

一種分散式異質性系統整合應用模式

An Application Model of the Distributed Heterogeneous System Integration

蕭國裕
德明財經科技大學
資訊科技與管理研究所
碩士研究生

韓孟麒
德明財經科技大學
資訊科技與管理研究所
副教授

陳文賢
德明財經科技大學
資訊科技與管理研究所
教授

摘要

本研究提出一種具有跨平台能力的分散運算模式，簡稱為異質平台整合模式(Heterogeneous Platform Integration Model, HPIM)。此模式將異質系統中的各項功能元件建立目錄，同時也建立與功能元件的對應關連。透過 HPIM 提供的資訊服務，使得用戶端能直接驅動異質系統中的各項程式元件，達到異質性系統整合之目的。

關鍵詞：異質性系統整合、主從式架構、分散式運算、雲端運算

一、緒論

異質平台整合模式(Heterogeneous Platform Integration Model, HPIM)採用主從架構(Client Server Model, C/S)為基礎，將其延伸成為具有跨平台能力的分散運算模式。簡而言之，基於 C/S 架構下的 HPIM 與傳統的 C/S 架構都是由用戶端的使用介面觸發。不同之處在於 HPIM 將每項服務的完整流程建立功能對應目錄，因此使得用戶端能直接觸發服務端提供的程式元件位址，使各元件以機械化的反射動作進行運算並回傳。此種服務模式讓流程執行的控管工作在用戶端進行，使得服務端無需維持連線或是保存運算過程而消耗過多運算資源，只需要做一問一答的簡單動作，而用戶端獲得的功能目錄中也包含服務端提供的異質性系統之資源位址。異質性系統透過網路連結後，無需限制其電腦規格，可採用低成本的伺服器或工作站，甚至可以是更低成本的個人電腦。

二、文獻探討

(一)異質性系統整合

根據過去相關研究顯示，異質性的資訊系統整合，在提高整體效能的同時，亦能降低各項成本的支出，因此成為一種良好的系統解決方案[1]。雖然整合能帶來相當多的好處，但異質性系統整合後的運行環境較單一系統更加複雜。原因是在異質系統的整合環境中，不僅需要慎重規劃資料流程的策略，更必須確保各系統的可靠性。

隨著電腦軟、硬體與網際網路的各項技術在近年來快速提升，在最近的幾年中，對於異質性系統整合領域產生非常多具有吸引力的研究項目。例如控制機械的異質系統、工業流程中的異質系統，以及商業流程中的異質系統。在這些不同領域的應用中，尋找更流暢的方式存取分散而廣泛的各種資訊，為異質性系統整合之最終目的。

(二)主從式架構(Client-Server Architecture)

資訊系統的服務架構上不斷有新技術推出，但 C/S 架構與其衍生的各種模式仍然保有優勢，例如由瀏覽器與伺服器組成的三層式(Browser-Server, B/S)架構等。C/S 架構的環境下，無論從系統執行效率，或是從成本考量的等各方面來觀察，都有著不同層面的必要性[2]。因此，C/S 架構目前普遍應用在各種領域，例如企業、學校、政府機關等。在傳統的 C/S 架構下，都能有效提供這些不同規模的資訊系統中各種需求。

C/S 架構區分為用戶(Client)端及服務(Server)端，廣泛應用在各種類型的資訊系統。在每個成功的系統案例中均能夠容易地發現，當 Server 端佈署具有高速運算能力的高階電腦，能提供 Client 端幾乎等同於 Server 端高階電腦的運算效率。根據 P. Varekova 的研究結果[2]，若服務端的運作搭配得當，則效率倍增。同時，若是經過良好的設計，透過豐富使用者介面(Rich Internet application, RIA)技術[3]提供更流暢的使用者介面並預處理部分輸入、出的運算，能使 Server 端能將所有運算資源運用在必要的工作上。

(三)分散式運算與雲端運算

分散式運算將大量的運算工作區分成小塊後，分別交由眾多電腦各自進行運算再彙整結果，以完成單一電腦無力勝任的工作。而雲端運算則強調在運算資源有限的情況下，利用網路取得遠端的運算資源。探討其概念及機制，有助於讀者對本研究接下來的內容有更進一步的體認。

1. 分散式運算(Distributed Computing)

科學研究為了探索大自然或宇宙間的事物，因此需要龐大運算資源。但為了建立起強大運算資源的成本費用，並非所有研究計畫都負擔得起。於是許多的科學研究利用網路，將成眾多分散各處的電腦連結起來，分派適當的運算工作進行研究。以參與運算的電腦觀點，每項工作時間只處理少量的運算工作，而研究計畫則能在較短的時間內，得到所有電腦的運算結果，既能縮短研究所要消耗的時間，亦減少成本的花費[4]。

2. 雲端運算(Cloud Computing)

雲端運算的概念屬於廣義的分散式運算，簡而言之，就是由服務供應商提供程式運算、軟體應用與資料儲存能力。隨時隨地透過網際網路存取使用。資料儲存不必再完全倚賴個人電腦或企業所架設的伺服器，而是分散在服務供應商建構的大型資料中心（data center）。

據 Gartner[5]將雲端運算區分為兩大類，分別為雲端服務(Cloud Computing Services)與雲端科技(Cloud Computing Technologies)。

雲端服務專注在於透過網路取得遠端服務。使用者所需的各項服務，可以全部在遠端電腦執行。因此使用者無須配備相關的軟、硬體，即能夠隨時使用雲端提供的各項服務。

而雲端科技則是將規劃出各種運算資源，套用在整個企業的內部系統上。有鑒於一般企業的投資在資訊系統發展的經費中，約有 80%耗費在既有系統的維護，而非開發新功能。若企業能將資訊系統的維護工作交付給雲端運算的服務供應商，將能分配較多的經費投資於企業核心的業務或研發領域。

三、問題探討

(一)異質性系統整合問題

企業內部的系統整合問題隨著企業發展會不斷地發生，所以在進行異質性系統整合的過程中，必須考慮眼前問題，也要將未來可能發生的問題納入考慮。若企業內部各系統之間缺少整合，將使得整體的效益反而受到資訊系統的限制，甚至造成第一線使用者的困擾。而相同的資料不斷重複的輸入至不同的系統中，單只是簡單的查詢與核對工作，卻要開啟許多套系統同步進行且步驟十分繁瑣。這些狀況的發生，對於企業所累積的損失是相當難以估計的。

(二)C/S 架構產生的問題

C/S 架構需要配備較高規格的電腦，因此所需

的建置與維護經費成本過高，汰換率也隨著硬體科技的提升而不斷增加。而這些特殊規格的電腦，其功能也缺乏彈性，當系統需要修改或新增功能，花費的時間及成本大，且作業流程相當複雜。

其次，C/S 架構下，必須承擔系統尖峰使用時的效能緩慢甚至系統崩潰的風險。雖然能夠透過良好的設計令用戶端分擔部分運算工作，但用戶端仍然僅能夠在輸入、出的過程中提供部分簡單的運算工作，對於服務端進行的主要運算工作幫助有限。因此當系統使用尖峰時，容易產生用戶端的需求數量過多，導致服務端運算資源不足形成服務端回應遲緩，甚至造成服務端崩潰的狀況。雖然近年來的科技不斷創新，使得電腦設備的運算能力水準普遍提高。但是，根據上述的問題，造成 C/S 架構的服務模式下，無法有效的提升運作效率。

基於上述問題因此勢必要為 C/S 架構思考其功能上的改進，以因應逐漸產生的相關問題。

(三)分散式運算與雲端運算

隨著分散式運算(Distributed computing)的興起，各大廠商推廣的雲端運算(Cloud computing)技術也成為一種解決方式。雖然都得到了不同程度上的成功，但是前者的解決方式並不能換來永久的效益；而後者則要根據不同型態的雲端技術導入來重新設計，換句話說兩項選擇都會產生相對的成本和風險。

四、異質性系統整合應用模式

有鑒於目前常見的資訊系統架構衍生出的各種問題，本研究提出以異質性系統的合作模式(HPIM)解決。此模式共分為 3 階層，分別為用戶層、媒介層、處理層等。每個階層包含的工作均以簡單、快速、準確為其特性，以提升系統的整體效能為優先考量。其運作模式如圖 1 所示。

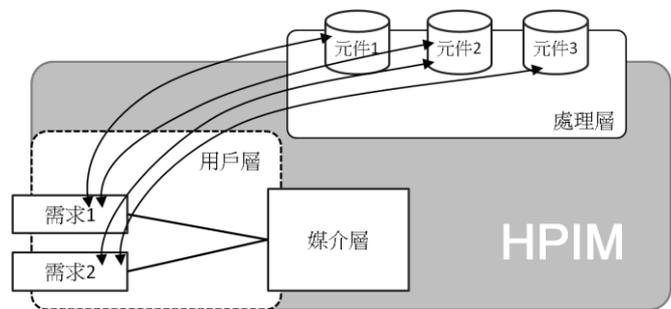


圖 1 HPIM

(一)處理層

HPIM 處理層的建構具有分散式處理與負載平衡的作用，其模式下可將所有的功能元件分散佈署

在不同的電腦主機上，利用 RS232 通訊埠編號識別元件。當通訊埠接收到其相對應的訊息格式時，立即進行單一運算工作，並且立即將處理結果回傳。若某項單一工作的複雜度較高，或執行的需求量增加時，可以適時將相同元件複製到其他的電腦或通訊埠，分散輸入、出，以及運算的工作。因此可確保功能執行過程中，不浪費任何系統資源進行暫存或等待。

建置 HPIM 元件伺服器並沒有特殊限制，元件可以是由不同的程式語言編碼。透過網路位址(IP)和通訊埠(Port)的對應，能夠佈署在異質作業系統中運行。每一個元件伺服器可將元件佈署在 0 到 65535 當中；作業系統尚未使用的通訊埠運行。這代表每一個元件伺服器都具有非常大的空間佈署元件。

(二) 媒介層

HPIM 之所以能夠整合異質性系統，關鍵在於媒介層提供的索引服務。其服務將執行處理的流程，以及功能所需的元件所在 IP、Port 建立功能目錄。當使用端提出需求時，經由媒介層將目錄中對應的資訊提供使用端，讓使用端能夠順利運用 HPIM 整合下的異質性系統提供的運算資源。

(三) 用戶層

為了避免複雜的工作佔去用戶端與使用端過多的時間相互等待，因此 HPIM 將執行的工作切分為不同的元件服務，透過媒介層提供的元件位址及流程，使得用戶端所需的各項工作得以分散至不同元件伺服器進行處理。

五、實驗

本研究模擬一個具有複雜運算以及遠端資料庫存取的資訊系統，其功能包含：資料輸入、出、資料加密、資料儲存等 3 項功能。實驗內容將此 3 項功能的流程分別採用一般 C/S 架構以及 HPIM 架構下分別運行。其功能元件均採用相同程式碼編寫；品質完全相同，因此能夠進行公平的比較。

(一) 元件設計

實驗過程首先依據模擬的功能需求設計各項元件。本實驗依據功能需求在設計了 3 個共用元件，分別為：輸入元件、加密元件、資料庫存取元件等。

1. 輸入、輸出元件

本研究以讀取文字檔案的方式模擬大量資料輸入的動作，因此輸入元件的主要功能為解析文字檔內容並按照行數依序回傳。此元件在本實驗中如

同用戶端的使用者操作介面。

2. 加密元件

為模擬複雜的運算工作，本實驗設計以 DES 演算法加密文字的元件。雖然此元件仍可細分出多個元件，為了讓讀者容易解讀數據，因此，本研究將整個加密工作包裝成為 1 個單一元件。此元件的運作模擬出服務端的運算過程。

3. 資料庫存取元件

常見的 C/S 架構皆具有資料庫系統的存取動作，本實驗因此建立 1 個 MySQL 資料庫伺服器，作為實驗中進行資料庫連線及存取的對象。也因此本實驗將資料庫存取工作包裝成為 1 個單一元件，透過資料輸入觸發其運作。

由以上 3 項基本元件即可提供一般 C/S 架構運行。為了實現本研究提出的理念，本實驗增加另外 2 項元件，以實現 HPIM 異質性系統整合的功能。分別為：目錄元件、流程控制元件。

4. 目錄元件

HPIM 架構下，用戶端無法得知服務端具備的各項功能，亦無法取得各項元件所在的網路位址和通訊埠編號。因此，在該架構下的媒介層將提供索引的服務。目錄元件在本實驗中不進行任何運算，僅提供方法對應的各項元件位址，用戶端取得相關資訊後，才能開始運用到異質性系統整合帶來運算資源。

5. 流程控制元件

目錄元件產生相關資訊後，用戶端需要進行解析並且將工作依序執行。本實驗將流程控制元件佈署於用戶端，流程控制元件在本實驗中不進行任何運算，僅用於觸發 HPIM 架構下各元件運算工作，並將處理完成的結果依序進行下一階段之運算工作。

(二) 實驗過程

1. 對照組

設計完成的元件首先採用標準的 C/S 架構運行；其流程由用戶端連結服務端，連結後將未加密的資料傳送至服務端，用戶端開始等待服務端回應執行結果。而服務端將收到的資料執行加密，並將加密後的資料傳寫入資料庫，再將結果回傳至使用端，此流程重複運行至文字檔案結尾，過程如圖 2 所示。

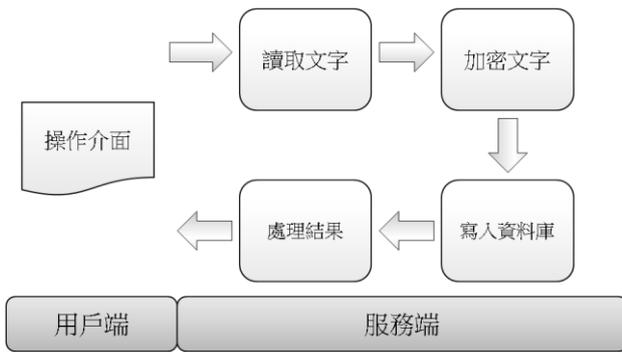


圖 2 對照組中各項元件執行流程

2. 實驗組

實驗組採用與對照組相同的元件建置 HPMI 架構，與對照組不同之處分別於用戶端增加功能索引功能，以及將元件分別部署在服務端不同的伺服器通訊埠中運行。實驗過程中，用戶端程式的工作模式是先令用戶端連結服務端取得方法，用戶端取得方法後，根據方法實行各項工作。其方法包含：程式運作流程，以及元件對應的伺服器位址以及通訊埠編號。實驗的內容始於用戶端依照方法將欲加密資料傳送至元件伺服器通訊埠，當加密元件伺服器接收到連接埠通訊時執行加密工作，完成後立即將加密資料回傳至用戶端，用戶端依據流程將加密後的資料傳送至資料庫存取元件，而資料庫存取元件立即執行與資料庫的存取工作，並將結果傳回用戶端。

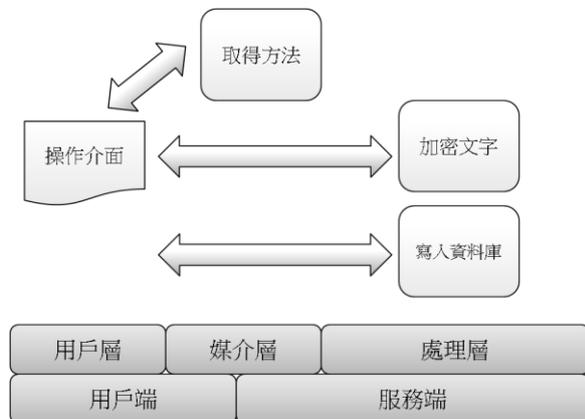


圖 2 實驗組中各項元件執行流程

五、實驗結果

本實驗共分為十個區間，以 1000 筆資料遞增。每區間測試 10 次並記錄用戶端的執行時間，加總後平均如表 1 所示。

表 1 HPIM 實驗結果 (單位:秒)

次數	C/S	HPIM
1000	26.945	0.544
2000	60.742	1.089
3000	86.263	1.482
4000	111.459	2.145
5000	136.883	2.672
6000	164.165	3.152
7000	196.237	3.677
8000	221.961	4.142
9000	252.297	4.604
10000	277.493	5.262

圖 3 為表 1 產生的數據所制成的直條圖，此圖能夠使讀者更加清楚的瞭解，此 2 種模式的運作下，對於用戶端的執行時間的差異性。

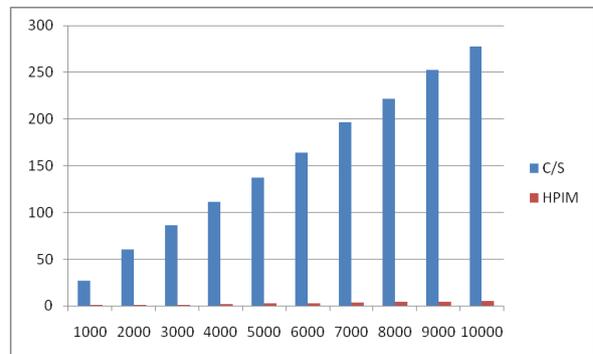


圖 3 以直條圖顯示 HPIM 實驗結果

以傳統 C/S 模式運作的過程中，服務端包含過多的運算邏輯。當大量的用戶端提出服務需求時，服務端必須依序將每項需求中的運算邏輯逐一完成，當計算資源不足或被佔用時，所有的需求只能堆積在服務端等待處理，因此而產生用戶端的等待時間。反觀以 HPIM 運作的過程中，用戶端透過媒介層取得元件位址的服務位址，直接以一問一答的作業模式完成所有工作，在過程中沒有產生任何等待時間，因此能得到較為良好的效能。

六、研究貢獻

本研究的貢獻首先是對於用戶端的執行效率的提升，其次為分散運算帶來服務端運算工作負載的平衡，第三則是整合異質性系統協同作業帶來的各項效益。本研究的實驗過程證明對使用端效能上帶來的大幅提升，透過元件伺服器的分散佈署代表分散運算的可行性，而利用目錄對應的技術則代表能夠驅動異質性系統中的各項服務。

七、後續研究

本研究提出 HPIM 的概念，依據此概念進行實驗，並進行初步的分析。雖然在實驗中得到了些許的成果，但是 HPIM 各層面的設計仍為雛型，換句話說，HPIM 仍有相當大的發展空間。因此，後續的研究可朝向 HPIM 各層面之細部設計進行更周詳的規劃，以實現更高品質的異質性系統整合。

(一) 元件對應目錄

本研究設計的元件對應目錄以電腦位址、通訊埠進等 2 項屬性進行元件對應。此種對應方法可滿足小型系統的運作。若系統持續發展，需求增加導致元件不斷增加，對元件目錄的複雜度也相對增加。因此後續的研究工作朝向導入電腦輔助軟體工程(Computer Aided Prototyping System, CAPS)[6]之技術，提升元件的重複使用性，並且能夠更快速的進行開發與相關維護工作。導入 CAPS 相關技術後，也相對導入了各種規格與屬性，對於用戶元件的對應需求，也能提供更快速的服務[7]。

(二) 輸出、輸入

本研究設計的輸出、輸入方法，均採用字元編碼，每項指令均以無格式化的文字傳送及接收。因此，訊息仍需回覆至用戶端進行確認並重新導向新的元件伺服器進行下一階段的運算。因此後續研究參考以運用格式化將輸出、輸入的訊息加以定義[8]，使得用戶端在取得功能目錄提供的方法後，直接送交服務端，令各元件參照格式化的訊息完成所有工作。過程中若能夠減少用戶端的接收服務端的回應次數，就能更加地提升系統整體效率。其次，在輸出及輸入的過程均是透過網路傳送，因此會產生資訊安全的隱憂，無論是惡意或非惡意的攻擊都會造成問題。未來也應針對資料傳遞過程中，任何可能產生的問題進行後續的相關研究。另外在網路傳送資訊的過程中，也可導入最短路徑的相關演算法。

(三) 負載平衡

在本研究的實驗過程中，僅模擬以少數電腦所組成的環境，由於受到實驗環境的限制，因此缺乏對於負載平衡的相關設計。在後續的研究中，可探討負載平衡的相關技術，如輪詢機制、權重比例、最短路徑等相關理論[9]，導入其演算法，使 HPIM 能更有效控制異質系統間相互存取的複雜的環境。

八、結論

本研究嘗試以過去不同的方式實現異質系統整合，經過實驗證明能帶來的良好效能。解決使用

端對於服務端效率的要求，透過 HPIM 所提供的異質性系統整合，可將用戶端的需求有效的分散傳送到各元件伺服器上。而 HPIM 目前仍是一種概念，這項研究只是第一次嘗試。經由本研究初步探討之後，衍生出的相關議題，值得從各方面進行後續研究。

參考文獻

- [1] Xiaoyong Tang, Kenli Li, Renfa Li, Bharadwaj Veeravalli, "Reliability-aware scheduling strategy for heterogeneous distributed computing systems," *Journal of Parallel and Distributed Computing*, Vol. 70, Issue 9, September 2010, pp. 941-952.
- [2] P. Vařeková, I. Vařeková, I. Černá, "Automated Computing of the Maximal Number of Handled Clients for Client-Server Systems," *Electronic Notes in Theoretical Computer Science*, Vol. 260, 1 January 2010, pp. 243-259.
- [3] Pawan Vora, "Rich Internet Applications," *Web Application Design Patterns*, 2009, pp 225-258.
- [4] Jim Gray, "Distributed Computing Economics," *Queue*, Vol. 6, Issue 3 2008, pp. 63-68.
- [5] Gartner: Seven cloud-computing security risks, <http://www.networkworld.com/news/2008/070208-cloud.html>
- [6] Luqi P. D. Barnes, M Zyda., "Graphical tool for computer-aided prototyping," *Information and Software Technology*, Vol. 32, Issue 3, April 1990, Pages 199-206.
- [7] P. Luqi, ; Berzins, V. ; Shing, M. ; Nada, N. ; Eagle, C., "Computer Aided Prototyping System (CAPS) for Heterogeneous Systems Development and Integration," *Electronic Notes in Theoretical Computer Science*, Vol. 260, 1 January 2010, pp. 243-259.
- [8] Mark E. Williams, Gary R. Consolazio, Marc I. Hoit, "Data storage and extraction in engineering software using XML," *Advances in Engineering Software*, Vol. 36, Issues 11, December 2005, pp. 709-719.
- [9] Vladimir V. Korkhov, Valeria V. Krzhizhanovskaya, P.M.A. Sloot, "A Grid-based Virtual Reactor: Parallel performance and adaptive load balancing," *Journal of Parallel and Distributed Computing*, Vol. 68, Issue 5, May 2008, pp. 596-608.