

行政院國家科學委員會專題研究計畫 期中進度報告

子計畫三：高可擴充性與可用性行動電子商務中介軟體技術
之研發與實作(2/3)

計畫類別：整合型計畫

計畫編號：NSC91-2213-E-002-046-

執行期間：91年08月01日至92年07月31日

執行單位：國立臺灣大學電機工程學系暨研究所

計畫主持人：郭斯彥

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 92 年 5 月 23 日

行政院國家科學委員會專題研究計畫成果報告

高可擴充性與可用性行動電子商務中介軟體技術之研發與實作

Design and Implementation of Highly Scalable and Available Middleware Technology in Mobile E-Commerce Environment

計畫編號： NSC 91-2213-E-002-046

執行期限： 民國91年08月01日到92年07月31日

主持人：郭斯彥 台灣大學電機工程學系教授

一、中文摘要

隨著網際網路與行動通訊的普及化，二者技術的整合產生了許多新穎的應用，所謂的行動電子商務就是其中之一。

要如何提供不間斷的服務，提供一個具備高可用度(availability)的系統，將是一個挑戰。另一方面，各項新技術的快速發展，使得軟體系統的生命週期變得相當短促。因此在考量網際網路應用程式伺服器系統架構時，必須將系統的維護與日後功能擴充的可能性列入考量的項目。本計畫使用分散式元件技術(distributed object)，設計以多中介層次(multi-tier)的架構提供不同的功能層級，以達成行動電子商務系統之高度可擴充性(high scalability)。另外，由於分散式元件具有良好的替代性，用於架構系統將有利於系統的維護與更新。本計畫同時著重於研究如何使系統具備高可用度。我們使用元件層級的容錯群組技術，達成系統高可用度的需求。

本年度計畫將完成交易系統中介軟體之決策邏輯與流程控制邏輯，並以CORBA技術提供系統元件之容錯，來實現系統之高可用度及高可擴充性。

關鍵詞：多層次式架構，分散式元件技術，高可用度，高可擴充性。

Abstract

As the use of the Internet and mobile communication becomes popular, the combination of these two technologies has brought many novel applications into the real world. The so-called mobile e-commerce is one of them.

However, it's not easy to convert a legacy client-server style system into Internet-compliance. The Internet clients can be worldwide. The system must have high availability to provide continuous services. Furthermore, the rapid development of new technology makes the life cycle of one system even shorter. As a result, it's important to put scalability and maintainability into consideration when constructing

the system architecture. In this project, we make use of the distributed object technology to design and implement the multi-tier system architecture. By building different function levels in the multi-tier fashion, high system scalability can be achieved. Furthermore, because of the ease of component replacement, building the system by distributed objects is also beneficial in the maintenance and upgrades of the system. At the same time, we aim at providing the system with high availability. It is achieved by using the component-level fault-tolerant group technique.

During this year we will finish the decision-making logic and flow-control logic of the middleware in the trading system. The high system availability and scalability is achieved by using fault-tolerant CORBA components.

Keywords: Mobile E-commerce, Middleware, Multi-tier, Availability, and Scalability.

二、計畫緣由與目的

電子商務可以說是一種在網際網路與行動通訊網路上執行的應用程式伺服器。然而發展網際網路應用程式與行動通訊應用程式，直接會面臨到的一項挑戰即是傳統伺服器端/客戶端(Server/Client)的通訊概念不再適用。系統架構本身需要進行相對應的調整，才能達成新的需求。另一方面，電子商務本身對於系統的需求亦和一般的網站型應用伺服器系統不相同。

電子商務最重要的一個特質是其無遠弗屆的廣泛性，網際網路所能涵蓋的服務範圍將不僅局限於某一區域網路、某一特定廣域網路；客戶端程式可能遍佈全世界，換句話說，客戶所在位置可能分布在不同時區。這表示伺服器必須能夠提供每天 24 小時不間斷的服務。系統本身需要具備相當的穩定度才能負擔這項工作。

一般用來評比一個系統是否穩定的方式包括系統可靠度、系統可用度、系統維護度三項：

- 系統可信度(reliability): 系統能持續提供正確運算服務的能力
- 系統可用度(availability): 系統在一段時間內維持正常功能的時間比
- 系統維護度(maintainability): 故障系統在一段時間之內能夠恢復正常運作的機率

對“穩定”的要求隨系統的運作方式不同而有差異。例如衛星以及海底電纜系統等，由於維修不便，因此期望能有自我修復機能(maintainability)；用於飛行控制的系統則是必須要求在一次航行的期間內系統必須能正常運作；需要的是高可靠度 (high reliability)。而一般開放大眾使用的系統，例如電話網路，行動通訊網路等則可以忍受短暫的故障，重視的是系統的可用度。現今商務與金融體系的運作，最重視的即是資料(帳目)的正確性，而短暫的交易終止、服務中斷或是

可以被接受的。因此在本計畫中系統穩定度的需求將著重於系統的可用度。

仔細審視本計畫的需求，電子商務的運作過程，可以規劃出一個比較明確的查核點與回復的問題領域：

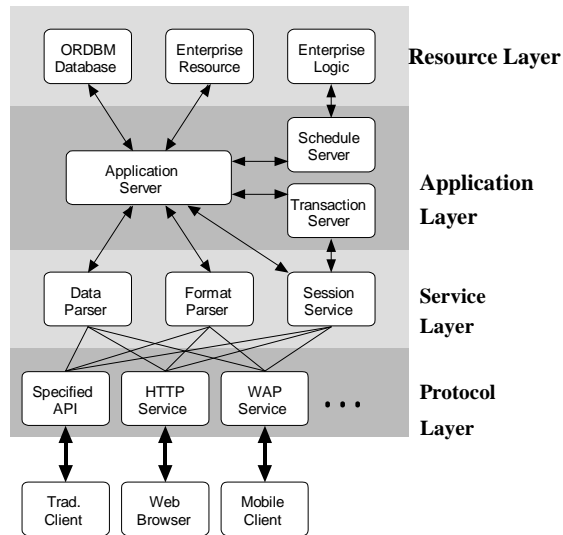
- 交易是以人為主的互動，因此處理的效率不是最優先考量的因素
- 客戶端可以忍受短暫的交易終止或服務中斷
- 資料(帳目)必須絕對正確
- 失敗的交易必須回應給客戶端

依據這些特性，本計畫可以用一個比較實用的解決方式：採用交易管理服務(transaction service)機制來進行資料查核點的建置。在商務與金融程式中為確保帳目的正確，需要大量使用交易管理服務；在交易中的每一個步驟都需經雙方確認(commit)才能正式生效。在這種情況下向後回復的機制是可以做到的。我們可以在交易確認時記錄負責進行交易過程的元件查核點，藉由元件生命週期查詢服務，在系統進行後退回復時則可以訂出回復的範圍，以確保資料與交易管制的正確進行。

三、研究方法與成果

在設計電子商務系統時所必要面對的一個問題即是：應如何在不對系統運作造成整體性的改變的前提下，增列新的支援，並同時兼顧其擴充性(scalability)。物件導向(object-oriented)式的程式寫作方式是極為適合的解決方案。物件導向程式的程式重新使用(code reuse)精神不僅利於程式的開發、維護，同時適合用於商業用途。然而物件導向程式本身重視的是抽象化資料型別(abstractive data type)的定義與抽象化物件呼叫(abstractive object invocation method)，並未著重於系統“擴充性”的提升。系統的擴充性須仰賴系統本身的良好規劃，才能順利達成。

多層次式架構現在被廣泛運用在網際網路之上；一般所謂的 CGI 程式(Common Gateway Interface)，例如 Servlet、JSP、ASP、PHP，即是由“SQL 資料庫—CGI 程式—網頁瀏覽器”三者所構成的三層式架構。這種三層式架構為一般網站型網路應用程式(website-based application)大量運用，但是要作為一個具有抽象資料型別與抽象程式運作原則的泛用型網際網路應用程式平台(Internet-based application platform)，三層式架構並不完全。在圖一中顯示了一個網際網路應用程式五層式邏輯架構層級圖。



圖一、網際網路應用程式五層式邏輯架構層級圖

本計畫第二年度的工作重點在於設計一具備容錯回復機制的元件式系統架構，討論如下：

要使系統達到高穩定度的要求，可以使用錯誤偵測(fault detection)，錯誤排除 (fault removal)，錯誤預防(fault forecasting)等方式：

- 錯誤偵測(Fault Detection): 偵測系統發生錯誤的位置及類型，以便系統容錯機制進一步處理。
- 錯誤排除(Fault Removal): 藉由容錯機制的運作，將發生錯誤的部分恢復正常，或是利用備份系統取代發生錯誤的部分，維持系統正常運作。
- 錯誤預防 (Fault Forecasting)：藉由預測可能出現的錯誤及其影響範圍，來提供處理系統錯誤的方式。

這些方式通常用在系統發展或是系統維護時，可使用種種工程技巧來進一步提升系統的穩定度。讓系統具備在發生錯誤時依然能夠繼續運作的能力，即是容錯(fault tolerant)。容錯的基本原理就是備份。“備份”指的就是在系統運作所需的基本組件之外，額外配置的一些組件。備份的目的是在系統組件發生故障之時，可以利用備份組件校正錯誤，或是用作故障組件的替換。在一個軟體系統中備份的範圍包括軟體程式碼本身，程式運作所需要的資料，以及重算等技巧。在傳統的軟體容錯領域中程式碼的備份是沒有意義的，然而在元件式架構系統中則可以做到軟硬體混合式的備份：負責某項服務的元件可以複製成多份放在不同的機器上，同時利用 heart beat 訊號等技巧互相檢視運作狀況。如此一來就可以達成備份的目的：當其中一個元件失效時，可以使用其他的元件取代。

要達到這個目的，備份與主要元件之間必須能夠共有一致的程式參數狀態，如此在備份元件啟用時才能接續服務現在正在使用中的客戶端程式。對於一個與狀態相關的伺服器應用程式來說，在正常運作時即需對其關鍵性的資料做查核點

(checkpoint)，以便讓備份能夠回復。

一般關於查核點設置的研究多著重於建立查核點的時機，以及建置查核點的資料。以平行分散式系統來說，查核點的設置必須能達到一致性(consistence)的資料點，如此才能用後退回復的方式繼續執行運算工作。頻繁的查核點設定很顯然的比較容易在多個不同的執行緒中取得一致性的資料點，然而查核點的設置本身即會佔用部份系統資源，而對於系統效率有實際而明顯的影響。現今研究的趨勢是以系統事件(event)的方式進行持續性的查核紀錄(incremental)，並將研究的焦點放在執行緒與網路和儲存媒體的互動上。但在實際執行卻缺乏一套統一的範式 (infrastructure)。元件程式的出現使得這個問題變得更為複雜。由於元件本身資料封裝的特性，比較難制定查核點所必須涵蓋的元件範圍與負責執行查核點的元件。

本計畫使用 Java 程式語言進行系統開發，使用 CORBA 作為元件的框架，並使用 XML 作為抽象資料形態的標準格式。Java 程式語言本身即是完全物件導向，同時 Java 程式使用虛擬機器(virtual machine)的做法使得一個編輯完成的 Java 程式即可在不同機器上被執行。CORBA 是共通物件呼叫架構(Common Object Request Broker Architecture)的簡寫，由 OMG 聯盟所提出的一個元件式框架。CORBA 本身具有完整的界面定義與物件函式庫(CORBA Facilities & CORBA Services)，目前業界有不少 CORBA 實作的框架套件可以選用。部份 CORBA 框架包含有容錯群組的機能，可配合查核點與後退回復服務一併運作，以達到更高的穩定性。

四、結論與討論

本年度計畫我們預計完成之工作項目為包含：

1. 參考真實交易狀況制定決策邏輯與流程控制邏輯
2. 設計仿真實交易狀況的模型，進行分析模擬
3. 利用 CORBA 框架容錯服務設計元件備份機制
4. 利用 CORBA Service 中的常態性物件服務與交易服務進行查核點建立
5. 利用交易服務查核點進行資料與服務回復研究

目前我們已進行決策邏輯的規劃，並以 CORBA 技術實作元件備份機制與查核點的建立，以及模擬錯誤發生時的系統回復機制設計。

五、參考文獻

- [1] Andrew S. Tanenbaum, "Computer Networks", Prentice Hall Inc., 1996
- [2] Thomas Wierlemann & Thorsten Kassing, "Performance of TCP/IP and its Application Protocols over Narrowband Bearers with high Latency",

- <http://www.w3.org/1998/11/05/WC-workshop/Papers/wierlemann.html>
- [3] Hunter & Crawford, “Java Servlet Programming”, O’Reilly, 1999
 - [4] Daniel J. Berg, “Advanced Techniques for Java Developers”, John Wiley & Sons, Inc., 1998
 - [5] Michael S. Jenkins, “Abstract Data Types in Java”, McGraw-Hill, 1998
 - [6] Matthew Siple, “Java Database Programming”, McGraw-Hill, 1998
 - [7] Elliotte Rusty Harold, “Java Network Programming”, O’Reilly, 1997
 - [8] David S. Frankel, “CORBA components”, JavaReport, Oct. 1999
 - [9] W. Richard Stevens, “Advanced Programming in the UNIX Environment”, Addison-Wesley, 1998
 - [10] Thomas J. Mowbray & William A. Ruh, “Inside CORBA”, Addison-Wesley, 1998
 - [11] Markus Voelter, “Aspect-Oriented Programming in Java”, JavaReport, Jan. 2000
 - [12] Richard Deadman, “XML as Distributed Application Protocol”, JavaReport, Oct. 1999
 - [13] William J. Pardi, “XML in Action Web Technology”, Microsoft Press, 1998
 - [14] Danny Goodman, “Dynamic HTML”, O’Reilly, 1998
 - [15] “Mobile Network Computing Reference Specification Consortium”,
<http://www.mncrs.org/>
 - [16] “Considerations for Web Transaction Security”, RFC2084,
<http://www.cis.ohio-state.edu/rfc/rfc2084.txt>
 - [17] “HTTP State Management Mechanism”, RFC2109,
<http://www.cis.ohio-state.edu/rfc/rfc2109.txt>
 - [18] “HTTP Spec”, <http://www.w3.org/Protocols/>
 - [19] “XML Spec”, <http://www.w3.org/XML/>
 - [20] “WAP – W3C White Paper”, <http://www.w3.org/TR/NOTE-WAP>
 - [21] IPng Mobility Considerations, RFC1688,
<http://www.cis.ohio-state.edu/rfc/rfc1688.txt>