖 3 : (b)對應於(a)之網路切割。

(二) 階層式部署

在階層式部署中採用二層式架構，根據[10]，外層是由主機形成的樹狀架構，這個樹狀架構的節點為”複合式節點”，包含了物件與物件形成的 ODG 圖形架構，如圖 4。

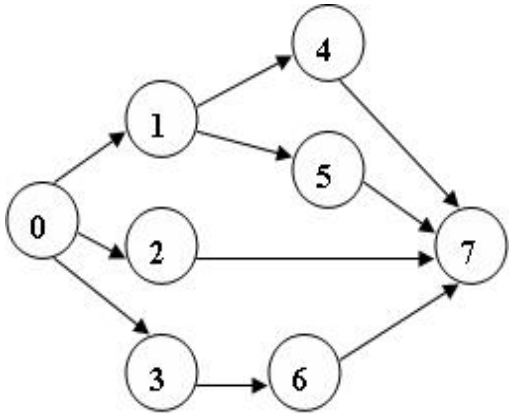


圖 4:物件與物件行成的 ODG 圖形

在此，利用主機 ODG 內的節點分別與各子主機內的相應節點來將內外層連結形成一個服務部署網路。每個複合式節點中都有相同的 ODG，圖 5 顯示服務部署的一個簡單的例子，圖 5 中，圖 5(a)為一個階層式的內容伺服器網路架構，圖 5(b)則展示了將 ODG 部署在階層式網路的一組部署策略。以圖 5(b)為例，外層主機與子主機的連結是以 ODG 相對應的節點相互連結（例如 node 1 分別與 node 9、node 17 連結）。當圖 5(b)的圖形建構完成後，我們再將其視為一個單一的 ODG 圖形，利用 2.1 節的方式求其最佳解。

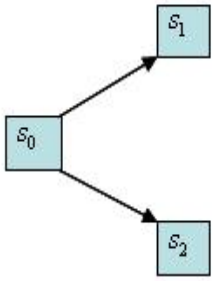


圖 5(a):內容伺服器網路架構 S=(S0,S1,S2)

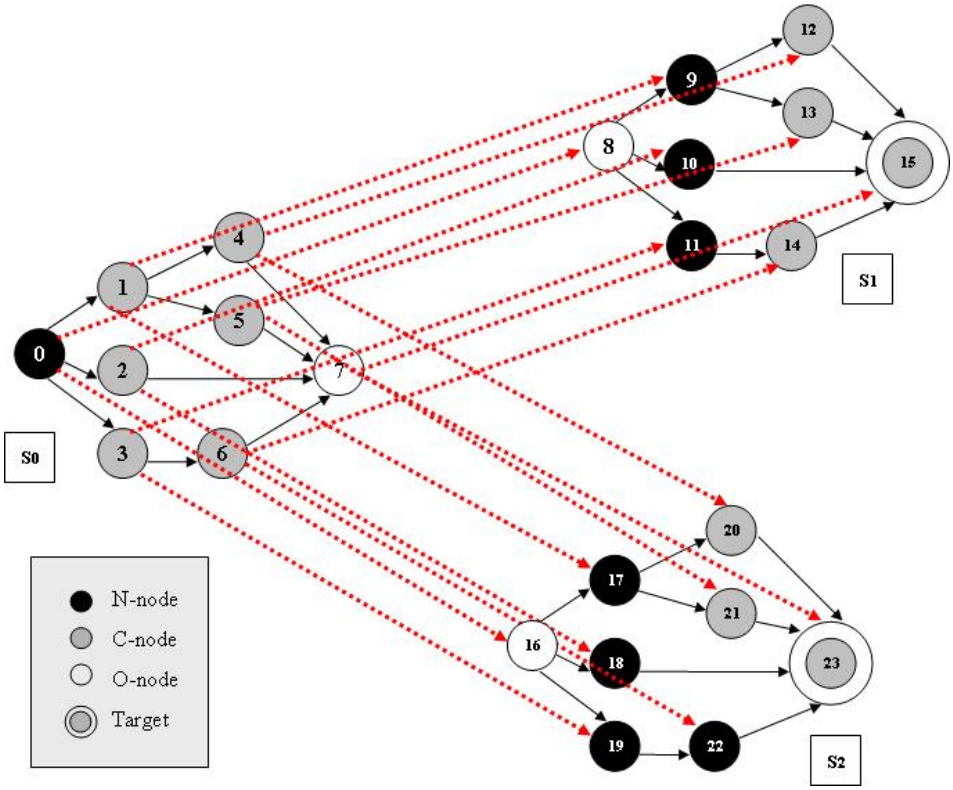


圖 5(b): ODG 部署在階層式網路的一組部署策略

參、實驗方法與結果

為了驗證在多重主機環境下服務部署演算法的部署效果，本篇研究中以 OMNet++[9]套件來模擬分散式系統的運作。OMNet++可以利用多種機制來進行分散式系統的模擬，例如，MPI 和指定的通道，能讓我們進行大量網路節點的模擬。我們的模擬方式是實作真實的 Web service，由於 OMNet++具有外接真實網路的 gateway，我們可以將真實應用程式的網路訊息導入 OMNet++模擬網路，如此能模擬大量主機的運作，而且又能實際進行 Web service 的運作。

一、OMNet++

OMNet++ 是一套 discrete event network 的模擬平台軟體。它使用物件導向程式的技術，適用各種應用領域：

- 有線及無線網路的模擬
- P2P 網路的模擬
- 協定模式
- Queuing network 模式
- 多處理器系統
- 硬體架構的驗證
- 複雜軟體的效能驗證

- 一般性的 discrete event network 應用

在 OMNet++ 中，被模擬的系統是以基本元件 “module” 組成，module 是以一段程式碼配合 script 表示，可以重複使用。module 中定義了數個 gate 用來與其它 module 連接。module 可以是巢狀結構而且結構的深度沒有限制，其最頂層的 module 稱為系統模型，系統模型也包含了子模型，而子模型還可以在包含子模型如圖 6，即允許使用者在模擬的環境中繪製實際系統的邏輯結構。

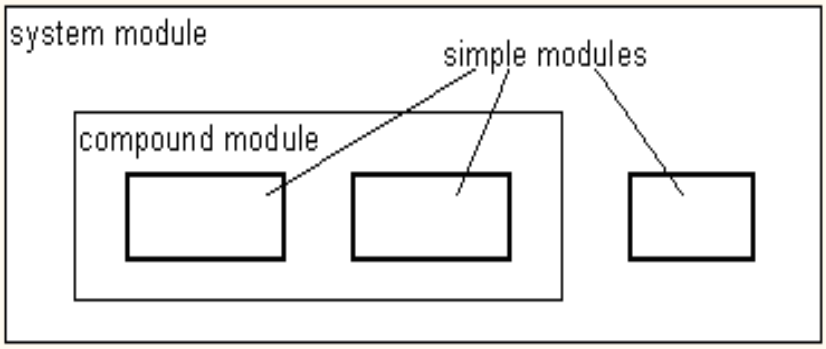


圖 6:OMNet++ 模型

module 透過 gate 傳送 message 與其它 module 溝通。Message 的種類與行為皆可以程式化定義，非常有彈性而且可以重複使用。目前已有許多無線與有線網路的特徵的 OMNeT++程式被發表在相關網路論壇中。OMNet++ 也包含了 GUI 介面與程式開發環境如圖 7(a)。

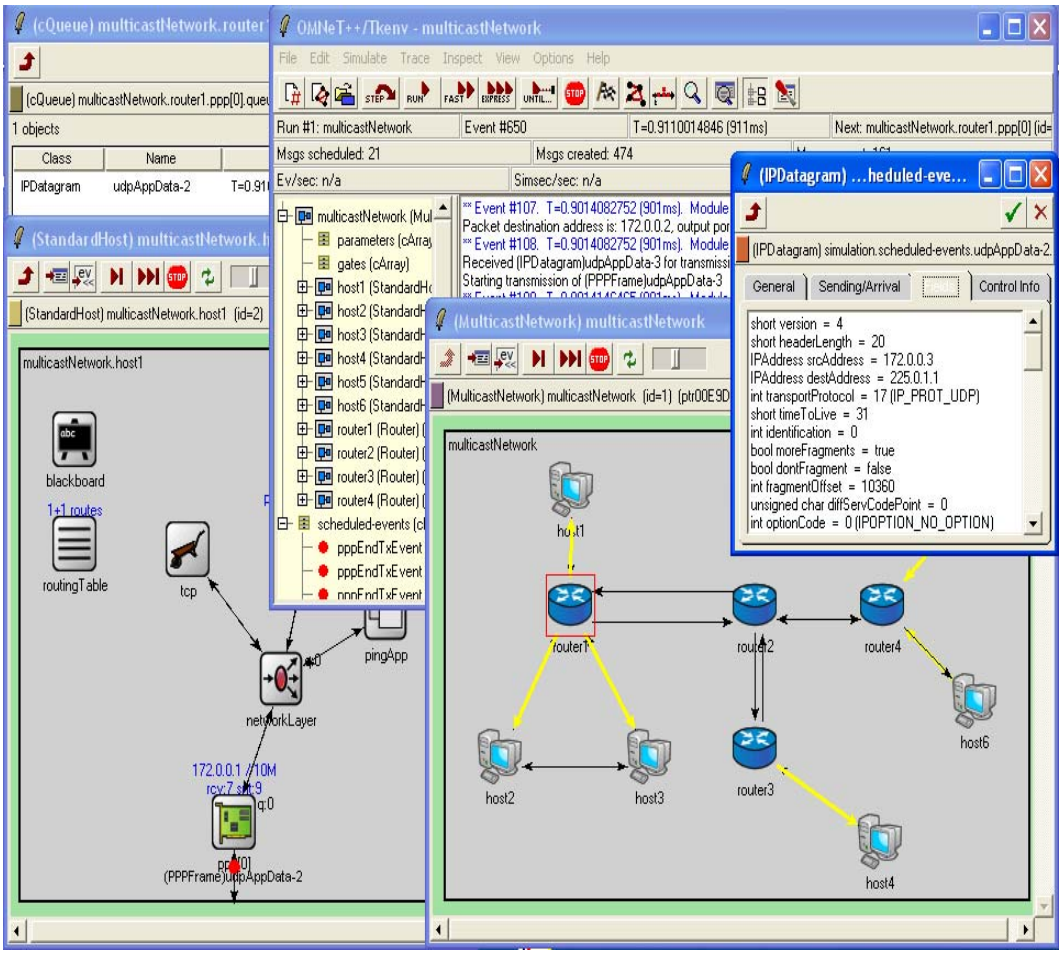


圖 7: (a) OMNeT++程式編輯與執行 GUI 畫面

方便使用者開發測試模擬程式。最重要的，完整的執行、分析、紀錄、等除錯工具如圖 7(b)，使 OMNet++的使用者能精確的分析模擬工作的結果。目前 OMNeT++支援 Windows 以及 Linux 作業系統。學術使用為開放原始碼授權，而商業化用途則須取得 Simulcraft 公司授權。

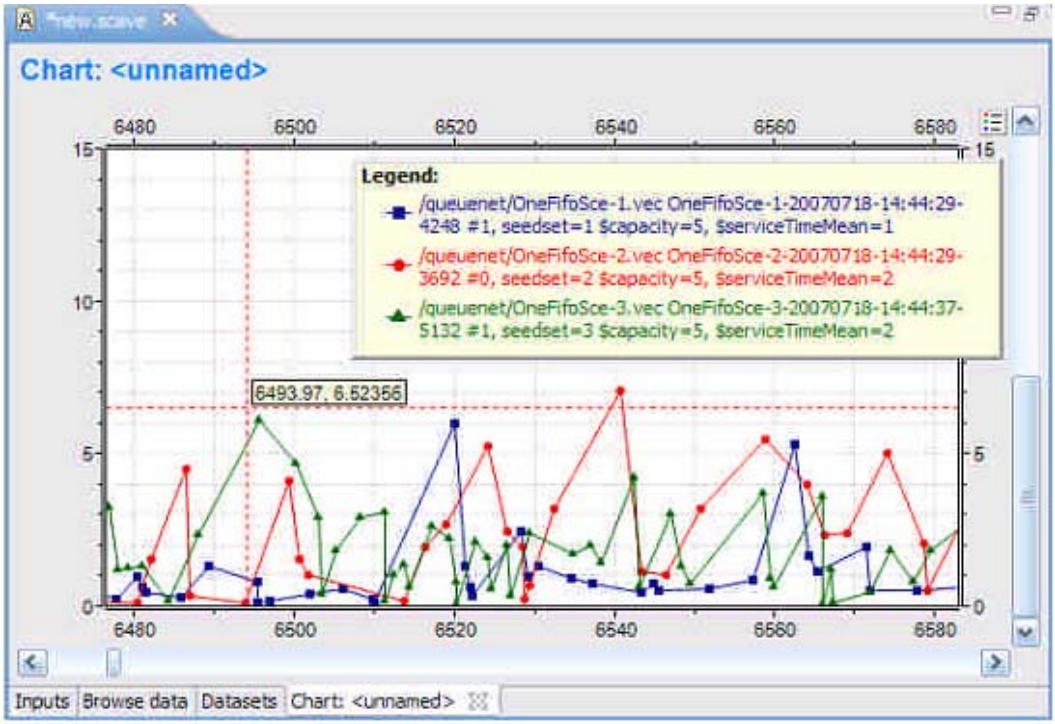


圖 7: (b) OMNeT++模擬分析工具 GUI 畫面

二、OMNet++程式開發

根據上述多重主機部署在 OMNet++的設定，當開始執行程式時，主機會開始由 initial-node(node 0)開始廣播，其餘 c-node 等待其前行節點完成傳遞後開始計算，c-node 完成計算後廣播給子主機相對應的 n-node。　此時子主機 n-node 接收來自主機 c-node 運算完成的廣播，子主機其 c-node 開始等待其前行節點的傳遞來完成計算，直到目標節點完成計算，則結束運作。

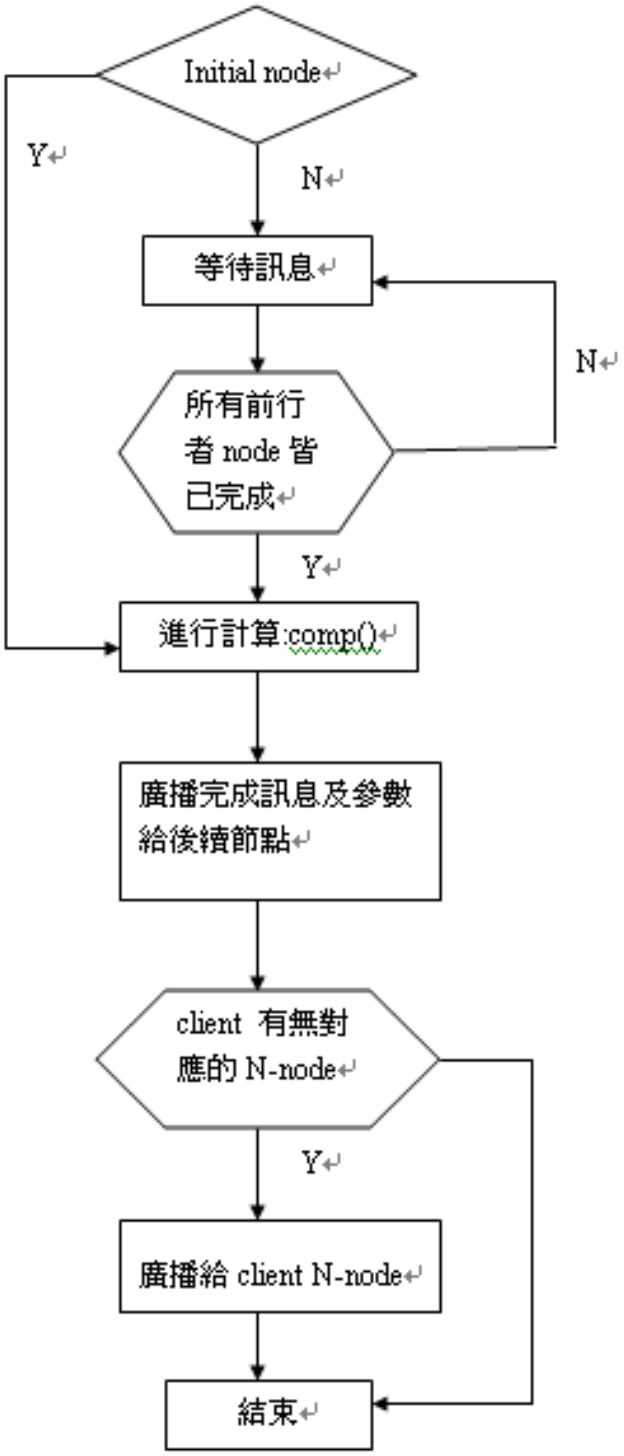


圖 8：OMNet++運作流程

三、實驗結果

(一)主從式架構

本實驗採用單一主機單一子主機的網路架構，由主機複製一份 ODG 圖形 8 個節點給子主機如圖 9，根據上述 OMNet＋＋的設定來求出該 ODG 圖形隨機部署 O，N，C 的執行時間，其隨機產生的 O，N，C 部署的平均時間為 6916ms。根據部署策略演算法，求出該 ODG 近似最佳的 O，N，C 部署，在 OMNet＋＋將該近似最佳的部署設定執行，其執行時間為 4500ms。如表一所示。

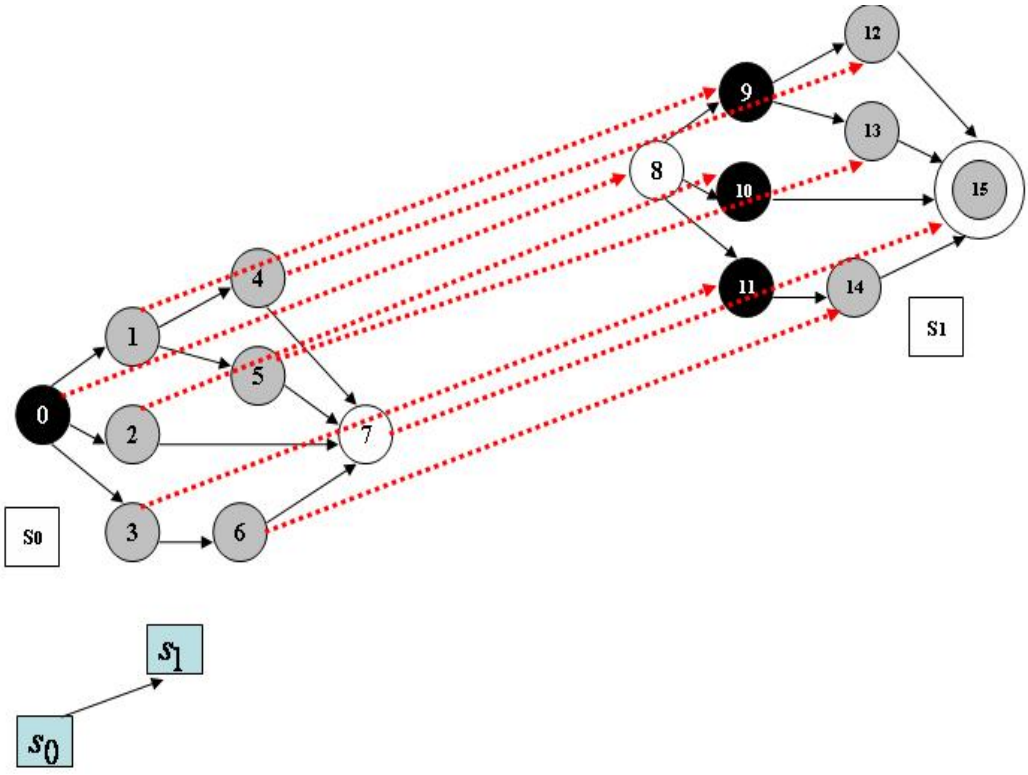
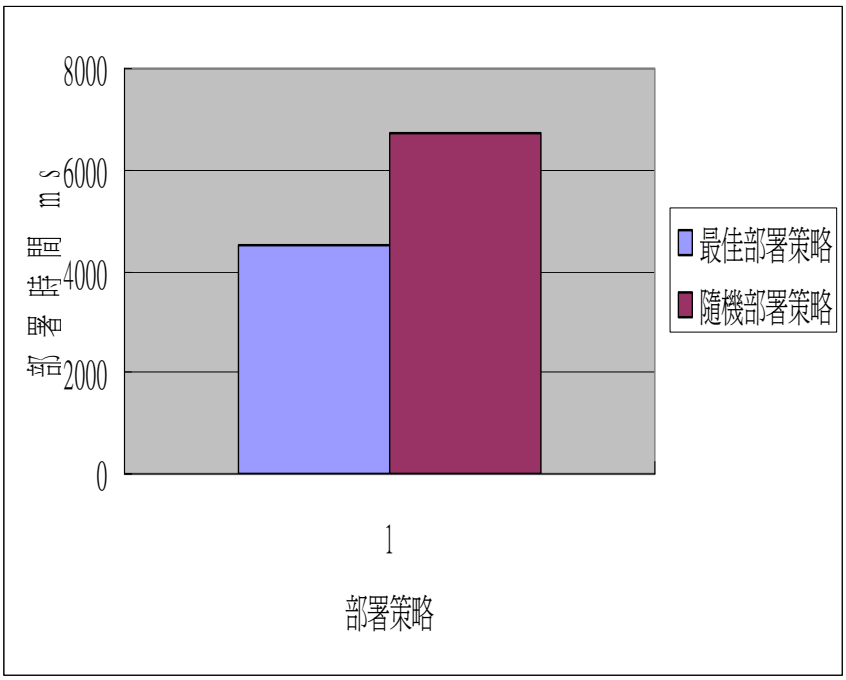


圖 9:主從式架構

表一：主從式架構部署時間



(二) 階層式網路

除了在單一主機單一用戶端的網路架構，本實驗同時也測試在多重網路架構下的 ODG 部署時間，在此建置單一主機三個子主機的階層式架構如圖 10，每個 host 都有來自於主機所複製的 ODG 圖形如圖 9 且皆有 4 個節點，同上述實驗步驟，利用演算法求出近似最佳解的部署 O，N，C，再利用 OMNet＋＋求出其部署時間，與由隨機產生的 O，N，C 部署時間，分別為 116ms 與 4293ms，如表二所示。

由上述實驗結果得知，最佳部署演算法在針對多重主機部署架構下，也能夠找出近似最佳部署策略，其部署的時間不論在單一主機與子主機或是在單一主機多重子主機架構下都能夠比隨機部署所花費

的部署時間都能夠更有效率與快速的執行。

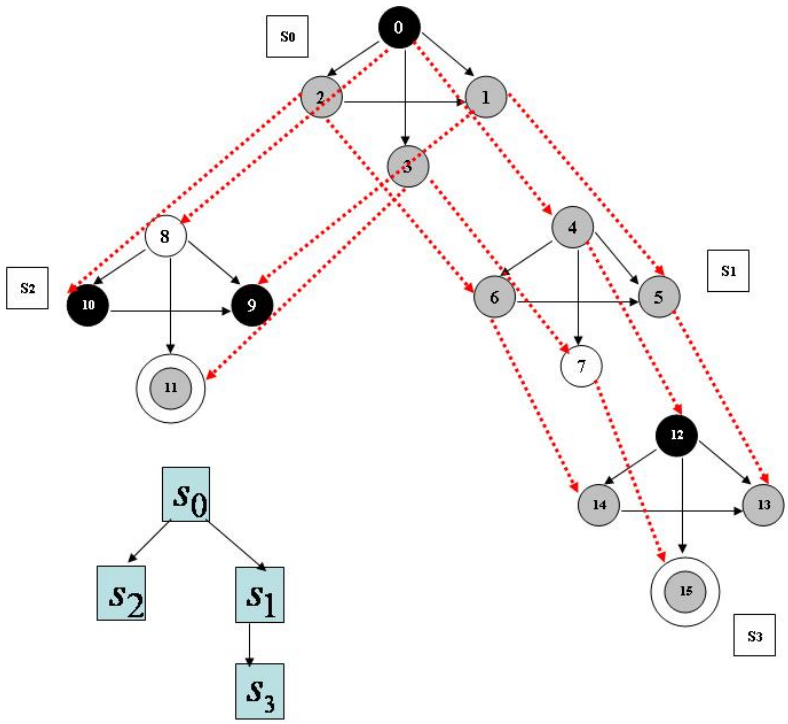
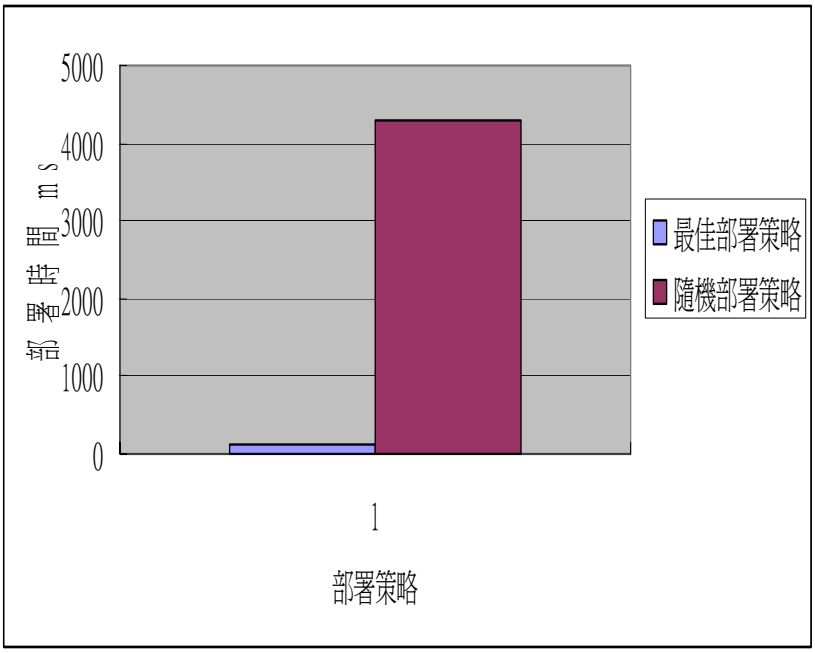


圖 10:階層式網路架構

表二：階層式網路架構部署時間



肆、結論

為了驗證最佳部署策略演算法在階層式內容遞送網路之服務部署的效能，本篇論文以 OMNet++套件來模擬分散式系統的運作。我們參考真實應用程式的網路訊息導入 OMNet++模擬網路，如此能模擬真實的網路服務元件的運作。

從上述實驗模擬結果得知，最佳部署演算法所演算出的最佳部署策略，在單一主機環境下，其 ODG 的 O、N、C 部署為最佳部署策略，而在階層式內容遞送網路架構下，也能夠求出近似最佳的部署策略，其近似最佳的部署時間也遠小於由隨機產生的部署策略。

由此我們可以知道，在階層式內容遞送網路環境下所求出的為近似最佳解，但效能上已經明顯的比隨機部署的效能快很多，這樣的實驗結果將有助於研發新的部署策略演算法，使其在階層式內容遞送網路架構下亦能求出最佳部署策略。

參考文獻

1. A.V. Goldberg and R.E. Tarjan, “A new approach to the maximum-flow problem,” Journal of the ACM, 35(4):921–940, October 1988.
2. A.Ehuchi S.Fujishige and T.Takabatake, “A polynomial-time algorithm for the generalized independent-flow problem,” J Oper Res Soc Jpn, 47:1–17, 2004.
3. C.E. Leiserson T.H. Cormen and R.L. Rivest, “Introduction to Algorithms,” The MIT press, 1989.
4. K.H. Y.L.Chin and W.Zhang, “Multimedia object placement for transparent data replication," IEEE Trans. Parallel and Distributed Systems, vol.18,no.2, pp.212-214,2007
5. K.Watanabe H.Tamura K.Nakano and M.Sengoku, “The pcollection problem in a flow network with lower bounds,” IEICE transactions on fundamentals of electronics, communications and computer sciences, E80-A:651–657, 1997.
6. L.R. Ford and D.R. Fulkerson, “Maximal flow through a network,” Can. J. Math., 8:399–404, 1956.
7. Lung-Pin Chen, I-Chen Wu, William Chu, Jhen-You Hong, and

Meng-Yuan Ho, "Incremental Digital Content Object Delivering in Distributed Systems," accepted by IEICE Transactions on Information Systems (SCI), 2009.

8. Netweaver NEC Taiwan Ltd. http://www.nec.com.tw/solutions/erp/netweaver/index.html, Sep 2009.
9. OMNet++, http://www.omnetpp.org , Sep 2009.
10. Lung-Pin Chen , “Optimal Update Delivery Strategy for Hierarchical Content service Envirnment ,” Private communications, Dec 2009 .
11. R.K. Ahuja T.L. Magnanti and J.B. Orlin, “Network Flows :Theory, Algorithms, and Applications,” Prentice-Hall, 1993.
12. X. Tang and S.T. Chanson, “Minimal cost replication of dynamic web contents under fl at update delivery,” IEEE Trans. Parallel and Distributed Systems, 15(5):431–441, May 2004.