200928813

發明專利說明書

（本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫）

※ 申請案號：97135340

※ 申請日期：97 年 9 月 15 日　　**※IPC 分類：**

G06F 17/30 (2006.01)

一、 發明名稱：（中文/英文）

於附屬高效能平行電腦中執行資訊密集資料庫使用者定義程式的系統及方法

SYSTEM AND METHOD FOR EXECUTING COMPUTER-INTENSIVE DATABASE USER-DEFINED PROGRAMS ON AN ATTACHED HIGH-PERFORMANCE PARALLEL COMPUTER

二、 申請人：（共 1 人）

姓名或名稱：（中文/英文）（簽章）

萬國商業機器公司

INTERNATIONAL BUSINESS MACHINES CORPORATION

代表人：（中文/英文）（簽章）

琳奈　D　安德森 / ANDERSON, LYNNE D.

住居所或營業所地址：（中文/英文）

美國紐約州 10504 亞芒克市新奧爾察德路

New Orchard Road, Armonk, NY 10504, U.S.A.

國籍：（中文/英文）　美國 / US

三、 發明人：（共 2 人）

姓名（中文/英文）

1. 拉麥許　那塔拉娟 / NATARAJAN, RAMESH

2. 麥可　安祖亞斯　柯齊特 / KOCHTE, MICHAEL ANDREAS

國籍（中文/英文）

1.美國 / US、2.德國 / DE

四、 聲明事項：

☐ 主張專利法第二十二條第二項☐第一款或☐第二款規定之事實，其事實發生日期為： 年 月 日。

☑ 申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號 順序註記】

☑ 有主張專利法第二十七條第一項國際優先權：

1. 美國、西元 2007 年 09 月 17 日、11/856,130
2.

☐ 無主張專利法第二十七條第一項國際優先權：

☐ 主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

☐ 主張專利法第三十條生物材料：

☐ 須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

☐ 不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

九、發明說明：

【發明所屬之技術領域】

本發明一般有關資料庫處理的效能增進，尤其有關一種在附屬高效能平行電腦(HPC)系統上調度及執行查詢工作流程之計算密集部分以加速計算密集資料庫查詢的系統及方法。

【先前技術】

本發明有關一種在附屬高效能平行電腦(HPC)系統上調度及執行查詢工作流程之相關計算密集部分以加速資料庫查詢的系統及方法。

【發明內容】

本發明動機係因觀察到，傳統上用於交易處理應用、線上分析及資料倉儲的商用資料庫在儲存、查詢、及分析各種複雜資料類型(諸如文字、影像、及多媒體)上的使用日益增加。商用資料庫在處理科學儀器的原始事件流或儲存高效能電腦模擬之未處理結果集上的使用亦日益增加(見 J. Becla 及 D. L. Wong「管理千兆位元組學習心得(Lessons Learned From Managing a Petabyte)」，創新資料系統研討會(Conference On Innovative Data Systems Research)，加州 Asilomar(2005 年))。

關於此在資料庫中儲存為複雜資料類型之原始資料的相關分析常常遠超出簡單的歸檔及擷取範圍，以包括特定計算密集操作及資料轉換，其一般可在採用此資料的各種外部應用程式上使用，諸如高階語義查詢及搜尋、基於內容的索引、精密資料模型化、資料採擷分析、及電腦輔助設計。可將這些計算密集操作及資料轉換建置為資料庫擴充項內的內建程式，其包含這些複雜資料類型的使用者定義儲存程序或使用者定義函數集合，以提供原始複雜資料類型至適於精密外部應用程式之表示法的必要轉換。結果，這些內建資料庫擴充項(可由熟習本技術者建置)提供外部應用程式開發者相關的功能及轉換，以在其應用程式中使用這些複雜資料類型、使用熟悉的集合導向或基於 SQL 的語法及查詢介面來調用這些轉換。此外，在將資料傳輸至用戶端應用程式之前，藉由提供原始資料之比較壓縮的表示法，或藉由對資料庫伺服器本身的原始資料進行實質的預先過濾，使用內建使用者定義程式常可降低網路上從資料庫伺服器移動原始資料至用戶端應用程式的耗用。最後，藉由提供對原始資料不需要複製或與外部應用程式共用之「類物件」的介面，使用內建使用者定義程式更容易確保資料庫內原始資料的隱私性、完整性、及連貫性。

然而，儘管有上述優點，對於執行資料庫伺服器上計算密集、使用者定義程式的相關處理需求仍然極大，且目前在習用資料庫效能基準中，或在通用資料庫伺服器系統硬體平台的設計及大小中，針對此方面加以解決的少之又少。

大型商用資料庫系統通常駐存於共用記憶體多處理器或網路叢集電腦平台上。在這些平台上，通常僅對協調由查詢最佳化工具所產生之平行查詢計劃的執行的資料庫控制器軟體公開基礎平行性，同時，通常不對任何應用軟體或內建使用者定義程式公開此基礎平行性。在一些情況中，可在由查詢最佳化工具所產生的平行查詢計劃上默許附帶使用者定義函數的執行，以利用多緒或資料分割平行性來排程查詢執行。然而，商用資料庫常常會對使用者定義函數加上預設限制，或甚至明示不允許多種使用者定義函數依此方式的默許平行執行。例如，對於平行執行的預設限制常常應用於以下使用者定義函數：使用便條紙記憶體儲存反覆函數調用間的資訊、實行諸如檔案輸入-輸出操作的外部動作、或涉及非決定性執行(即，其中該函數對於相同輸入可能傳回不同的輸出值，如亂數產生器)，或針對每一函數調用傳回多列值的使用者定義表函數(有關特定商用資料庫之此等預設限制的詳細論述，見 D.

Chamberlin「DB2 通用資料庫完全指南(A Complete Guide to DB2 Universal Database)」第 6 章，Morgan-Kaufman 出版社，舊金山，1998 年)。此外，雖然程式設計師在將「安全」序列語義保留於默許平行執行中的特定情況中可撤銷這些預設限制，但即使應用程式能夠利用更高等級細密度的平行性，可用於執行這些使用者定義程式的平行性等級卻仍受限於下列預先組態參數：資料庫平台中指定共用記憶體平台上的最大緒數目，或分散式叢集平台中資料分割區或處理器的最大數目。此外，雖然可將這些資料庫組態參數設定為基礎硬體平台所支援的最大值，但即使在此範圍內，對於每一個別資料庫應用程式為最佳的平行細密度仍將取決於涉及以下因素的複雜相互作用：每一應用程式中平行協調的等級、同步化、載入平衡及資料移動，因此不可能有一種通用設定對於所有在資料庫伺服器上運行的應用程式都是最佳的。最後，在此情況下，即使對於單一的應用程式，要在現有硬體平行性所加諸的限制外，提高資料庫效能，將需要對整個資料庫平台進行全面且昂貴的升級。

總之，因此，現有的商用資料庫系統只對查詢處理引擎及資料庫控制器公開基礎控制或資料平行性。這些資料庫系統不提供以下作用的特定應用程

式設計介面(API)：用於寫入通用、平行、使用者定義儲存程序及使用者定義函數；或用於根據不同的情況，在資料庫平台預組態限制內或延伸超出資料庫平台預組態限制，調整個別應用程式的可擴充效能。

已根據在較為一般性的資料庫伺服器平台內使用專用硬體加速器，為提高資料庫查詢處理效能提出許多建議。例如，K. C. Lee、T. M. Hickey、及V. W. Mak的「用於大型資料庫系統的VLSI加速器(VLSI Accelerators for Large Database Systems)」(IEEE Micro，第11卷，第8-20頁，1991年)收集資料庫查詢工作負載的研究統計以識別最昂貴的操作，並提出在磁碟儲存介面及CPU間的資料路徑中使用專用VLSI硬體篩選器來掌控這些特定操作(其包括聯合搜尋及彙總操作)。在P. Faudemay及M. Mhiri的「用於大型資料庫的聯合加速器(An Associative Accelerator for Large Databases)」(IEEE Micro，第11卷，第22-34頁)，及M. Abdelguerfi及A. K. Sood的「用於相關資料庫彙總操作的細密構造(A Fine-Grain Architecture for Relational Database Aggregation Operations)」(IEEE Micro，第11卷，第35-43頁)亦支持同樣的主張。文字導向資料庫應用程式(特別是用於字串及型樣匹配)的硬體

加速器之使用說明於 V. W. Mak、K. C. Lee、及 O. Frieder 的「利用型樣匹配的平行性：一種資訊擷取應用程式」，資訊系統的 ACM 會刊 (ACM Transactions on Information Systems)，第 9 卷，第 52-74 頁，1991 年。此方法比較近期的發展是「主動磁碟」技術，其利用逐漸取代磁碟控制器介面之自訂設計電路的通用微處理器 (E. Riedel、C. Faloutsos、G. A. Gibson 及 D. Nagle「用於大型資料處理的主動磁碟(Active Disks for Large-Scale Data Processing)」，IEEE 電腦，第 34 卷，第 68-74 頁，2001 年)。在此方法中，將某些查詢處理工作負載(一般在資料庫伺服器的主 CPU 上實行)卸載至這些在磁碟控制器介面處的個別微處理器。此方法在通常用於商用資料庫中的多磁碟系統中，利用儲存介面處之更高程度的平行性，以經由用於許多資料庫查詢的儲存系統互連傳送到主 CPU 的資料卷中達成實質預先過濾及縮減。對於可以此方式卸載的工作負載性質存在諸多限制，尤其由於個別的磁碟控制器無法彼此通信，卸載工作將限制在對其相應的資料流進行簡單的資料過濾及轉換操作。總之，雖然內建硬體加速器的使用對於簡單資料類型的簡單述詞處理非常有效，但整體方法對於需要平行同步化及通信的較複雜操作，仍然缺少彈性及可程式性。

以上論述促成以下發明需求：在單獨及有所區別的 HPC 平台上而非在效能有限的資料庫伺服器上執行計算密集、平行使用者定義程式。如先前所提，此方法的主要效能限制是在資料庫伺服器及HPC平台間之資料移動的耗用，即使對於極為長程的計算，HPC平台的計算效能增益仍可顯著抵銷或分攤這些資料傳輸耗用。此方法的一個問題是一般的資料庫使用者可能需要專門技術才能在 HPC 平台上執行所需程式。此外，此方法很難用這些平行使用者定義程式在SQL架構內撰寫複雜的資料庫查詢，因為這需要使用者使用特定且非自動的方法，對外部 HPC 平台上的必要計算密集操作做出明確的排程。

本發明因此是根據使用此外部 HPC 平台作為資料庫平台的後端計算伺服器(致使應用程式的終端使用者在查詢執行程序中，大體上與此HPC平台的使用分開，不像前一段所描述的前端組態)。儘管這麼做並無法消除在資料庫伺服器及後端 HPC 平台之間移動資料及結果的效能損失，但可使用各種資料快取策略確保在此方法中達成相同的效能等級，因為在專業程式設計師所建置的等效前端用戶端中，對於資料的移動及管理進行了明確的管理及最佳化。此外，本發明的資料移動是在嚴密控制的

系統環境中(包含資料庫伺服器及後端平行電腦系統)進行，因而很容易引用資料庫邏輯來確保資料的完整性及連貫性，或很容易使用專用硬體及協定來提高資料庫伺服器及後端 HPC 平台間的資料傳輸效能。本發明所需的「開發者」專門技術係用於程式化及安裝後端 HPC 系統上的計算服務，而在此完成後，只要是有關用戶端應用程式或終端使用者，整體查詢執行即如同等效的內建使用者定義程式已在資料庫伺服器本身上執行般進行。總之，履行查詢所需的各種步驟，包括需要的資料移動、計算密集操作的卸載、及結果的傳回，全都以自動方式來進行，不需要任何明確的使用者排程或同步化。因此，本發明所採用的方法，不管使用卸載的使用者定義程式與否，均不會影響在一般 SQL 查詢處理架構內撰寫複雜資料庫查詢的重要能力(在下文說明的本發明特定具體實施例中將論述此項目的一個範例)。

本發明的適用性可應用於若干應用程式領域中，尤其是有關於生物資訊學及生命科學的領域，下文說明的本發明特定具體實施例即以此領域為主。

本文所考慮的此特定具體實施例係在於序列相

似度及 DNA 比對及蛋白質序列資料庫所使用的演算法。近年來，基因及蛋白質序列的資料量快速成長，目前是使用各種格式，將此資料儲存於各種資料貯存器(包括商用關係資料庫及專有的非關係資料庫)中。生物資訊學的一個重要工作是在現有的序列貯存器中，對照序列子集比較新的序列或序列片斷，以找出序列的相似性或同源性。然後，將所得匹配與精確匹配序列上的其他科學資料及元資料結合(諸如構形及結構細節、實驗性資料、功能性註解等)，以提供資訊對新的序列做出進一步的生物學或基因學研究。由於此程序中許多步驟都需要資訊的整合及彙總，如果可從 SQL 查詢介面存取此序列資料及元資料全部以及序列匹配演算法，對此工作將有極大的助益。達成此點的一個方法(常稱為擷取/變換/載入方法)是將原始資料格式的相關序列程式庫匯入商用關係資料庫，這對於原始序列程式庫及元資料所儲存的每一專有資料格式，將需要自訂載入器指令碼。另一種供選擇的方法，如說明於 L. M. Haas、P. M. Schwarz、P. Kodali、E. Kotler、J. E. Rice、及 W. C. Swope 的「DiscoveryLink：一種對生命科學資料服務之整合式存取的系統(DiscoveryLink: A System for Integrated Accesss to Life Sciences Data Services)」(IBM 系統期刊(IBM Systems Journal)，第 40 卷，第 489-511 頁，2001

年)，其將序列資料保留在其原始資料貯存器中，但將此異質資料來源集的摘要或聯合觀點提供於主要的前端資料庫伺服器上，其中在此主要前端資料庫上的內建包裝函數集對於將在主要資料庫及後端異質資料來源集間交換的輸入查詢及查詢結果，提供必要的映射。

這兩個一般可供選擇的方法亦可用來使用資料庫伺服器中的 SQL 查詢介面，調用各種生物學序列匹配演算法。例如，可將這些演算法建置為內建使用者定義程式，如 S. M. Stephens、J. Y. Chen、M. G. Davidson、S. Thomas 及 B. M. Trute 的「Oracle 資料庫 10g：用於生命科學之 BLAST 搜尋及規則運算式型樣匹配的平台(Oracle Database 10g: a platform for BLAST search and Regular Expression pattern matching in life sciences)」(核酸研究(Nucleic Acids Research)，第 33 卷，資料庫議題(Database issue)，第 D675-D679 頁，2005 年)對於特定 BLAST 演算法所做的說明。或者，可延伸上述資料庫包裝方法，如 B. Eckman 及 D. Del Prete 的「使用 IBM DB2 資訊積分器對 BLAST 的高效率存取(Efficient Access to BLAST Using IBM DB2 Information Integrator)」(IBM 保健及生命科學刊物(IBM Healthcare and Life Science Publication)，2004 年)所說明，以在單獨

BLAST 伺服器上啟始必要的運算，及將結果映射回到資料庫伺服器上的表格中。這兩個方法在建置細節上大體上頗為不同，但二者實質上提供一些重要的功能，即使用資料庫 SQL 查詢介面存取及查詢一或多個含有生物學序列資料及元資料之資料來源的能力，及將序列匹配演算法(諸如 BLAST)整合至這些資料庫查詢中的能力。這些功能提供應用程式開發者產生複雜查詢的能力，諸如使用涉及序列元資料的述詞過濾序列的初始搜尋空間，及藉由連結從匹配演算法傳回的最高排序序列與這些在其他相關資料貯存器中之序列的資訊，對序列匹配結果進行後續處理。以此方式，序列匹配演算法的內建建置提供應用程式自動化、增強及加快序列資料之新科學發現程序的功能。然而，上述兩種方法均未在商用資料庫中以一般方式開發，以支援這些序列匹配演算法的平行建置。

有相當多的先前技術投入生物學序列匹配及比對之平行演算法的開發中，已在範圍涵蓋專用加速器至分散式記憶體電腦的多緒對稱多處理系統的各種 HPC 平台上建置這些平行演算法。

從可擴充性的觀點來看，分散式記憶體平台是最引人注目的，及在此情況中，有兩種主要的方法

利用生物學序列匹配演算法之平行性。

第一種方法稱為資料庫分段，其分割計算節點集合上的目標序列程式庫(較佳是使用足夠的計算節點，致使序列程式庫的每一個別分割區都能分配在節點記憶體內)。此方法的平行可擴充性最後受限於散布程式庫序列資料及在較大計算節點集合上收集結果的資料移動耗用。建置此分散式記憶體平行方法所需之效能最佳化的研究請見 A. E. Darling、L. Carey、W. Feng 的「mpiBLAST 的設計、建置及評估(The Design, Implementation and Evaluation of mpiBLAST)」(叢集世界研討會(Clusterworld conference)會議記錄，2003 年)，及最佳化平行磁碟 I/O 效能的擴充請見 H. Lin、X. Ma、P. Chandramohan、A. Geist 及 N. Samatova 的「平行 Blast 有效資料存取(Efficient data access for parallel blast)」(國際平行及分散式處理討論會(International Parallel and Distributed Processing Symposium)會議記錄，2005 年)。

第二種方法稱為查詢分段，其可在出現一批相似但獨立的查詢時使用，致使可對照目標序列程式庫同時平行執行每一查詢。因此，可在分散式記憶體平台的多個節點上複製此目標序列程式庫，如說

明於 R.C. Braun、K.T. Pedretti、T.L. Casavant、T.E. Scheetz、C.L. Birkett、及 C.A. Roberts 的「對工作站叢集上的區域 BLAST 服務進行平行化的三個補充方法 (Three Complememtary Approaches to Parallelization of Local BLAST Service on Workstation Clusters)」(第五屆國際平行計算技術 (PACT) 研討會 (5th International Conference on Parallel Computing Technologies (PACT)) 會議記錄，電腦科學演講稿(Lecture Notes in Computer Science，LNCS)，第 1662 卷，第 271-282 頁，1999 年)。此方法受到個別節點上可能不夠儲存整個目標序列程式庫之記憶體的限制，但此特定難處可藉由以下方式克服：使用資料庫及查詢分段的組合，這對於具有成千上萬個處理器的分散式記憶體平行電腦，是最有效及可擴充的方法，如說明於 H. Rangwala、E. Lantz、R. Musselman、K. Pinnow、B. Smith 及 B. Wallenfelt 的「用於藍色基因/L(Blue Gene/L) 的大量平行 BLAST(Massively Parallel BLAST for the Blue Gene/L)」(高可用性及效能計算工作坊 (High Availability and Performance Computing Workshop)，Santa Fe NM，2005 年)。

已知 BLAST(或其他序列匹配演算法)的平行建置均未考慮從 SQL 查詢介面使用這些演算法的議

題，以使 SQL 查詢介面可用於支援資料整合的支援及較大查詢工作流程的處理。如先前所提，也很困難的是在商用關係資料庫中將這些平行程式直接建置作為內建使用者定義程式，因為這些平行程式大量使用一般在資料庫程式化及執行時間環境中不受支援的訊息傳遞及其他平行程式化建構。

BLAST 演算法具有低的計算複雜性(在要匹配的兩個輸入序列字串大小上大致為線性)，但在生物資訊學中存在輸入大小上具有二階或更高複雜性的其他搜尋及匹配演算法，諸如 Needleman-Wunsch 演算法、Smith-Waterman 演算法、最大相似度匹配、及系統性匹配(即，其複雜性至少大約為兩個輸入序列字串大小的乘積；見 W. R. Pearson 的「蛋白質序列比較及蛋白質演化(Protein Sequence comparison and Protein evolution)」，分子生物學的智慧型系統(Intelligent Systems in Molecular Biology)，2001 年)。這些演算法的計算需求比 BLAST 演算法大上許多，致使這些演算法的內建使用者定義程式在同時也在處理其他工作負載的資料庫伺服器上，受到極大的效能限制。然而，對於這些演算法而言，從資料庫伺服器至外部 HPC 平台的資料傳輸耗用在與等效 BLAST 建置比較時，將是整體執行時間的較小部分。因此，對於此情況，本發明尤其適合，尤

其是因為在後端 HPC 平台上使用最佳化(諸如記憶體中資料結構及細密平行性)，大幅縮減了執行時間。

本發明所針對的另一生命科學應用程式集屬於系統生物學的領域，其著重於各種生物學網路(諸如新陳代謝路徑、反應網路、基因調節網路、及蛋白質-藥物交互作用)之關係的研究。這些關係中有許多被儲存為圖形結構，及可經由這些圖形提出與生物學有關的查詢，其在資料庫中可儲存為複雜的使用者定義資料類型，或者即時實質化為較簡單資料類型集合(包含節點、邊緣及屬性資訊)上的彙總。這些圖形資料類型的系統生物學圖形資料庫擴充項連同對這些資料類型相似度、搜尋及推論的圖形操作集合說明於 B. A. Eckman 及 P. G. Brown 的「分子及細胞生物學的圖形資料管理(Graph data management for molecular and cell biology)」(IBM 研究及開發期刊(IBM Journal of Research and Development)，第 50 卷，第 545-560 頁，2006 年)，其中像圖形同型性、子圖形匹配、連接組件最短路徑、生成樹等標準圖形操作中有許多在此資料庫擴充項中均已建置為使用者定義函數。對於極大型的圖形及昂貴計算的圖形演算法，可使用本發明將這些使用者定義函數的一些卸載至附屬 HPC 平台來

提高查詢效能。

在使用內建使用者定義程式儲存、查詢及分析複雜資料類型時，對商用關係資料庫的使用日益增加，並且已察覺到在現有的商用資料庫平台上執行昂貴計算的使用者定義程式時，有許多阻礙效能的項目。

因此，本發明有關在不同且獨立的平行高效能計算系統上，自多個資料庫查詢中執行一或多個計算密集部分的新穎系統及方法。從資料庫伺服器至附屬 HPC 平台對此工作負載實行整體調度及遠端執行，使得從核發資料庫查詢的應用程式終端使用者的觀點來看，就好像是由資料庫伺服器本身上的等效使用者定義程式實行此工作負載，但因為遠端執行而具有更好的平行效能。因此，總而言之，本發明揭示一種電腦系統，其包含：

一商用資料庫平台；

一附屬高效能計算(HPC)平台；及

具有用於以下項目之組件集合的一系統：將查詢工作負載的計算密集區段及對應的目標資料表格從資料庫平台調度至附屬 HPC 平台；在此 HPC 平台上執行此工作負載；及將結果傳回至資料庫系統，由此將這些結果併入原始使用者查詢的最終結

果集中。

資料庫及 HPC 平台為具有其慣用系統及軟體堆疊的標準「現成」產品，由本發明的架構整合用於查詢調度、遠端執行及結果集合。

此架構亦提供「捷徑」功能，以直接經由網頁服務介面，在 HPC 平台上調用查詢執行的計算密集區段。已察覺到在測試及開發期間以及在一些網頁服務式應用程式中，需要直接使用此網頁服務函數來對照資料庫內在資料啟始基於 HPC 的應用程式，不必明確經歷資料庫查詢介面。然而，此網頁服務調用介面無法在資料庫伺服器上提供複雜查詢處理所使用之 SQL 查詢介面的好處，在複雜查詢處理中，遠端執行在某資料庫查詢工作流程中可作為中間步驟。

參考附圖，即可清楚瞭解本發明關於其結構及操作二者的細節，圖中相似的參考數字代表相似部分。

【實施方式】

本發明一般有關從資料庫伺服器將計算密集使用者定義操作卸載至附屬高效能平行電腦。以下說

明係使熟此技藝者能夠實現及使用本發明，並在專利申請案及其需求的內容中提供以下說明。熟習本技術者很容易即可明白對於本文所述較佳具體實施例及一般原理與特色的各種修改。因此，本發明並無意受限於所示具體實施例，而是旨在符合與本文所述原理及特色一致的最廣範疇。

為詳細說明本發明特色，現將連同附圖參考以下論述。

圖 1(數字 10-16)是本發明的高階圖式，其中用戶端應用程式在步驟 1 中核發包含一或多個計算密集操作的 SQL 查詢。查詢工作負載之計算密集部分的一些或全部(通常可能在資料庫伺服器上被建置為內建使用者定義程式)在步驟 2 中在附屬高效能平行電腦上卸載及執行。將來自這些所卸載計算之每一者的結果集傳送回到資料庫，以在步驟 3 中進行任何進一步處理，包括可能為整合至結果集(最終在步驟 4 中傳回至用戶端應用程式)所需的任何處理。在圖 1 中圖解了本發明兩個不可或缺的方面。第一，將計算密集工作負載卸載至附屬平行電腦，可針對相同目標資料庫表格的單一查詢調用或多個相關查詢調用，提高資料庫伺服器上的查詢效能及查詢回應時間。第二，藉以獲得此效能提高的整個程

序不需要對用戶端應用程式做出顯著修訂，因為在後端平行電腦上執行使用者定義程式，如同此使用者定義程式在資料庫伺服器本身上執行般，採用相同的語義及可靠性來進行。然而，本發明提供用戶端應用程式在資料庫伺服器上使用 SQL 介面，自訂及最佳化此所卸載之遠端執行之特定方面的能力。

圖 2(數字 18-22)圖解可為本發明一般或特定具體實施例之部分的各種組件。可使用這些組件初始化附屬平行電腦上的服務，使附屬平行電腦可準備執行未來所卸載的計算、在受到請求時排程這些計算、及收集結果並將結果傳送回到資料庫伺服器。這些個別組件通常部署在平行電腦本身上，或部署在高效能平行電腦系統的一或多個前端主機電腦上。本發明的另一組件集合係部署在資料庫伺服器本身上，及由使用者定義程式存根構成，使用者定義程式存根視情況使用諸如網頁服務或 JDBC(Java 資料庫連接)等標準協定在後端平行電腦上調用對應服務。此外，資料庫伺服器提供各種臨時表格，以在給定查詢工作流程中儲存中間或最終結果時使用。在 HPC 平行平台本身上，本發明的主要組件是在每一平行計算節點運行的服務包裝，每一平行計算節點可壓縮在該節點上的實際服務以執行平行工作。此服務包裝負責與前端主機上的其他組件通

信，以實行整體排程及同步化。服務包裝可由基礎節點服務使用簡單程式化介面在每一子分割區內擷取所需表格列(或每一此種資料庫表格列中的欄位值子集)以高效率存取的形式，儲存相應目標資料庫表格或實質化視圖的不同子分割區。

如文中所述，前端主機電腦含有本發明許多的重要組件，其包括：

一服務部署模組，負責在平行機器的所需節點子集上載入應用服務。

一服務節點調度器組件，其維持現用分割區上的狀態，或在已部署應用服務的平行電腦節點集。

一查詢調度器組件，其與服務節點調度器為特定的服務調用聯合請求平行機器上的節點子集，且能夠在目標資料庫表格或實質化視圖沒有任何變化時，在相同查詢分割區上重新調度未來查詢(藉此避免將目標表格資料從資料庫再次複製至平行電腦的耗用)。

一結果收集器組件，其彙總平行機器上個別計算節點的結果，及可將這些結果傳回至資料庫伺服器上調用中的服務函數，或將其插入資料庫伺服器

上預先指定的臨時表格。

一資料庫中繼組件，其係可在本發明特定具體實施例使用的項目，因為許多平行HPC平台並不支援任何協定或程式化 API 以進行互動式資料庫存取。在這些情況中，此資料庫中繼組件管理在資料庫伺服器及平行電腦節點之間的資料輸送，在資料往來於資料庫伺服器傳輸時所使用的 I/O 協定及資料往來於平行電腦節點傳輸時所使用的協定之間進行調解。

圖 3 至 5(數字 46-60)顯示本發明所需的步驟序列，其中每一個圖式對應於所卸載平行查詢執行的連續階段。此處階段 I 指的是應用程式的部署，階段 II 指的是資料初始化，及階段 III 指的是 HPC 平台上所卸載工作的執行及對資料庫伺服器的結果傳回。

圖 3(數字 24-32)說明部署階段或階段 I，其中在步驟 1，熟習應用服務特定技術者提供所要應用服務的特定軟體建置，其係內建在壓縮一般服務為應用服務的服務包裝內，其進一步說明如下。在步驟 2，使用平行電腦上個別計算節點集合執行的相應平行程式庫及執行時間，將此應用服務(連同服務

包裝)編譯成在平行電腦上用於個別節點程式的二進制。在步驟 3，在確定服務節點調度器組件自一些先前例項未被初始化及運行後，在平行電腦主機上啟動服務節點調度器組件。注意，步驟 1-3 係由擁有特定應用程式背景及在平行電腦上執行此應用程式之特定技能的熟習本技術者實行。在步驟 4，自資料庫伺服器接收特定請求作為其應用程式工作流程執行的部分，其中平行電腦主機上的程式載入器在平行電腦上給定的計算節點集合上啟動應用服務(這通常是用於平行二進制的特定平台載入器，諸如基於 MPI 之應用程式的 MPIRUN，請見 http://www-unix.mcs.anl.gov/mpi)。隨著在這些計算節點上載入應用服務，可將控制傳輸至服務包裝，其啟始訊息以用主機電腦上的服務節點調度器登錄節點。服務節點調度器維持目錄及計算節點的相關計數，計算節點的相關計數可用依此方式部署的每一特定應用服務取得。

圖 4(數字 34-44)說明應用服務的資料初始化階段或階段 II，其中在步驟 1，從查詢分割區調度器組件的資料庫伺服器接收請求，以載入目標表格在後續階段 3 中運行未來查詢請求時加以對照。可結合來自資料庫伺服器本身或某其他外部資料來源(諸如 ftp 伺服器)的各種來源資料庫表格取得此目標

表格，接著將此目標表格載入在其上初始化及運行應用服務的節點分割區子集。在步驟 2，查詢分割區調度器檢查在此表格載入的情況下是否存在現有的分割區已在處理新查詢或查詢集的就緒狀態。若找到此分割區但此分割區卻因另一查詢在其上運行而無法使用，則可使用已載入該分割區的目標表格資料複製另一現用查詢分割區(致使所有資料傳輸在 HPC 平台系統本身內以高速進行，而非回復到此資料的原始資料庫，如此會有較高的通信耗用)。否則，若未找到此分割區，則查詢分割區調度器與服務節點調度器進行協調，以配置另一閒置的應用服務節點子集並建立新的現用查詢分割區，如步驟 3 所示。在步驟 4，此現用查詢分割區之個別應用服務節點上的服務包裝接著啟始另一資料傳輸請求，以使用步驟 5 的資料庫中繼組件，從資料庫伺服器複製所需資料之彼此獨立但互無遺漏的列分割區，由此如步驟 6 將資料分割區儲存在區域資料快取中(較佳是在應用服務之服務包裝的記憶體內資料快取中)。特別是，可使用在平行電腦或平行電腦主機上運行的資料庫中繼組件，在步驟 5 及 6 調解在平行電腦節點及中繼主機之間的資料傳輸及通信協定(可根據 MPI 訊息傳遞、或通信協定的 UNIX 通訊端類型)，及在步驟 7 調解在中繼主機及資料庫伺服器之間的資料傳輸及通信協定(可根據資料庫存取的

JDBC 協定)。因此，計算節點上的應用服務包裝運送所需 SQL 查詢函數至資料庫中繼組件，其再完成查詢及將結果集以相應表示法傳送回到計算節點，以便儲存在應用服務包裝中所維持的資料快取中。其後，在下文說明的後續查詢執行階段期間，只需要從此區域資料快取讀取此資料，及應用服務為此目的使用特定 AP 存取此資料。

圖 5(數字 46-60)說明查詢執行階段或階段 III，其中在步驟 1，由在資料庫伺服器上執行的使用者定義函數啟始查詢請求。此查詢請求壓縮為平行電腦節點上運行之應用服務所需的所有輸入參數值，包括用於執行查詢的特定目標表格。此查詢請求的端點是在平行電腦之前端主機上運行的應用服務主機組件。在步驟 2，此應用服務主機進而將此查詢請求插入在查詢分割區調度器中維持的佇列集合(單獨佇列係維持用於已在上文階段2配置及指派至特定目標表格的每一分割區)。在步驟 3，查詢分割區調度器最後將此查詢請求提交至合適的分割區，及在步驟 4 等待任務完成，其中可選擇產生任務狀態碼及將其傳回核發應用程式執行請求的資料庫伺服器中的使用者定義函數。將查詢字串本身複製至在現用查詢分割區中每一節點之應用服務的服務包裝配置的記憶體緩衝區中。在步驟 5，在平行電腦

主機之結果收集器組件內，彙總儲存在此分割區每一節點之應用服務之服務包裝的結果快取中的查詢結果，及其後將所彙總的結果資料傳回資料庫伺服器上的原始使用者定義函數，如步驟 6 所示。由於調用遠端執行的原始使用者定義函數係為表格使用者定義函數或內建在表格使用者定義函數中，因此可將這些結果進一步處理作為複雜SQL查詢工作流程的部分，即根據結果的行值實行排序依據(ORDER BY)或群組依據(GROUP BY)操作，或如查詢執行的整體工作流程所需，將此結果表格加入其他資料表格。或者，每一節點上的應用服務包裝亦可使用如在階段 II 的資料庫通信中繼，將結果直接往回插入資料庫上特定結果表格，如步驟 7 及 8 所示(如果允許用於遠端執行的原始使用者定義表格函數存取此結果表格以產生其自己的傳回表格值，此方法為預設，但在一些資料庫平台上，通常阻止此使用者定義表格函數存取資料庫表格)。

本發明特定具體實施例

對於特定具體實施例，將描述生物資訊學序列匹配應用程式，其提供可建置為資料庫內使用者定義程式之計算密集演算法的良好範例，且其因此可從本發明獲得好處。特別是考慮到 FASTA 封裝的 SSEARCH 程式(詳見 http://helix.nih.gov/docs/gcg/

ssearch.html)，其提供 Smith-Waterman 演算法(T. F. Smith 及 M. S. Waterman 的「生物序列的比較(Comparison of Bio-sequences)」，應用數學進展(Advances in Applied Mathematics)，第 2 卷，第 482-489 頁(1981 年))，如 W.R. Pearson 的「搜尋蛋白質序列資料庫方法的比較(Comparison of methods for searching protein sequence databases)」(蛋白質科學(Protein Science)，第 4 卷，第 1145-1160 頁，1995 年)所實施。與其他序列匹配演算法(諸如先前所論述的 BLAST)相比，此為計算更密集的演算法但可產生高靈敏性相似度匹配結果，且可用來在比較測試中對甚至相關極微的序列對恢復匹配。

在特定具體實施例中使用的商用資料庫伺服器平台係為 IBM DB2 版本 9.1(http://www.ibm.com/software/data/db2)，其運行於雙處理器 Xeon 2.4 GHz CPU 上，及 RAM 儲存為 2GB 且乙太網路介面為 1000 Mbit。

查詢工作負載之計算密集部分之遠端執行所使用的 HPC 平台由 IBM 藍色基因/L e-server 平台的單一機櫃 (http://www.research.ibm.com/bluegene) 組成，其由 1024 個計算節點組成，其中每一計算節點包含兩個以 700 MHz 操作的 PowerPC 440 處理器且

每個節點 RAM 儲存為 512 MB。雖然 IBM 藍色基因/L 的程式通常使用 MPI 訊息傳遞程式庫寫入，但此處對藍色基因/L 平台的特定使用並不需要任何通信程式庫，不過本發明亦不排除對此類程式庫的使用。熟習本技術者可使一般方法適合用於其他平行電腦，及例如，可使用已在其他平台上建置及成為基準之 SSEARCH 演算法的多緒或共用記憶體版本，獲得本發明其他特定具體實施例(如，Y. Chen、J. Mak、C. Skawratananond 及 T-H. K. Tzeng 的「在 AIX 及 IBM e-server pSeries 690 之 Linux 上之應用程式的生物資訊學可擴充性比較(Scalability Comparison of Bioinformatics for Applications on AIX and Linux on IBM e-server pSeries 690)」，http://www.redbooks.ibm.com/abstracts/redp3803.html，IBM 紅皮書(IBM Redbook)，2004 年)。

為實現本發明而組態資料庫平台及 HPC 平台的其他組合時，可能存在特定的技術性議題，但其可由熟習本技術者解決。對於當此組合為 IBM DB2 資料庫平台及 IBM 藍色基因/L 平行電腦平台時的情況(以及對於已知商用資料庫及平行電腦平台的其他等效組合)，尚未有任何 API 或程式化支援可用於在資料庫伺服器及平行電腦上個別計算節點之間的通信。對於 IBM 藍色基因/L 而言，個別計算節點為

無磁碟系統，其僅提供在完整獨立的作業系統可供使用的服務子集。因此，運行完整 Linux 作業系統且經由區域網路上連接至藍色基因/L 系統的單獨 IBM P-series 伺服器可用來代管本發明中的各種組件，其包括：

1)排程器組件，其含有可供查詢處理應用程式使用之藍色基因/L 計算節點分割區的登錄；

2)網頁伺服器組件，其支援自資料庫伺服器啟始之基於 SOAP 的網頁服務呼叫，以執行查詢工作流程的不同組件；

3)任務提交介面組件，其可保留及啟動藍色基因/L 電腦之計算節點上的應用程式；

4)資料庫中繼組件，其維持一或多個對個別藍色基因/L 計算節點的通訊端連接，及負責在這些通訊端連接上執行自計算節點中繼的各種資料庫命令，及將結果集或這些資料庫命令的完成碼傳回啟始資料庫查詢請求的計算節點。

使用簡單的結構描述將相關的 FASTA 序列資料庫匯入 DB2 關係資料庫，藉此對於每一序列資料庫(如「drosophila」)，將資料儲存在分別由各行[id(整數)、名稱(字元字串)、描述(字元字串)、序列(clob)]組成的表格中。此處，id 欄位是連續的記錄數字，而名稱(為 NCBI 識別符)及描述欄位則使用剖析

FASTA 資料庫及將其插入對應 DB2 表格的指令碼，以 FASTA 格式取自每一序列的標頭。可將含 14331 列的「Drosophila」資料庫(包含「Drosophila」核甘酸資料庫的蛋白質編碼序列轉譯)用於本文說明的結果。

Smith-Waterman 演算法建置是 FASTA 封裝中的 SSEARCH 程式(見 http://helix.nih.gov/docs/gcg/ssearch.html)，及改變此舊有碼以將此建置為平行電腦上的應用服務，其改變微乎其微。修改主要的進入點即可用必要的埠及位址啟動節點服務包裝，使其可以連接至應用程式排程器(如圖 3 的步驟 3 所示)。其後，此節點應用服務包裝(其含有暫存記憶體，用於保持目標程式庫序列的相應分割子集及對應的查詢結果集，且含有用於輸入查詢序列之記憶體緩衝區)負責以應用程式排程器登錄應用服務用戶端，實行目標程式庫序列的緩衝區更新，及對於每一個新的查詢請求，對照此目標程式庫來運行 SSEARCH 演算法。FASTA 封裝已經支援廣泛的檔案及資料庫格式清單用於讀取目標程式庫序列，致使很容易將從節點服務包裝的暫存記憶體存取資料的存取函數新增為另一個輸入來源。

因此，總之，除了 FASTA 封裝應用程式，為實

行此特定具體實施例所需的一般組件包括：計算節點服務包裝函數、用於節點服務包裝暫存記憶體的存取函數、及資料庫中繼組件，所有這些組件可供與 FASTA 封裝之 SSEARCH 相似之應用服務的廣泛類別再次使用。

根據此特定具體實施例的 SQL 查詢範例如圖 6 所示。此查詢以參數清單啟始 DB2 使用者定義表格函數 ssearch_call，此參數清單包括：目標序列程式庫、要匹配之輸入序列的描述符字串、輸入序列本身、及所要最高排序匹配的數字。如圖所示，在從 SQL 查詢介面調用時，在藍色基因/L 計算節點上實行匹配及排序，然後如圖 2 所示傳回結果。加速及效能測量將在本專利的最後版本中提供。

雖已按照所示具體實施例來說明本發明，但一般技術者應明白，各具體實施例可以有所變化且這些變化均在本發明的精神及範疇內。因此，一般技術者可在不悖離隨附申請專利範圍的精神及範疇下進行許多修改。

【圖式簡單說明】

圖 1 概要說明所提出之發明，其中用戶端應用程式核發一或多個 SQL 查詢，在附屬高效能平行電

腦(HPC)上調度及執行查詢工作負載的計算密集部分，並將結果傳回至資料庫，用於最後併入傳回至用戶端應用程式的最終結果集；

圖 2 圖解本發明可用以產生本發明特定具體實施例的各種組件，將這些組件個別地部署在資料庫伺服器、平行電腦的主機節點、及平行電腦本身上；

圖 3 圖解查詢執行之階段 I 所涉及的步驟，其中將負責執行所需卸載之資料庫查詢的應用服務安裝在平行電腦中的計算節點集合(稱為應用服務節點)；

圖 4 圖解查詢執行之階段 II 所涉及的步驟，其中將查詢中所使用的目標資料程式庫從階段 1 複製到應用服務節點的子集(下文將此節點子集稱為現用查詢分割區)；

圖 5 圖解查詢執行之階段 III 所涉及的步驟，其中將相關查詢參數從階段 2 傳遞至相應的現用查詢分割區，及收集結果並將結果傳回至調用的資料庫函數或資料庫伺服器中的結果表格；

圖 6 圖解下文所考慮之本發明特定具體實施例

中的查詢請求範例，該查詢請求經核發至資料庫伺服器，以對照資料庫伺服器中所儲存且複製至現用查詢分割區的特定目標序列程式庫，匹配所給定的序列(在遠端平行電腦上使用 Smith-Waterman 演算法的平行化)，以及在遠端執行後傳回的結果。

【主要元件符號說明】

12　用戶端應用程式
14　關係資料庫伺服器
16　高效能平行電腦
20　資料庫伺服器
22　平行電腦節點
26　在服務包裝內安裝應用程式
28　應用程式服務
30　平行節點上的應用程式服務
32　啟動服務節點調度器
36　查詢分割區調度器
38　服務節點
40　服務中的資料快取載入器
42、58　資料庫通信中繼
44、60　資料庫
48　應用程式服務主機
50　查詢分割區調度器
52　平行服務安裝

54　結果快取

56　結果收集器

五、中文發明摘要：

本發明有關於在附屬高效能平行計算平台上調度及執行資料庫查詢工作流程之計算密集部分的系統及方法。在資料庫平台及執行工作負載的高效能計算平台之間移動所需資料及結果的效能耗用可以下列若干方式分攤：利用平行計算平台上的細密平行性及優異硬體效能，以加速計算密集運算；使用平行計算平台之記憶體中資料結構來快取相同資料之時間滯後查詢序列間的資料集，以在沒有進一步資料傳輸耗用的情況下處理這些查詢；複製平行計算平台中的資料，以使用高效能計算平台的獨立平行分割區同時處理相同目標資料集的多個獨立查詢。

本發明之特定具體實施例係用於在經由乙太網路區域網路連接至平行超級電腦的資料庫系統上，部署涉及使用 Smith-Waterman 演算法之基因及蛋白質序列匹配的生物資訊學應用。

六、英文發明摘要：

The invention pertains to a system and method for dispatching and executing the compute-intensive parts of the workflow for database queries on an attached high-performance, parallel computing

platform. The performance overhead for moving the required data and results between the database platform and the high-performance computing platform where the workload is executed is amortized in several ways, for example, by exploiting the fine-grained parallelism and superior hardware performance on the parallel computing platform for speeding up compute-intensive calculations, by using in-memory data structures on the parallel computing platform to cache data sets between a sequence of time-lagged queries on the same data, so that these queries can be processed without further data transfer overheads, by replicating data within the parallel computing platform so that multiple independent queries on the same target data set can be simultaneously processed using independent parallel partitions of the high-performance computing platform.

A specific embodiment of this invention was used for deploying a bio-informatics application involving gene and protein sequence matching using the Smith-Waterman algorithm on a database system connected via an Ethernet local area network to a

parallel supercomputer.

七、指定代表圖：

(一)本案指定代表圖為：圖 1。

(二)本代表圖之元件符號簡單說明：

10 電腦系統

12 用戶端應用程式

14 關係資料庫伺服器

16 高效能平行電腦

八、 本案若有化學式時，請揭示最能顯示發明特徵的化學式：無。

十、申請專利範圍：

1. 一種電腦系統，包含：

(i)一高效能平行計算(HPC)平台；

(ii)一通用資料庫平台；

(iii)用於調度計算密集使用者定義工作負載及對應資料視圖之一裝置，該資料視圖形成該資料庫平台上一或多個查詢之工作流程的部分，以在該 HPC 平台上執行；

(iv)用於將在該 HPC 平台執行之該工作負載的該結果傳回該資料庫平台的一裝置，以將這些結果併入該資料庫查詢的該最終結果集中。

2. 如請求項 1 之所述系統，其中該高效能平行計算平台就處理單元及快速記憶體而言，能夠部署更多計算資源，致使當工作負載包含一單一查詢、或一序列相似或相關查詢、或一同時獨立查詢集合時，該所調度的計算密集工作負載可以一高效率方式執行。

3. 如請求項 2 之所述系統，其中該高效能平行計算平台支援用於共用記憶體或分散式記憶體訊息傳遞應用程式的平行執行，藉此支援一般不為資料庫內建使用者定義程式直接支援的程式化模型。

4. 如請求項2之所述系統，其中該高效能平行計算平台可利用其大規模的記憶體可用性來快取在記憶體中結構內的表格資料，以進行更快速的計算處理，並可使由該相同資料表格上一序列查詢所產生的一序列計算密集工作負載，以高效率的方式執行，且不需要為了每一個新的查詢調用而在該資料庫伺服器及該高效能平行電腦系統之間進一步交換資料。

5. 如請求項2之所述系統，其中該高效能平行計算平台能夠對該資料庫資料建立快速複製，致使源自該資料庫伺服器上同時、獨立、平行查詢之一對應集合的同時、獨立、計算密集工作負載集合可以高效率的方式執行，且不需要因為回復該資料庫伺服器以取得該相同資料表格之額外複製所需之大量通信耗用。

十一、圖式：

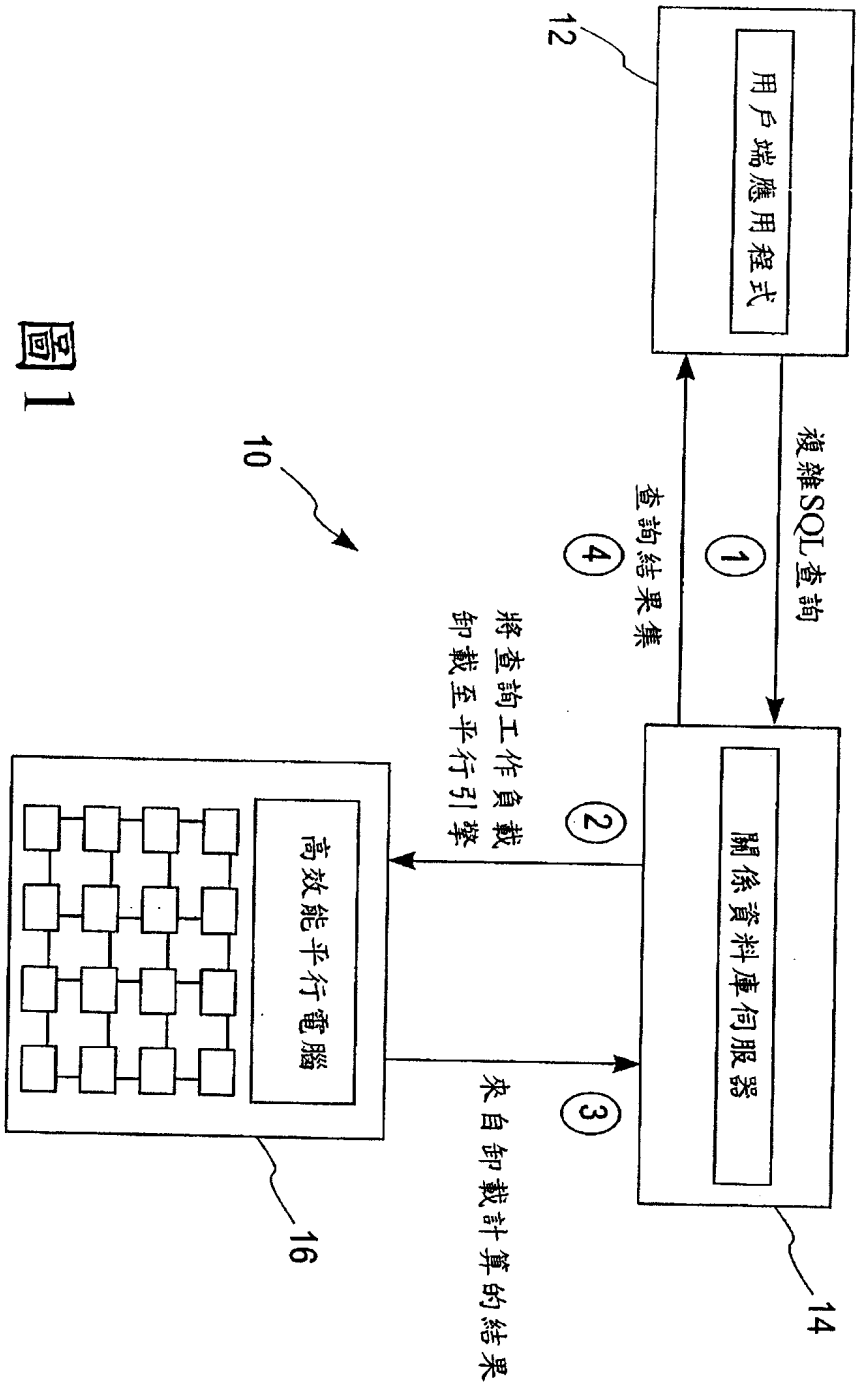
用戶端應用程式
12
複雜SQL查詢
①
查詢結果集
④
關係資料庫伺服器
14
將查詢工作負載
卸載至平行引擎
②
來自卸載計算的結果
③
10
高效能平行電腦
16

圖1

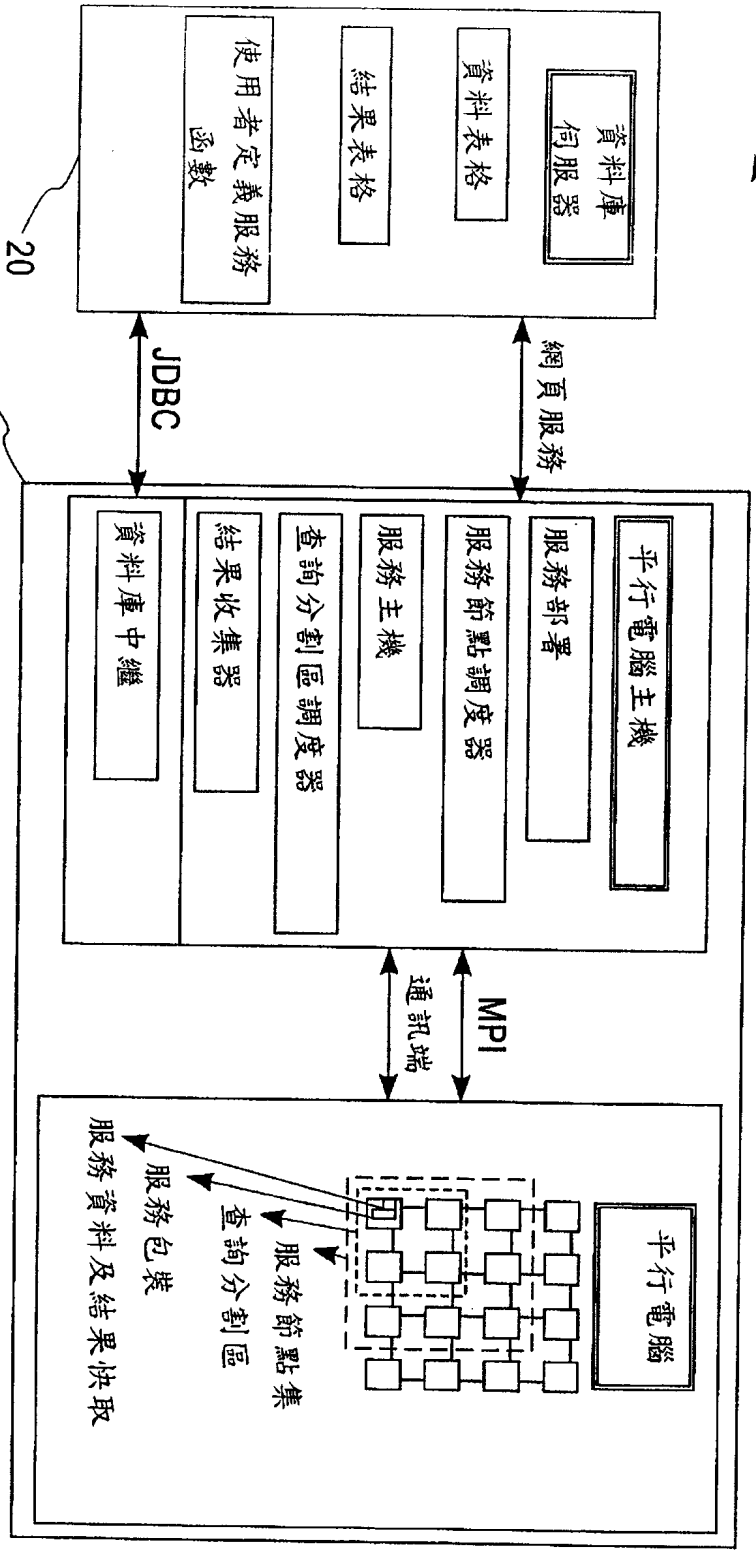
18
資料庫伺服器
資料表格
結果表格
使用者定義服務函數
20
JDBC
網頁服務
22
平行電腦主機
服務部署
服務節點調度器
服務主機
查詢分割區調度器
結果收集器
資料庫中繼
通訊端
MPI
平行電腦
服務節點集
查詢分割區
服務包裝
服務資料及結果快取

圖2

200928813

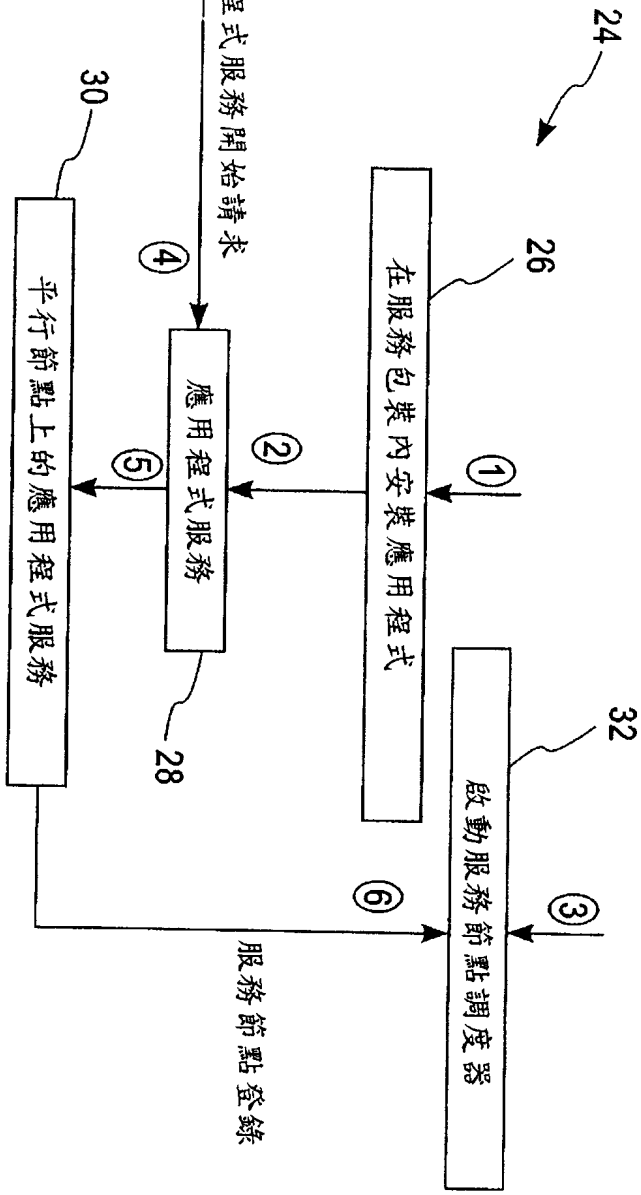

圖3

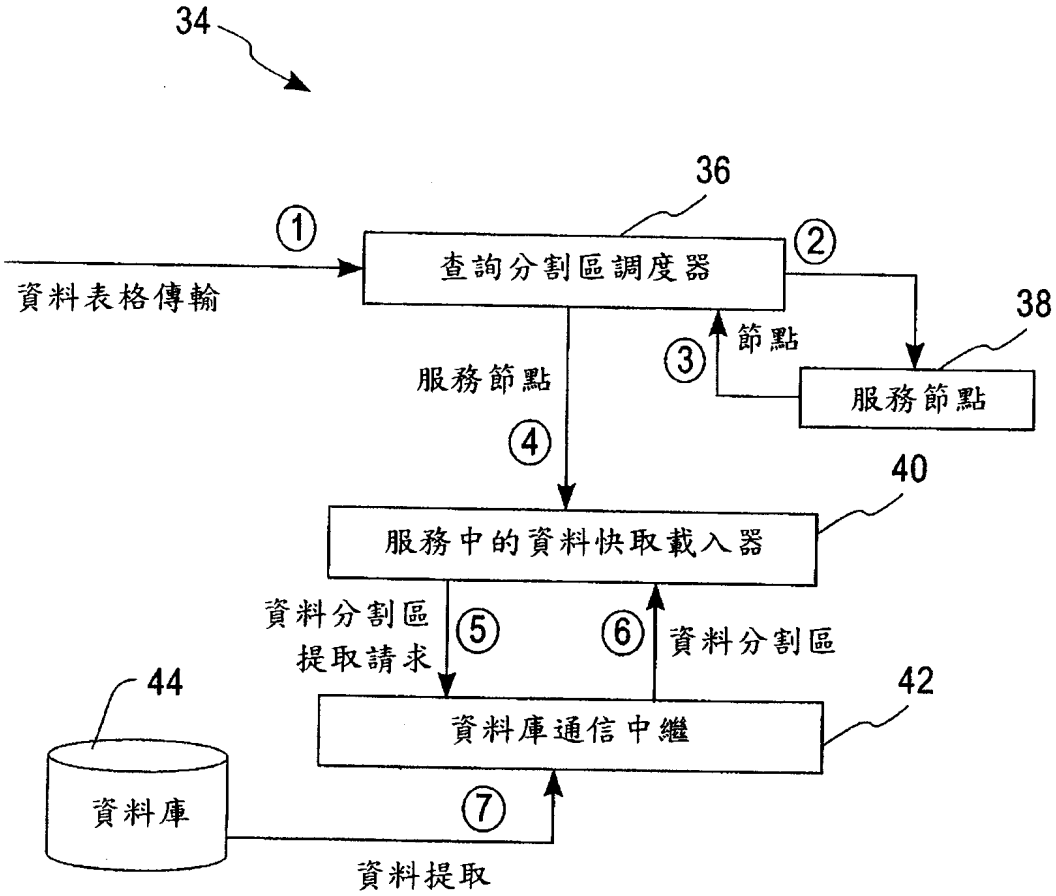
34
36
①
查詢分割區調度器
②
資料表格傳輸
38
節點
③
服務節點
服務節點
④
40
服務中的資料快取載入器
資料分割區
提取請求
⑤
⑥
資料分割區
44
42
資料庫通信中繼
資料庫
⑦
資料提取

圖4

200928813

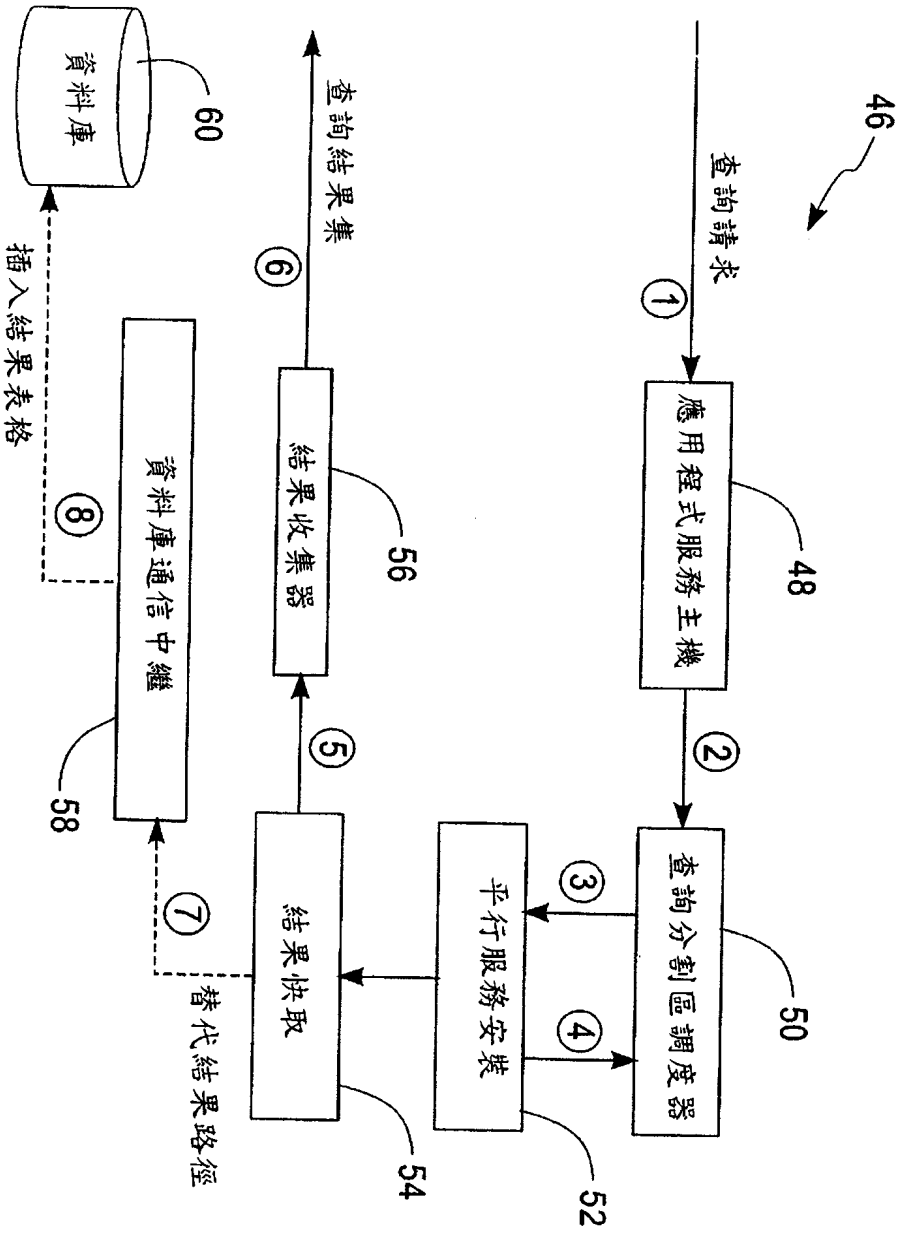

圖5

62

```
select * from table(ssearch_call('drosoph','query sequence',
'MPMILGYWNVRGLTHPIRMLLEYTDSSYDEKRYTMGDAPDFDRSQELNEKFKLGDFP
NLPYLI', 6)) as A
```

對於查詢之search_call之引數的描述:

'drosoph" 是執行匹配查詢所對照的程式庫/分割區名稱。
'query sequence' 是查詢序列的名稱/描述，為可取代的。
'MP ....' 為其所要匹配的查詢序列。
'6' 為要求顯示查詢結果集中前10個命中的請求。

此查詢傳回以下結果表格，其中ID是序列的資料庫內部整數id。

ID	SW	E	Z	BIT
1280	0	+8.22683627922823E-001	+1.06285301282655E+002	+2.72131589312291E+001
1070	0	+1.08072106272098E+000	+1.04158194576657E+002	+2.68195743620312E+001
1191	0	+1.41969279048097E+000	+1.02031087870658E+002	+2.64259897928334E+001
296	0	+3.21837510633985E+000	+9.56497677526638E+001	+2.52452360852397E+001
927	0	+5.55390162202516E+000	+9.13955543406675E+001	+2.44580669468439E+001
127	0	+5.55390162202516E+000	+9.13955543406675E+001	+2.44580669468439E+001

圖6

200928813

parallel supercomputer.

七、指定代表圖：

(一)本案指定代表圖為：圖 1。

(二)本代表圖之元件符號簡單說明：

10 電腦系統

12 用戶端應用程式

14 關係資料庫伺服器

16 高效能平行電腦

八、 本案若有化學式時，請揭示最能顯示發明特徵的化學式：無。