

一個分散式資料庫結構及其資料查詢

A Database Distributed Database Architecture and its Query

呂芳禛
Fang-Yie Leu

東海大學資訊科學系
Department of Computer and
Information Sciences
TungHai University,
Taichung, Taiwan, R.O.C.

趙尚威
Shiang-Wei Chao

中正大學資訊工程所
Institute of Computer Science and
Information Engineering
National Chung Cheng University,
Chiayi, Taiwan, R.O.C.

摘要

分散式資料庫在目前的理論與應用上都存在著很大的瓶頸，本文擬提出一種新的架構，將虛擬機器(virtual machine)的特性加入分散式資料庫，我們稱之為分散式資料庫結構(Distributed Database Architecture, DDA)並在DDA的結構中，配合資料字典來建構分散式的查詢分解與整合之演算法。

關鍵字：查詢解析，資料整合，資料字典，分散式資料庫系統

Abstract

Distributed database currently still has bottlenecks. We propose a new architecture, called "Distributed Database Architecture", which gets some benefits. We try to build distributed data query decomposition and integration algorithms on this architecture.

Keywords: query decomposition, data integration, data dictionary, distributed database.

I. 前言

近年來電腦網路(computer network)的廣泛使用已高度地影響了資訊技術的發展，從作業系統[1-3]、系統程式[4]到資料庫系統[5-8]無不面臨這個新的挑戰。由網路而造成的分散式(distributed)環境也引發了若干中央式資料庫(centralized database)未曾遇到的新問題，也為資料庫領域帶來新的技術需求。

分散式資料庫的理論研究至今仍有許多的議題尚待突破，例如分散式資料庫的同步(synchronization)及確認(commit)等。這些方面雖

然已有部份理論的發表，例如 two-phase lock 及 two-phase commit 等，但由於網路頻寬(bandwidth)不足造成的延遲效應，以及系統的容錯能力無法滿足實際需要等問題，分散式資料庫至今在發展上依然有困難。儘管如此，仍有若干機構推出實驗性質的資料庫系統，例如 SDD-1 [7, 9]、R* [7, 10]、分散式 INGRES [7, 11] 等，但事實上它們的功能離商業化的要求還有一大段的距離。

本文便是試圖提出一種新的結構，並在此結構上製作演算法。在第二章中，我們將探討目前之研究現況；第三章介紹分散式系統架構；第四章為查詢的分解與資料的整合敘述；第五章為本文之結論。

II. 研究現況

分散式資料庫與網路資料庫目前在應用方面並不算成熟，以下略述其現況：

目前商業界所號稱的分散式資料庫系統，其實都僅止於在一般的資料庫中添加部份分散式的處理功能，例如 ORACLE7™ 的分散式資料庫任選(distributed database option) [12]，IBM DB2™ 的分散式資料設施(distributed data facility) [13]，及 INGRES/STAR™ [5] 等等皆是。

以 INGRES/STAR 為例，該架構中利用一具 INGRES/STAR server 來專門進行資料的分解與合併功能，並且在每一個節點(node)上加裝一個 GCA(global communication architecture)作為共同聯繫的機制。GCA 的加入在該架構中是一個突破，但 INGRES/STAR server 的存在仍限制了分散式的架構，並且容易造成網路流量與機器工作量的負擔[5]。

原有的中央式資料庫，為了因應網路的大量應用，紛紛開發功能較強的連結界面(connectivity interface)，例如微軟(Microsoft)公司的 ODBC (Open DataBase Connectivity)、ActiveX 界面，以及昇陽(Sun

Microsystem) 公司所推出的 Java-based 界面 JDBC (Java DataBase Connectivity), 都是提供一種統一的規格, 讓各種資料庫得以透過它們存取資料。但是此一趨勢僅停留在界面的發展, 仍未考慮分散式的處理能力, 也就是對於分散在各工作站 (sites) 的資料沒有提供結合 (join)、卡笛氏乘積 (Cartesian product)、投影 (project), 以及組合、分析的能力。

此外, 另有一類型處理方式稱為多重資料庫系統 (multidatabase system)。多重資料庫系統乃是因應業界既有的資料庫整合問題而產生, 主要的目的在於整合企業中正在使用的各種資料庫系統, 使得既有的資料庫可以藉由多重資料庫管理系統 (multidatabase management system) 而整合, 俾使全域使用者 (global user) 及區域使用者 (local user) 都可以同時使用該資料庫 [5]。

本文擬提出一個三層式 (three-tire) 的分散式資料庫結構 (Distributed Database Architecture, DDA), 請參看圖一。本文將以此架構為基礎, 配合資料字典及查詢處理演算法, 以解決表格 (tables) 的分散化之後的搜尋與整合問題。

III. 分散式資料庫結構

DDA 由從端 (client)、分散式資料庫伺服器 (Distributed Database Server, DDS) 及本地資料庫管理器 (Local DataBase Manager, LDBM) 三部分所組成。從端內部具有若干應用程式, 負責接受使用者的查詢與指令, 並顯示執行結果, 是使用者與系統間的界面。

DDS 是三層式架構中的中間層, 其主要工作包括:

- (1) 接受來自從端的查詢。
- (2) 分析及分解查詢, 並將分解後的子查詢傳送至擁有各資料的工作站上。
- (3) 接受各個伺服器傳回的執行結果。
- (4) 整合所有的執行結果, 經整理後回送給從端。

第三層包括 LDBM 與 DBMS (Data Base Management System) 兩部份。LDBM 的主要功能是執行 DDS 送過來的指令, 並呼叫 DBMS 來查詢本地資料庫 (local database)。其結果再由 LDBM 傳回給 DDS。

在系統中, LDBM 則隨每個工作站所使用的 DBMS 而各有不同; 而每一個工作站都具有一份相同的 DDS, 但每一種 LDBM, 它與上層的 DDS 都以相同的協定來溝通。實際上, LDBM 在不同的平臺上扮演著虛擬機器 (virtual machine) 的角色, 這種特性使得此架構具有跨 DBMS 的優點。DDS 能藉由 LDBM 跨越各種 DBMS 及作業系統, 使這樣的分散式資料庫系統具有相當高度的彈性。除此之外, DDA 架構還有下列的特性:

- A. DDS 分擔了部份的功能, 因而降低了從端的負荷。
- B. 使用者與系統維護者不需要學習各種 DBMS 語言, 只要熟悉 DDS 所提供的語法就可以取得與更改資訊。
- C. 系統的擴充性強, 當系統要安裝新的資料庫系統時, 只須在該資料庫上加裝 DDS 與可讀寫該資料庫的 LDBM 即可; 當然, 記錄該資料庫中資料的資料字典, 也必須拷貝到系統內各 DDS 中。
- D. 每一個工作站上的 DDS 都可以接受來自從端的查詢而成為主 DDS (master DDS), 主 DDS 具有主控權, 負責處理該次查詢之交易 (transaction), 這種做法可以降低從端和 DDS 之間的網路流量 (network traffic)。
- E. 從端不能直接存取本地資料庫, 它必須透過 LDBM 的安全管制, 以間接的方式來存取或查詢, 安全性比主從架構高。

IV. 查詢分解與資料整合

一個關聯式資料庫 (relational database), 是由若干表格所組成, 而分散式資料庫中, 每一個表格又由分散在不同地方的若干子表格 (sub-table) 所組成。分散式資料庫必須知道該從何處取得所需的資料, 也必須將取自各工作站的資料整合起來。基於透明化的原則, 這些資訊必須由資料庫本身來提供, 因此需要資料字典的協助, 表一是一個資料字典的範例。

再者, 使用者查詢的對象是整個資料庫, 但是資料是分散的, 所以查詢勢必須要加以解析, 以取得各個部份的資料。在 DDA 之中, 資料的分解與整合主要是由 DDS 來處理, 以下便陳述 DDS 解析查詢的演算法與說明:

分解及分析子查詢之演算法:

輸入 使用者之查詢 Q

輸出 修正後送給工作站 i 之子查詢 $Q_i, i=1, 2, 3, \dots, m$.

Begin

1. 取出 Q 之 from 子句中所有表格所對應的資料字典;
2. 以資料字典列出本次查詢所涉及的所有工作站編號 i, 假設 $i=1, 2, 3, \dots, m$, 並記錄工作站總數為 m;
3. 將 Q 複製成 m 份, 每一份都是一個子查詢 $Q_i, i=1, 2, 3, \dots, m$;
4. for (每一個子查詢 $Q_i, i=1, 2, 3, \dots, m$) {
 - 4.1 if (Q_i 之 from 子句中表格 T 的所有屬性皆不在工作站 i 中)

// 從資料字典的垂直條件中可得知
從 from 子句中刪除 T;
 - 4.2 if (Q_i 之 select 子句中屬性 X

```

    不在工作站 i 中)
    從 select 子句中刪除 X ;
    // X 不符合其垂直條件
4.3 if ( from 子句中表格 T 的主鍵
    屬性 Y 不在 Qi 的 select 子句中)
    將 Y 加入 select 子句中 ;
    // 未來結合子表格之用
4.4 將 where 子句中的敘述分解成
    C1, C2 ... Cn 等 n 個簡單
    述詞 ( simple predicates ) ;
4.5 for s = 1 to n
    { if ( Cs 為條件型敘述)
    // 屬性  $\Theta$  某值
    { if ( ( C = (Cs 'AND' 資料字
    典之水平條件 ) ) =  $\emptyset$  )
    {
    刪除 Cs 所在的最小項述詞 ( minterm predicate ) ;
    if ( where 子句已無最小項述詞存在)
    // where 子句已空了
    break ;
    }
    else 在 where 子句中以 C 取代 Cs ;
    }
    else if ( Cs 為連結型敘述 )
    // 假設為：屬性 A  $\Theta$  屬性 B
    { if ( 屬性 A 在工作站 i 中 )
    // 屬性 A 符合其垂直條件
    將 A 移至 select 子句中 ;
    // 未來結合子表格之用
    else if ( 屬性 B 在工作站 i 中 )
    // 屬性 B 符合其垂直條件
    將 B 移至 select 子句中 ;
    // 未來結合子表格之用
    刪除 Cs ; } } }
4.6 if ( C1 , C2 ... Cn 皆被刪除)
    { 取出 select 子句中各屬性 X ;
    if ( X 只存在於工作站 i 中 )
    // 其垂直條件為 site i
    刪除 where 子句 ;
    // 即 where 子句 always true
    else 刪除 Qi ; }
5. 將未被刪除之 Qi 送至工作站 i ;
End.

```

(4) 分散式資料的整合

資料整合的方式是將數個傳回來的子表格利用結合、聯集及取代等操作完成的，其演算法如下：

整合演算法：

輸入 各工作站送回來之子表格及

原始之使用者查詢

輸出 查詢結果

Begin

1 for(所有屬於同一表格之子表格)

repeat

for(所有可以聯集的任兩子表格 R 與 S)

// 原本是水平切割

{for(所有 R 或 S 中具有

相同關鍵屬性的任兩記錄 X, Y)

將 X 和 Y 的屬性 'OR' 起來；

外部聯集 (outer union) R 和 S ；

}

for(所有可以結合的任兩子表格 R 與 S)

依主鍵屬性結合之 ；

// 原本是垂直切割

until (無法再聯集或結合)

if (表格中尚有不完整紀錄)

{ printf("查詢有錯誤")；

停止 ；

}

2. 以上述步驟所整合之表格，執行使用者

之原始查詢；

End.

V. 結語與討論

本文提出之分散式資料庫架構 DDA，應用了虛擬機器的觀念來擴大各區域資料庫的相容性，且藉由主從 DDS 的結構來平均各 DDS 的工作負載。藉著 DDS 與 LDBM 的協定，更可以輕易地擴充系統中的資料庫，具有多方面的優點，對於時下的各種分散式資料庫應用將有極大的裨益。

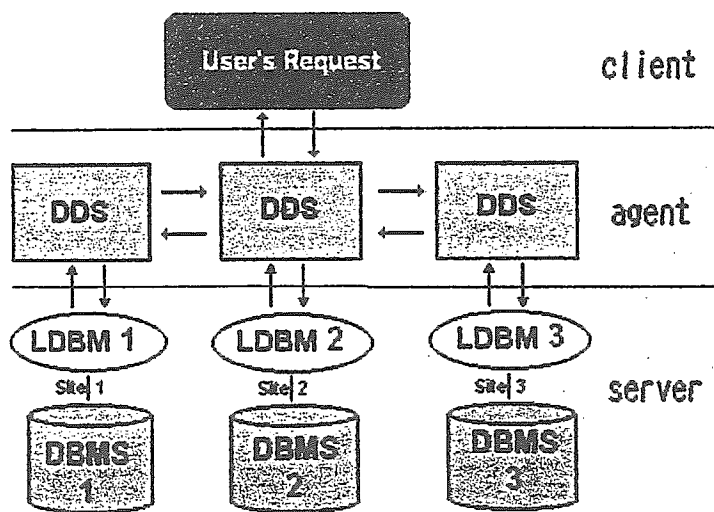
文中的分解演算法，將各子要求的內容作適應處理，使子要求送至各指定工作站後，能消除不存在的查詢敘述，進一步能儘量減少傳回結果的資料量。而在整合演算法的部份，系統將各處的子表格收回之後，重覆結合與合併的步驟，使原本分散於各區域工作站中的資料藉由鍵值來結合，完全還原成中央式的表格，最後系統就可以用還原完成的表格來作中央式的處理。

此外，在上述處理過程中，資料字典扮演著極其重要的角色，所以資料字典的結構設計也是至為重要的一環。另一方面，本研究亦發現，這一連串的查詢處理過程，同時適用於在中央式資料庫視野 (view) 的處理 [8]，因為視野也是由若干表格作選擇、投影、結合、聯集而成。但在分散式資料庫中視野的查詢處理，將更複雜，更難解決。

再者，在子表格整合完成之後，如果發現有資料不完整的情形時，該如何對系統做容錯查詢，仍需未來繼續研究之。

VI. 參考文獻

1. A. Goscinski, Distributed Operating System : The Logical Design, Addison Wesley, 1991.
2. A.S. Tanenbaum , Distributed Operating Systems, Prentice Hall, 1995.
3. G. Coulouris and J. Dollimore , and T. Kindberg , Distributed System : Concepts and Design, 2nd ed. , Addison Wesley, 1995.
4. K. Arnold and J. Gosling , The Java Programming Language, Addison.Wesley, 1996.
5. David Bell, Jane Grimson, Distributed Database Systems, Addison-Wesley, 1992.
6. C. J. Date, An Introduction to Database Systems, 6th ed, Addison-Wesley, 1995.
7. M. T. Ozsu and P. Valduriez, Principles of Distributed Database System, Prentice Hall, 1991.
8. R. Elmasri and Navathe, Fundamentals of Database Systems, 4th ed., Benjamin Cummings, 1994.
9. P.A. Bernstein and D.M. Chiu, "Using Semi-Joins to Solve Relational Queries," J. ACM, Jan. 1981, 28(1): 25-40.
10. P.G. Selinger, et al, "Access Path Selection in a Relational Database Management System," in Proc. ACM, SIGMOD, Int. Conf. on Management of Data, Boston, Mass., May 1979, PP 23 -34.
11. R. Epstein, M. Stonebraker, and E. Wong, "Query Processing in a Distributed Relational Database System," in Proc. ACM SIGMOD, Int. Conf. on Management of Data, Austin, Tex., May 1978.
12. Oracle Corp., OracleTM7 Server Application Developer's Guide, 1992.
13. C.J. Date and C.J. White, An Introduction to DB2, 2nd. Ed., Addison Wesley, 1988.



圖一 分散式資料庫結構之細部結構

| | | | | | | |
|-------------------|--------------|--------------|-------------|-------------|--|---------|
| TABLE 'Person' | <u>Name</u> | Address | Dno | Age | TEL | Email |
| | H: NULL | H: NULL | H:NULL | H:NULL | H: if FirstNum(Tel)=3 Site1 ; else Site2 ; | H:NULL |
| | V: Not Site3 | V: Not Site3 | V:Not Site3 | V:Not Site3 | V: Not Site3 | V:Site2 |

| | | | |
|-----------------------|----------------|-----------|-----------|
| TABLE 'Department' | <u>Dnumber</u> | Dname | Budget |
| | H: NULL | H:NULL | H:NULL |
| | V: Site 3 | V: Site 3 | V: Site 3 |

表 一 資料字典範例