

發明專利說明書

中文說明書替換頁(104年4月)

(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：096145702

※ 申請日期：96. 11. 30

※IPC 分類：G06F 11/16 (2006.01)

一、發明名稱：(中文/英文)

資料管理方法

METHOD OF DATA MANAGEMENT

二、申請人：(共 1 人)

姓名或名稱：(中文/英文)

日商日本電氣股份有限公司
NEC CORPORATION

代表人：(中文/英文)

西本 裕
NISHIMOTO, HIROSHI

住居所或營業所地址：(中文/英文)

日本國東京都港區芝5丁目7-1
7-1; SHIBA 5-CHOME, MINATO-KU, TOKYO, JAPAN

國 籍：(中文/英文)

日本 JAPAN

三、發明人：(共 3 人)

姓 名：(中文/英文)

1. 塞查利 杜布尼克基
DUBNICKI, CEZARY
2. 艾利克 克魯斯
KRUUS, ERIK
3. 克利斯汀 安古利奧
UNGUREANU, CRISTIAN

國 籍：(中文/英文)

1. 波蘭 POLAND
2. 美國 U.S.A.
3. 羅馬尼亞 ROMANIA

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；2006年12月01日；11/566,139

2.

☐ 無主張專利法第二十七條第一項國際優先權：

1.

2.

☐ 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。

五、中文發明摘要：

本發明揭示用於資料管理及資料處理之系統及方法。例如，各種具體實施例可包括選擇相對更大資料群組而具有相當或更佳執行選擇結果(例如高資料冗餘消除及/或平均塊大小)相關之系統及方法。在各種具體實施例中，該(等)系統及方法可包括(例如)一資料群組、區塊或塊組合技術或/及一資料群組、區塊或塊分割技術。各種具體實施例可包括一第一標準或典型資料分組、編塊或組塊技術及/或資料群組、區塊或塊組合技術或/及一資料群組、區塊或塊分割技術。(多個)範例性系統及方法可能係關於資料雜湊及/或資料消除。多個具體實施例可包括一預看緩衝器並決定是否基於下面資料之特性及/或本發明之特定應用(例如用於備份)來發射小塊或大型塊。

六、英文發明摘要：

System(s) and method(s) are provided for data management and data processing. For example, various embodiments may include systems and methods relating to relatively larger groups of data being selected with comparable or better performing selection results (e.g., high data redundancy elimination and/or average chunk size). In various embodiments, the system(s) and method(s) may include, for example a data group, block, or chunk combining technique or/and a data group, block, or chunk splitting technique. Various embodiments may include a first standard or typical data grouping, blocking, or chunking technique and/or data group, block, or chunk combining technique or/and a data group, block, or chunk splitting technique. Exemplary system(s) and method(s) may relate to data hashing and/or data elimination. Embodiments may include a look-ahead buffer and determine whether to emit small chunks or large chunks based on characteristics of underlying data and/or particular application of the invention (e.g., for backup).

七、指定代表圖：

(一)本案指定代表圖為：第 (2) 圖。

(二)本代表圖之元件符號簡單說明：

(無元件符號說明)

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)

九、發明說明：

【發明所屬之技術領域】

本發明係關於資料處理及資料管理領域且更明確而言係關於用於諸如資料雜湊及/或資料冗餘消除之應用之有效率處理相關的方法及系統。

【先前技術】

全世界每天產生越來越多的資訊且正在保持並傳輸的資訊數量不斷以令人擔憂的速度複合，從而引起關於資料處理及管理之嚴重憂慮。大多數的此資訊經產生、處理、維持、傳輸並電子儲存。試圖管理所有此資料及相關資料流及儲存之純粹量級係令人吃驚的。由此，已研發出許多系統及方法，以更有效率地處理資料並藉由盡可能多地消除複製資料來儲存並傳輸更少資料。例如，已研發出各種系統及方法來幫助降低儲存、傳輸等來自各種電子裝置(例如電腦、電腦網路(例如LAN、企業內部網路、網際網路等)、行動裝置(例如電話)、PDA、磁碟機、記憶體晶片等)之複製資料之需要。此類技術可用於或包括資料壓縮、資料加密及/或資料儲存。此外，需要使用密碼學來加密資料，特別係在(例如)資料傳輸期間。例如，已研發出提供強(即密碼)雜湊之系統及方法，故此類方法可相當自然地併入使用資料雜湊在不安全通信通道上完成資料冗餘消除之應用內。已研發出用於也在安全系統上進行資料雜湊及/或資料冗餘消除之系統及方法。在歸檔流內的複製資料識別與資料冗餘消除係一用以節省儲存空間之技

術。在各種電子資料管理方法及系統中，已研發出若干方法學，用於資料雜湊及/或從(例如)資料儲存(例如為電子郵件或家庭目錄歸檔、備份資料)及資料傳輸中消除冗餘資料。該些技術包括各種資料壓縮(例如zip技術)、資料雜湊及密碼學方法學。

一些特定類型雜湊可能包括內容組塊，其可包括全體檔案雜湊、固定大小組塊(編塊)及內容定義組塊。在各種論文中揭示一些用於資料流管理及資料處理之範例性技術，包括C. Policroniades與I. Pratt，用於在儲存系統資料中偵測冗餘之替代方案，USENIX_04：USENIX年度技術研討會會議錄(2004)，第1至14頁；R. Jain，用於在電腦網路中位址查找之雜湊方案之比較，IEEE通信學報40，1570(1992)，第1至5頁；N. Jain、M. Dahlin及R. Tewari，TAPER：用於在重複同步中消除冗餘之層次方案，德州大學奧斯汀分校電腦科學系Tech. Rep.，技術報告TR-05-42，第1至14頁；A. Chowdhury、O. Frieder、D. Grossman及M. C. McCabe，用於快速複製文件偵測之集合統計學，ACM Trans. Inf. Syst. 20, (2002), ISSN 10468188，第171至191頁；F. Douglass及A. Iyengar，經由相似偵測之特定應用增量編碼，USENIX年度技術研討會會議錄(2003)，第123頁；P. Kulkarni、F. Douglass、J. LaVoie及J. Tracey，在大型檔案集合中的冗餘消除，USENIX年度技術研討會會議錄(2004)，第114頁；J. Barreto與P. Ferreira，用於資源受限行動裝置之複製檔案系統，IADIS應用計算國際研討

會會議錄，(2004)，第1至9頁；T. Denehy與W. Hsu，用於參考資料之複製管理，技術報告RJ 10305，IBM Research (2003)，第1至14頁；G. Forman，K. Eshghi及S. Chiocchetti，在大型文件庫中發現相似檔案，KDD '05：第十一屆ACM SIGKDD資料探勘知識發現國際研討會會議錄，美國紐約(NY)的ACM出版社，(2005)，第394至400頁；L. You、K. T. Pollack及D. D. E. Long，深度儲存：一種歸檔儲存系統架構，ICDE '05：21屆資料引擎國際研討會會議錄，IEEE電腦社團，美國華盛頓特區，(2005)，第1至12頁；K. Eshghi與H. K. Tang，用於分析並改良以內容為主組塊演算法之框架，技術報告HPL-2005-30R1，HP實驗室(2005)，第1至10頁；P. L'Ecuyer，不同大小線性同餘產生器表及較佳格子結構，數學計算，68，249 (1999)，ISSN 0025-5718，第249至260頁；A. Tridgell 與P. MacKerras，"技術報告TRCS-96-05 -同步演算法"，澳大利亞國立大學，電腦科學系，FEIT，ANU (1996)，第1至6頁；及L. You與C. Karamanolis，"有效率歸檔儲存技術評估"，21屆IEEE/NASA Goddard MSS (2004)，第1至6頁。還有許多美國專利及專利公告案揭示各種相關範例性技術，包括美國專利公告案第2006/0112264、2006/0047855及2005/0131939號及美國專利第6,658,423及6,810,398號。該些參考文獻指示各種關於更有效率資料處理及資料管理之範例性技術。

上述各種參考文獻提供一選項(例如gzip、增量編碼、固

定大小編塊、可變大小組塊)簡介、組塊與增量編碼(增量編碼可能係一用於特徵在於頻繁較小變化之事物(例如日誌檔案及電子郵件)之較佳技術)之比較及針對實際資料固定大小組塊與可變大小組塊之比較。

然而，該等已知技術均缺少特定的有用能力。一般高度通用性資料壓縮或雜湊技術傾向於在某些資料類型上比在其他資料類型上(例如較短資料區塊對較長資料區塊)，對於特定應用比其他應用(例如壓縮而非資料儲存或備份)效果更佳，並相對於要處理之資料大小，資料處理速度不同且縮放屬性不同。此外，各種硬體及應用軟體會影響一資料處理或資料管理技術可能起到的效果。例如，如下所述，存在一些資料壓縮或冗餘消除技術，其在較短資料區塊(例如32k大小的區塊)或可能中等大小的資料區塊上效果極佳，但在大型(例如Gb大小的區塊)資料區塊上效果卻不佳。

不幸的係，該等已知技術一般既不針對特定用途、應用或硬體充分地考量資料圖案，也不在一資料集或資料流內識別一較高實際資料冗餘度時在處理期間有效率地管理資料片段之大小。已知複製消除方案難以增加儲存或傳輸資料片段之平均大小而不嚴重影響所獲得複製消除程度、所需時間及/或方案可縮放性。

因此，需要一種資料處理及管理技術，其在與歸檔資料、備份資料及/或更有效率傳輸或儲存於大型區塊或塊內之資料一起使用時具有合理的效能並特別有效率，同時

獲得一較高程度的冗餘消除。用於複製消除之效能目的可包括速度、一較大平均塊大小與較大複製消除數量組合及/或對極大資料集的可縮放性。

【發明內容】

本發明一般而言係關於提供用於資料管理及資料處理之系統及方法。例如，各種具體實施例可包括選擇相對更大資料群組而具有相當或更佳執行選擇結果(例如較高資料冗餘度消除及/或平均塊大小)相關之系統及方法。在各種具體實施例中，該(等)系統及方法可包括(例如)一資料群組、區塊或塊組合技術或/或一資料群組、區塊或塊分割技術。各種具體實施例可包括一第一標準或典型資料分組、編塊或組塊技術及/或資料群組、區塊或塊組合技術或/及一資料群組、區塊或塊分割技術。多個具體實施例可包括一預看緩衝器並決定是否基於下面資料之特性及/或本發明之一特定應用(例如用於備份)來發射小塊或大型塊。此外，(多個)範例性系統及方法可能係關於一種用於一資料集或一資料串之資料雜湊及/或資料冗餘識別及消除技術。本發明可能係一電腦實施之發明，其包括用於有效率地改良資料處理而不顯著降低資料處理結果之品質的軟體及硬體。

在至少一具體實施例中，各種用以最佳化內容定義組塊技術、功能或方法之構件可用於歸檔或備份資料流或資料集。各種具體實施例可包括用以使用多重選擇準則(如上述相關專利申請案中所述)加速組塊程序之構件。各種具

體實施例還可包括一用以使用備份切割點、斷裂點或塊點及切割位準、斷裂位準或塊位準以在給定最小及最大允許輸出資料群組、區塊或塊長度情況下，最大化產生可重製切割點、區塊點或塊點之可能性之方法。各種具體實施例可基於在一些歸檔或備份資料類型中複製圖案的一直覺模型。各種具體實施例可包括合併(多個)資料群組、(多個)區塊或(多個)塊，以便重新分佈該(等)資料群組、區塊或塊以依據不同參數及/或長度來決定改良的切割點、區塊點或塊點。此類具體實施例可能能夠增加複製消除比(DER)並藉由明智的試探法而不會嚴重地減小平均塊大小。在各種具體實施例中，可提供(多個)即時系統及方法，其可能能夠增加平均塊大小而不會嚴重減小複製消除比(DER)。

該些系統及方法可能包括分割或打散資料群組並可能涉及使用較大平均塊大小來樂觀地組塊長序列的"新"資料，並對於充分靠近可能係複製資料者的"新"資料使用更小塊。依此方式，一些類型的備份資料可能更容易係可能對應於局部變化的括號插入或刪除區域。為此可能需要一極有限的預看，使用一預看緩衝器及/或每輸入單元一限定數目的查詢，以決定是否先前已發射一預期塊。該等查詢可針對一後端儲存單元及/或在實際或虛擬記憶體(例如一布魯姆(Bloom)過濾器)內所維持的一有限大小局部歷史來確切或直接進行。

本發明之各種具體實施例已展現增加平均資料群組、區

塊或塊大小及複製消除比之能力。此外，各種具體實施例已能夠使用比先前可獲得的大約2至4倍之資料群組、資料區塊或資料塊來獲得相等的複製消除比。該些更大資料群組、區塊或塊可提供一儲存資料之元資料成本降低並可在傳輸資料及/或讀取或寫入另一裝置，例如一備份器具(例如一磁碟或網路儲存系統)時引起輸出量增加。

基於下列揭示內容及其附圖之研究，習知此項技術者將會明白用於各種具體實施例之另外包括方面。

【實施方式】

本文提供用於改良資料管理及資料處理之(多個)系統及方法。例如，本發明之各種具體實施例可包括關於選擇相對更大資料群組而具有可接受、相當及/或更佳執行選擇及/或冗餘消除結果(例如較高資料冗餘消除)之系統及方法。在各種具體實施例中，該(等)系統及方法可包括(例如)一資料群組、區塊或塊組合技術或/或一資料群組、區塊或塊分割技術。本發明可應用於各種類型的資料處理、資料傳輸及/或資料儲存，並本發明之特別有用應用可包括(例如)數個類型的網路檔案系統、參考檔案集合之空間最佳化歸檔以及檔案同步、備份資料儲存等。下面各種具體實施例將針對與備份資料儲存一起使用本發明提供多個範例，作為一特定範例性應用。本發明可能係一電腦實施之發明，其包括用於更有效率地改良資料處理及資料維護之軟體及硬體，且不會顯著降低資料處理結果之效能。該電腦實施方案可包括各種電子裝置(例如PDA、蜂巢式電

話、GPS等)、電腦(例如PC)、儲存裝置或系統及/或電腦網路(LAN、企業內部網路、網際網路等)，下面將更詳細地予以說明。

一般而言，壓縮演算法較適合於較短資料區塊之特性。例如，當視為位元組或英文字時，一上下文無關文法一般將會發現越來越少的涉及較長長度序列之產生。合理壓縮可藉由涉及一局部資料模型或以字典為主查找表技術之機制來獲得。例如，gzip在較短(例如32k)區塊上操作，使用一版本(距離、長度)指標來代替重複字串。在此情況下，gzip在一局部範圍內獲得冗餘消除，並使用長度可以相對較小的冗餘字串來完成此舉。

在歸檔資料中，情形可能相當不同。在大多數檔案存取係唯讀的實際檔案系統中，檔案傾向於主讀或主寫，且一小組檔案會產生大多數區塊覆寫。整個檔案可能係複製的，故甚至在改變時，該等變化可能侷限於一相對較小的編輯區域。此處，一壓縮方案必需有效地處理較長的重複資料片段。甚至更具約束性的係，資料重新出現的"未來"可能係將來十億位元組(即下一備份列)。在此情況下，可能針對在一大得多的輸入資料區域上在一全域範疇內的複製識別及冗餘消除。因為與儲存該些冗餘字串之"位置"相關聯的額外負擔，故關注塊要比(例如)資料壓縮技術(諸如gzip)所針對的該等塊大得多。

因為用於歸檔資料之基本假定係如此不同，故以字典為主的壓縮技術可能使用稱為複製消除(DE)的程序。在複製

消除中，只是可重製地打散一輸入資料流，並接著儲存(或傳輸)任何複製一先前發射塊之塊之唯一副本。

複製消除可能有別於進行相當多額外工作以改良可壓縮性的更大膽技術。在更大膽壓縮方案中的一常用技術係調用相似性偵測及某版本的增量編碼。不幸的係，在大型(十億位元組)集合中發現最長複製或發現相似(或相同)檔案係一極為棘手繁瑣的任務，其一般包括專用演算法(例如參見美國專利第6,658,423號)。

各種具體實施例可包括一資料分組、編塊或組塊技術及/或資料群組、區塊或塊組合技術或/及一資料群組、區塊或塊分割技術。在各種具體實施例中，該資料分組、編塊或組塊技術可利用一標準或典型資料分組、編塊或組塊方案。多個具體實施例可包括一預看緩衝器並決定是否基於下面資料之特性及/或本發明之一特定應用(例如用於備份)來發射小塊或大型塊。此外，(多個)範例性系統及方法可能係關於一種用於一資料集或一資料串之資料雜湊及/或資料冗餘識別及消除技術。在一些範例性具體實施例中，將在應用於資料歸檔及/或資料備份時說明本發明且其將經組態用以在該應用中表現特別出色。

在至少一具體實施例中，可提供各種用以最佳化內容定義組塊技術、功能或方法之構件用於歸檔或備份資料流或資料集。各種具體實施例可包括用以使用多重選擇準則(例如如上述相關專利申請案中所述)來加速組塊程序之構件。各種具體實施例還可包括一用以在給定最小及最大允

許輸出資料群組、區塊或塊長度下使用備份切割點、斷裂點或塊點及切割位準、斷裂位準或塊位準來最大化產生可重製切割點、區塊點或塊點之可能性的方法。此類切割、區塊或塊點可以一期望具有較佳複製偵測能力之方式來產生。各種具體實施例可基於在一些類型歸檔或備份資料中可能出現之複製圖案的一直覺、測量及/或動態模型。

在各種具體實施例中，可提供(多個)即時資料處理及/或資料管理系統及方法，其可能能夠增加平均塊大小而不會嚴重減小複製消除比(DER)。當考量將本發明應用於歸檔或備份資料時，一用於最新或新資料之假定可能係其在一未來備份列中具有一較高重新出現可能性。在該情況下，可證明使用較大資料群組、區塊或塊最具效率。因而本發明之該(等)系統及方法可能涉及使用(多個)較大平均塊來樂觀組塊長序列的最新或"新"資料。此可能組合對於充分靠近可能係複製資料者的"新"資料使用更小塊。依此方式，一些類型的備份資料可能更容易係可能對應於局部變化的括號插入或刪除區域。為此可能需要一極受限制的預看並使用一預看緩衝器及每輸入單元一限定數目的查詢來決定是否先前已發射一預期塊。該等查詢可針對一後端儲存單元及/或在實際或虛擬記憶體(例如一布魯姆過濾器)內所維持的一有限大小之局部歷史來確切或立即進行。本發明之各種具體實施例已展現增加平均資料群組、區塊或塊大小並維持一可接受(或甚至改良)複製消除比之能力。此外，各種具體實施例已能夠使用比先前可獲得者大大約2

至4倍之資料群組、資料區塊或資料塊來獲得大致相等的複製消除比。該些更大資料群組、區塊或塊可提供一儲存資料之元資料成本降低並在傳輸資料及/或讀取或寫入另一裝置，例如一備份器具(例如一磁碟儲存系統)時可引起輸出量增加。

本發明可能係一電腦實施之發明，其包括用於改良資料處理效率而不顯著降低資料處理結果之品質的軟體及硬體。在至少一具體實施例中，本文所提供的該(等)系統及方法可使用一計算裝置來實施，並可在一網路內的一或多個電腦上運作。稍後參考圖19及圖20在某細節上說明(多個)範例性計算裝置及網路之細節。預先參考該等範例可能證實有助於促進更清楚瞭解本發明之各種細節。

在任一情況下，為了方便理解，將會針對與雜湊功能及/或資料冗餘識別及/或資料複製消除一起使用來更詳細地解釋本發明。然而，習知此項技術者將會瞭解，本發明可應用於其他資料管理及處理系統及方法，包括要處理或儲存一資料串之電腦、具有資料要傳輸之無線通信、網際網路及企業內部網路應用、資料加密技術等。特定言之，本文用以解釋本發明之該等範例性具體實施例主要係關於資料雜湊及資料複製消除。

在全部檔案雜湊之情況下，可藉由施加一雜湊函數至整個檔案之所有資料來執行雜湊。例如，可使用一SHA-1雜湊函數並將其施加至一整個資料檔案。該SHA-1雜湊函數計算上較複雜且可能相對於一些其他雜湊函數而言速度較

慢。無論如何，在此情況下，為了識別並消除複製之目的，發現並消除最少數量的資料複製，因為當一單一位元資料在一檔案內變化時，所產生的雜湊值可能不同於先前保存者且將需要傳輸或保存與修正後檔案相關聯之全部數量資料(例如當改變在一文字檔案中的一字母時，該文字檔案之整個資料表示及其雜湊值將會變化，使得其不再係相同文字檔案之一先前版本之一重複)。另一方面，因為該雜湊函數僅需針對一整個檔案的資料操作一次，故該雜湊速度較快。

一固定大小資料區塊雜湊函數可在一全體檔案中發現的整個資料之多個部分或區塊(例如一單一文字檔案可打散成多達10個相同大小的10K位元資料區塊)上執行雜湊，並可以一不重疊的固定大小來設定資料區塊。對於各種用途，編塊或組塊之速度較吸引人。最簡單且最快速的方案係將輸入流打散成固定大小的塊。在一些再同步檔案同步技術中可能採取此方案。然而，使用固定大小組塊時會存在一潛在再同步問題；考量在一檔案開始附近進行一插入或刪除編輯時所發生的情況：在改變一單一塊之後，將會改變整個後續組塊。一新版本的檔案將可能具有極少的複製塊。同樣，對於固定大小區塊，可將一SHA-1雜湊函數施加至一組成一全體檔案(例如100K位元)之固定大小區塊組之各區塊(例如10K位元)。在此情況下，因為每次雜湊的資料區塊會更小，故可發現更多複製，且在一全體檔案內某處的一單一位元變化將只會導致組成一全體檔案之多

個區塊變化(例如10個10K區塊之9個將會係重複)。區塊越小，冗餘偵測便越佳，但可能導致一略微更緩慢的處理，因為必須對於在全體資料檔案內發現的相同數量資料更多次運行雜湊函數(例如SHA-1)。

一化解該再同步問題之方法係藉由使用內容定義組塊(CDC)技術；其中一局部視窗(例如12至48位元組長)可用於將資料流可重製地分成可變大小塊。給定隨機資料輸入，在某種程度上控制平均輸出塊大小或平均塊產生速率之意義上，此類組塊係機率性的。CDC產生可變大小的塊，其更能夠抑制一局部編輯對一有限數目塊的改變。應用CDC可包括數個類型的網路檔案系統、參考檔案集合之空間最佳化歸檔及檔案同步。低帶寬檔案系統(LBFS)在建立CDC成為一廣泛使用技術方面具有影響力。通常，基本組塊演算法一般會增加在最小及最大型塊尺寸上的限制。在一些情況下，在到達最大型塊大小之情況下或在輸入資料流內發現輔助特殊資料序列之情況下，可進行一更複雜的決策。

該內容定義資料塊雜湊可藉由施加一相當緩慢且稍微更佳執行(例如更精確且有差別)的計算來識別並產生一用於其內容所定義之各種資料塊之值來加以執行。一此類雜湊函數可包括一拉賓(Rabin)辨識與SHA-1雜湊函數組合。可多次施加該拉賓辨識用於交迭資料檔案內資料之資料視窗(例如滑動視窗)，以基於一預定邊界點準則(例如在指紋內的一組預定位元係0)來決定在一資料檔案內應設定塊邊界

之位置，接著可將該SHA-1雜湊函數施加至該等決定資料區塊之各資料區塊(其大小係基於正被分析的下面資料而變化)。再次，各輸入位元可鍵入於一些SHA-1雜湊計算內，如先前所述。然而，比較固定大小組塊時，該拉賓辨識呈現一額外計算負擔(例如處理時間)。儘管此方案極善於識別更多資料冗餘，但該些函數二者可能較耗時且組合時可能使雜湊及/或資料冗餘識別及消除極為耗時。實際上，該拉賓辨識函數可能在試圖最佳化冗餘資料識別及/或資料消除時用於識別各種資料區塊切割或雜湊點應該的特定位置特別耗時。

在關於資料歸檔或備份之各種具體實施例中，在藉由使用一基本組塊技術到達最大區塊或塊大小時可使用一備份切割點、塊點或斷裂點之優先化階層。此技術可能由增加能夠再同步一包含局部編輯之輸入流之可能性的一願望推動。該技術可增加一塊組合或合併技術或一塊分割或打散技術。藉由考量組塊一隨機資料輸入之統計，本文中將說明可有效地使用此類方案而具有超過(多個)先前系統及方法之顯著改良。該基本組塊技術可使用一參數係最小及最大型塊長度之內容定義組塊方案來操作，並可能包括一備份切割點、塊點或分裂點之概念，並可在本文中稱為"基線"複製消除技術或功能。

複製消除(DE)之一益處在於，對於許多類似檔案系統之輸入，可識別大多數重製內容。比較主動搜索在某些壓縮相關意義上最佳的(多個)複製區域，複製消除無效率主要

出現於橫跨新舊資料之間邊界的資料群組、區塊或塊處。故隨著資料群組、區塊或塊大小遞減，複製消除之效率一般會遞增。然而此可導致每塊(群組或區塊)儲存成本的增加(更多元資料，更長擷取及重新裝配時間等)。

複製消除比(DER)可定義成輸入位元組/儲存位元組比(即使用位元組作為資料單位，輸入資料/儲存資料)。考量資料歸檔應用，目前資料歸檔系統可獲得大約5至20之DER。一考量係對於DE目的每塊元資料可能如何影響所需資料群組、區塊或塊大小。例如假設每資料群組、區塊或塊具有一100位元組的管理資訊以指示可儲存資料群組、區塊或塊之位置及方式。若此類系統支援一最大(例如)50 DER，則平均資料群組、區塊或塊大小應至少為 $50 \times 100 = 5000$ 位元組(在此資料群組、區塊或塊，添加一複製仍耗費100個位元組)。

若每塊成本較高，則可能有動機去產生更大資料、群組、區塊或塊大小。由此，典型複製消除方案使用在(例如)2k至16k範圍內的資料群組、區塊或塊，且明確不嘗試圖在短字串位準上移除冗餘。此類冗餘可藉由在個別塊上使用一快速"局部"壓縮方案來移除至一有限程度。再次，在可使用更大平均大小的資料群組、區塊或塊之情況下，每塊(群組或區塊)之效率可能會上升。然而值得主要的係，比較複製消除所節省之資料儲存空間數量，因為其僅每儲存塊發生一次，此特定效益仍有些微小，而在每次碰到塊時均出現由複製消除所產生之增益，對於一般資料，

其可能係5至20次。

本發明之一目標可能係發現一改良複製消除方案，其可相對於傳統("基線")複製消除方案，增加DER及/或平均資料群組、區塊或塊大小。本發明之各種類型應用可能要求依據用於該應用之資料集或資料流之獨特方面或特性來發展本發明之特定功能。例如，兩個主要原則可應用於改良處理及管理歸檔或備份資料流：原則1，即長段新或先前未見過資料可假定為稍後出現(例如在下一備份列)的較佳候選者；及原則2，即可藉由更短塊來最小化橫跨於複製資料與新(先前未見過)資料之間邊界的"變化區域"周圍發生的無效率。在此情況下，原則1可藉由對於"典型"檔案或備份資料(例如家庭目錄、電子郵件等)，用於歸檔或備份資料之DER經常位於5至20範圍內之事實來證明是合理的。另一方面，原則1可能事實上對於具有一較高內容滾動或插入頻繁非複製標記之內容資料流的系統未被證明是正確的。原則1支援使用可產生具有較大平均大小之資料群組、區塊或塊的一資料分組、編塊或組塊技術或程序，特別係在新或先前未見過資料之一延伸區域內時。

根據原則2，可能引起該基線技術或程序之許多變化方案：主要不同在於識別該等變化區域之方式以及以更精細粒度分組、編塊或組塊在一變化區域內新或先前未見過資料之數量的變化方案。一方案可開始於一初始技術，其可先將一切分組、編塊或組塊成平均包含大量資料的資料群組、區塊或塊。隨後，可識別包括複製及非複製資料之區

塊，並選擇該些區塊之一子集。此選定子集可接著使用一打散或分割技術以在一更精細位準重新組塊該等識別資料區域。此方案之一優點可能係可能要求很少用於複製塊的查詢。然而，在此類方案中，一更小插入/修改可使一整個大型塊無效。若複製消除比(DER)效能比最小化查詢數目更重要，則在下一段落中所論述的合併程序可能係一更佳選擇。藉由在較大型塊可位於之位置方面提供更多彈性，可獲得一稍微更佳的DER。

在可能係一略微更具彈性方案者中，可最初以一精細位準塊並將較小塊組合成更大者之資料構建或合併技術或程序可能較佳。此類方案可能藉由允許大型塊形成方式的一增加數量可能性而更具彈性。該構建或合併分組、編塊或組塊技術可更精細地分類，並稍後再利用一單一插入/修改塊。然而，如上所述，提供如何構造較大資料群組、區塊或塊之更多變更或可能性可能會耗費一增加數目的查詢以決定不同的預期大型群組、區塊或塊是否可能係重複。數個變化方案可能由於限制在進行一資料分組、編塊或組塊決策之前所需之查詢數量而不同。

一般而言，可用於建立更大資料群組、區塊或塊(例如每資料群組、區塊或塊的更大數量資料)以及分類變化區域之彈性越大，在同時增加DER及資料群組、區塊或塊大小方面技術或程序效能便可能越佳。在(例如)一資料歸檔設定下該些技術之確切行為可能取決於頻率分佈、複製區域長度及從一備份序列或會話至下一備份序列或會話的平

均內插或修改"編輯"大小。

特別相對於涉及待儲存資料數量或檔案數目的具體實施例，本發明之各種具體實施例可能既快速又可縮放。若放鬆該些要求，則本發明之各種資料編塊技術可耦合更複雜的方案。例如，各種具體實施例還可包括儲存關於一大型塊所包含之特定子塊的資訊，並產生用於儲存並重組一資料流之更複雜壓縮樣式配方。此外，本資料分組、編塊及組塊技術或功能可耦合更具挑戰性的技術，例如相似性偵測與增量壓縮。然而，基於主動發現"類似"檔案或塊之各種方案可能通用性較低，因為其傾向於線性縮放或隨儲存塊數目而變差。更佳方案可能係使用主要由上述簡單的原則1及原則2所支配的稍微"被動"技術，其可在恆定時間內運作，由於其可能僅要求一限定數目的資料群組、區塊或塊存在查詢，以進行最後資料分組、編塊或組塊決策。為了與資料歸檔及備份應用一起使用，本發明可使用(例如)一32k平均資料分組、區塊或塊大小之標的。此標的可能有用於獲得所需位準的一讀取/寫入要求與元資料儲存成本組合(更大型塊有助於該些目的)。然而，還可能需要獲得與一般使用僅平均4k至8k塊大小之現有複製消除方案相當的DE位準。

本發明可包括一用於滾動視窗切割點、斷裂點或塊點選擇之多重選擇準則技術。內容定義組塊一般藉由選擇一組預定位置以打散一輸入流來操作。在組塊決策(切割點決定)基於在一局部視窗內所含的資料時，組塊係內容定義

的。一般目標係若產生切割點之視窗集係充分隨機的，則對於實際資料輸入，塊大小分佈仍將類似於隨機輸入的統計期望值。為了獲得快速切割決定及合理較佳的執行切割點二者(如在同日申請的授予 Cezary Dubnicki、Erik Kruus、Cristian Ungureanu及Krzysztof Lichota之相關專利申請案，標題為用於使用多重選擇準則之資料管理之方法及系統中所述，其係以引用版本併入本文)，一第一快速選擇函數(例如一箱車和)與一第二更緩慢選擇程序(例如CRC32c或SHA-1)可用於決定初始切割點組。

該兩階段切割、斷裂或區塊點選擇程序可選擇以影響該箱車和之速度。第一，一相當樂觀的選擇可使用一快速箱車和來進行以產生一組潛在切割、斷裂或區塊點。根據此組，一第二(更緩慢)雜湊函數可用於選擇最終切割、斷裂或區塊點。該些"組合"組塊函數對於大多數輸入視窗在該箱車雜湊之速度(一般記憶體帶寬受限)下操作，並僅偶而需要使用該更緩慢雜湊函數來加以評估。由於該第二雜湊函數不必滾動，故可使用任一雜湊函數而幾乎不影響速度。在各種具體實施例中，可能較有益的係組合該更快速箱車函數(例如雜湊函數)用於一初始選擇並隨後使用一更緩慢CRC32c或SHA-1函數(例如雜湊函數)用於最終選擇。可能的組塊切割、區塊、雜湊或斷裂點可藉由諮詢雜湊值之最低有效零位元數目來決定。若此零位元數目係 l ，則吾人會說，此視窗在位準 l 產生一切割點。在表1內說明各種範例性切割、斷裂、雜湊、區塊點函數之一表格，其顯

示在模擬期間的各種速度獲得。

名稱	碼	速度MB/s
箱車	雜湊 +=b	360
roIN-xor	雜湊=ROL(雜湊, N) ^ b	280
MLCG	雜湊=雜湊* A+b	270
Rabin	雜湊=((雜湊<<8)lb) ^ A[雜湊>>N]	195
xor	雜湊 ^=A[b]	175
roIN-xor[]	雜湊=ROL(雜湊, N) ^ A[b]	175
xAdler	s1 +=b; s2 +=s1; 雜湊=s1 ^ s2; 雜湊^=雜湊>>6;	160

表 1

在表 1 內上面提供一些簡單雜湊函數。雜湊係指派一恆定初始值。對於在視窗內的各位元組，藉由添加下一位元組值 b 來修改雜湊。此處所列之該等雜湊函數具有快速滾動版本。此處 A 及 N 係雜湊特定常數而 ROL 係一左旋轉操作。該 xAdler 雜湊係 Adler(/Fletcher) 雜湊的一簡化版本。速度值表示具有一最大數量編譯時間常數之相當程度最佳化碼，並可提供雜湊函數速度的一合理指示。該等速度測試測量滾動雜湊及數百百萬位元組記憶體內隨機資料上檢查終止零位元。

在各種具體實施例中，該(等)系統及方法可包括產生並維持多個備份切割、區塊、雜湊或斷裂點。普遍設定用於任一滾動雜湊運算法之參數可能係：最小塊大小 m；最大型塊大小 M；組塊"位準" ℓ ；及備份切割點數目 b。對於隨機輸入資料，任何在位準 ℓ 切割的視窗核對和應產生一平均塊大小 $\bar{L}=2^{\ell}$ 。該期望平均資料塊大小係在使用一包含

隨機輸入資料之輸入流呈現時選擇組塊邊界之統計期望率。用於塊大小 s 之統計期望值應遵循一具有平均值 $\bar{L}$ 之指數分佈：

$$P(s) = \frac{e^{-s/\bar{L}}}{\bar{L}} \quad (1)$$

在 S 個輸入位元組之任一序列中，確切看到 k 個切割點之機率應遵循一波松(Poisson)分佈：

$$P(S, k) = \frac{(S/\bar{L})^k e^{-S/\bar{L}}}{k!} \quad (2)$$

在區域 S 內確切具有一切割點 ($k=1$) 之機率在 $S=\bar{L}$ 時最大化。對於最小塊大小 m ，平均塊大小應在 $m+\bar{L}$ 周圍。對於一最大塊大小 M ，一平常位準 ℓ 組塊程式將會命中具有機率 $e^{-(M-m)/\bar{L}}$ 之最大值。該指數分佈具有一相當長的末尾。若該函數在一 $(\bar{L} \approx 2') \approx (S \approx M-m)$ 體系內操作，則其可能產生一最大大小塊 $e^{-1} \sim 37\%$ 的時間。此係一相當大部分的塊，且在此情況下可選擇一替代性切割點。一易於維護的替代方案係可能在位準 $\ell-1$ (及等) 碰到的任一切割點。在此第一備份位準，在一平均塊大小 $m+2^{\ell-1} \sim m + \frac{M-m}{2}$ 情況下，吾人具有零切割點之可能性較低 ($\approx 10\%$)。若在此位準存在一切割點，則將會比在第 M 個資料位元組的一任意切割更清楚地提供一更佳 (例如) 再同步二輸入流之機會。藉由維持若干備份位準用於切割點，可減少不具有任何獨特切割點之機率。在本文所述之各種實驗中經常使用 3 個備份切割點且其係一實際選擇。在一位準具有數個備份切割、區

塊、雜湊、斷裂點可能實現選擇以在最早或最晚的此點進行切割。為了複製消除之目的，在最早切割點的切割可能一般提供最佳再同步。在一稍後切割點的切割可能以增加鬆弛為代價來提供最大的塊大小增加(例如參見美國專利公告案第2006/0047855號)。在本文所呈現的各種實驗資料中，可選擇該等備份切割、區塊或雜湊點以產生最長且最大的區塊或塊。現在將參考圖示來提供本發明之各種具體實施例之一更詳細論述。

圖1提供依據本發明之至少一具體實施例之一資料管理系統100之一範例性概略，其可識別在一輸入流內的資訊並可有時在產生輸出之前修改該資訊。可提供一輸入資料流110。輸入資料流110可能係一資料流或資料集並可包含(例如)一或多個電子位元/或位元組資料。資料流110可能來自任一數目或種類的設備或來源(例如在一PC上操作的電腦程式、一LAN、網際網路、一儲存磁碟、一無線手持裝置、一微處理器輸出、一記憶體晶片、一記憶棒、一接收傳輸等)之一或多個者。一資料識別系統120可識別各種位元及/或位元組的電子資訊並可在各種記憶體、緩衝器等內儲存其用於使用(其可包括於一或多個電子裝置內)。該資料識別系統還可包括各種斷裂、區塊、塊或雜湊點決定函數。此資料及資訊可供一資料操縱系統130使用。在各種具體實施例中，資料操縱系統130可包括一資料雜湊函數、一複製消除函數、一雜湊表、一壓縮函數等，並可包括於一電子及/或電腦系統或網路內。但基於資料類型

及特性與資料操縱系統130之用途或應用，有時可能或可能不操縱來自資料流110之資料。在任一情況下，一旦資料已穿過資料識別系統120及資料操縱系統130，則其一些或全部可輸出作為一輸出資料流140。此輸出資料流可隨後以任一電子資料一般所經歷之方式來進一步加以處理、傳輸、儲存等。在本發明之各種具體實施例中，可獲得該輸出資料流並將其作為備份資料儲存在若干儲存媒體之任一者上。

現在參考圖2，提供一資料識別程序200之一範例性概略。資料識別程序200可利用(例如)一第一選擇函數220(可包括上述的一第一快/慢多重選擇準則)並接著可利用一第二修改選擇函數，該第二修改選擇函數可由一合併230及/或打散240特定現有塊之內容定義組塊策略組成。該程序可在步驟210開始，其中將一輸入資料流輸入一資料識別系統(例如系統120)。接著，在步驟220，一內容定義組塊程序(函數、演算法等)可應用於該資料流以獲得建議的資料切割、區塊、斷裂或塊點。該內容定義組塊程序可包括一第一快雜湊與一第二慢雜湊函數或程序(例如針對前述相關專利申請案中所揭示之技術)。接著，可執行步驟230或240之一或二步驟。在步驟230，可藉由施加一區塊或塊組合程序來修改該等提出資料切割、區塊、斷裂或塊點以基於一特定規則來合併現有塊，下面將更詳細地予以說明。在步驟240，可藉由施加一區塊或塊分割或打散程序來修改該等提出資料切割、區塊、斷裂或塊點以基於特定

規則來將現有塊分割成更小塊，下面將更詳細地予以說明。在步驟240，來自該些程序之輸出資料可決定將該等切割、區塊、斷裂或塊點或發射塊發送至一資料操縱系統130。

圖3a及3b係可用於(例如)圖2之步驟230之一範例性內容定義組塊策略之資料流。此範例性策略可能係：(i)在長段非複製資料內將小塊合併成大型塊，同時(ii)藉由不合併小塊，使用小塊來界定相鄰複製資料之非複製資料區域之該等邊緣及/或(iii)重新發射經發現係重複的大型塊。在此範例性情況下，原則1及2可應用於設定資料切割、斷裂、區塊或塊點，特別係在資料流之較大字串可能隨時間複製之情形下，例如在歸檔或正在處理備份資料時。在圖3a及3b中，該等資料流300及350分別說明資料組塊程序，其可包括一用於改變或修改一建議用於其資料流之組塊之程序。在圖3a中，在第一更早時間組塊時間(例如一第一周)的初始提出資料流切割、斷裂、區塊或塊點可能包括資料流之一摘要，此處顯示為330。在此情況下，可能存在一長序列(例如30)資料群組、區塊或塊，其可能已由一內容定義組塊程序(例如在圖2中220)產生，其中沒有發現任何複製資料。非複製狀態指派係指示為標號"n"。如所指示的，資料組塊程序300可決定該等30個個別塊之各塊不複製以及決定五個鄰接塊301-326之所有合併不複製。在此情況下，在程序300內所說明的塊合併策略230可選擇以藉由合併連續五個小塊群組來形成大型塊，從而產生一輸出

流 340，其由(例如)六個大型塊 341 至 346 組成。

在圖 3b 中，其可對應於(例如)一第二稍後時間，輸入流 351 之一摘要之建議組塊再次包含 30 個塊。如所示，資料流 351 可能包括視為一起的複數個複製資料區塊或塊，例如首五個塊 352 以及視為一起的最後五個塊 357 可能係重複(指定為 "d" 並稱為大型重複)。由於保留複製大型塊較有利，故可選擇在輸出流 380 內合併包含 352 及 357 的該等小塊，從而產生大型塊 390 及 391。該等兩個複製群組 352 及 357 可能以一已指派(例如)非複製狀態之二十資料區塊或塊 353 至 356 之字串為邊界。在區塊 353 至 356 內，"n" 指定無法作為個別塊或作為任一五個連續塊群組發現任何複製。原則 1 建議在塊點 360 與 370 之間的此大量非複製資料可使用一塊組合或合併程序而分組在一起。然而，基於原則 2(不產生橫跨或充分靠近在視為複製資料與非複製資料之間邊界的大型塊)，可能最佳的係劃分非複製區域 360 至 370 以包括初始及最終小塊緩衝區。若變化區域 360 至 370 係充分小(例如小量連續資料群組或非複製資料之"短孔")，則該程序可選擇以發射整個鄰接塊區域而不執行任何合併。或者，如在範例性輸出流 380 中所呈現，該組合或合併程序可選擇將來自 360 至 370 之區域作為三個大型塊 395、396 及 397，之前有兩個更小塊 381 及 382，並跟隨三個更小塊 383、384 及 385 來發射。如此，在從認為複製區域轉變至認為非複製區域之後及/或在從非複製區域轉變至複製區域之前，輸出流 380 可包括一未合併小塊轉變區

域。一併入程序300及350以產生輸出流320及380所示之效果的程序可藉由維持一可接收數量的複製偵測及/或增加平均儲存或傳輸資料片段大小來為(例如)歸檔應用、網際網路通信等的典型資料流產生改良效能。

參考圖4a及4b，依據本發明之至少一具體實施例提供資料流400及450，其顯示一範例性內容定義組塊策略，其可打散或分割在以複製資料為邊界之鄰接非複製資料開始及結束附近的更大型塊，以便更佳識別在此區域內的可能複製。資料流400及450說明資料組塊程序，其可包括一用以改變或修改該等提出組塊410及460之程序(對應於(例如)圖2之步驟240)，以便產生範例性輸出420及480。

在圖4a中，在一第一更早時間(例如一第一周)時的程序220之輸出可包括資料流之一摘要，此處顯示為410。在此情況下，可能再次存在一已歸類為非複製(指定為"n")的資料群組、區塊或塊序列。在410中顯示六個此類塊。依據原則1，該組塊修改策略可選擇以在其輸出流420中呈現這些相同塊，預期這些大型塊可能有極佳機會在稍後再次出現。在圖4b中，在一第二稍後日期(例如一第二周)時圖2之程序220之輸出可能包括資料流460之一摘要，其中此時可能決定一些資料群組、區塊或塊複製(461及466，指定為"d")，同時已決定其他462至465不複製(指定為"n")。由於一般情況需要大型複製輸出塊，故一塊修改策略可決定傳遞未修改的複製塊461及466至該輸出流。在此情況下，在切割、斷裂、區塊或塊點470及475之間的區域完全由已識

別為非複製之群組、區塊或塊組成。一組塊修改策略在已決定點470開始一非複製區域之後，可能選擇使用一產生更短平均塊長度之替代性程序來分割一初始塊462，同時不修改內部非複製塊的一區域。此相同策略在已決定點475起始跟隨一非複製區域的一複製區域之後，還可能決定使用(例如)一產生更短平均塊長度之替代性程序來分割一前面塊465。如所示，初始非複製資料區塊462可能產生複製數個更小型區塊或塊481至485，同時最後非複製資料區塊可能在輸出流內產生該等四個區域491至494。依此方式，該等更小塊可能在捕獲更多複製資料區塊時更佳，由於此策略可能能夠在原始資料流460內更佳界定變化區域開始及結束。例如，如圖4b所示，該輸出流可能實際上複製區塊481、482及494。

習知此項技術者應認識到，與資料流300、350、400及450相關之程序之特定實施方案可在如何明確定義複製及/或非複製區域、小塊緩衝區可選擇的長度、一小塊或大型塊之大小等細節上變化。小(或更小)塊可能具有大約2至18k之一般大小，並對於隨機資料輸入可能展現一重新裝配一指數分佈之塊大小分佈。在一簡單合併具體實施例內的更大型或大塊可能在一對應更大型塊大小範圍下始終由(例如)確切八個連續小塊構成並可平均為一大約16k至144k(例如8倍大)之大小，並可能展現一更狹隘的塊大小分佈，其比在使用隨機輸入資料呈現時的指數更狹窄。

在各種具體實施方案中，一第一片段化程序可跟隨有一

第二片段化程序，且該第一資料片段化程序與該第二片段化程序之期望平均資料群組大小可能不同。該期望平均資料塊大小係在使用一包含隨機輸入資料之輸入流呈現時選擇組塊邊界之統計期望率。該期望資料群組大小可能與一選擇準則相關，如在授予Cezary Dubnicki、Erik Kruus、Cristian Ungureanu與Krzysztof Lichota之相關專利申請案用於使用多重選擇準則之資料管理之方法及系統中所解釋或如上示等式(1)所提出之用於塊大小 s (不存在最小或最大塊大小限制)之理論期望值所建議。一些實施方案可能喜歡利用下面資料流之實際長度或數量來支配合併限制或關於複製或非複製狀態指派的各種限制或關於應合併資料數量的限制，而不使用"塊"作為計算單位，如在圖3中。其他實施方案可包括可合併該等小塊之若干方式或程度。一些實施方案可能額外或替代性地選擇以使用輸入流之另外(及確實實質不同)特性來引導小塊與大型塊之間的決策程序。例如，實施方案可包括用以引導從先前組塊所輸入及/或輸出之統計模型中所產生之組塊策略230及/或240以引導不同期望輸出塊長度之組塊程序之間的轉變。其他用以此類策略之有用輸入可包括基於(例如)先前歷史或快速熵估計方法(例如雷尼(Renyi)熵)的可壓縮性估計。

一些實施方案還可包括基於關於本複製消除目的之目前結果狀態來設定或動態改變組塊大小之可能性。例如，若該程序正在獲得複製估計目的，則該程序可切換至組塊新非複製資料與一更大型平均塊大小並還降低元資料。或

者，若該等複製消除目的正無法得到滿足，則該程序可切換至略微更小的大型塊大小。一些實施方案還可包括基於關於平均資料群組大小目的之目前結果狀態來設定或動態改變組塊大小之可能性。若該程序之輸出流之平均塊大小正在超出目標，則在能夠稍後為連續數目的一可變數目小塊(直至某最大值)適當操作複製查詢時，(例如)該程序可在全新資料區域內合併一更小數目的塊。一些實施方案還可包括基於相對於較小及/或更大型資料群組之估計或實際可壓縮性或基於複製估計特性之理論或實際模型來設定或動態改變組塊大小之可能性。若一先前效能評估(例如)顯示在一小塊位準，例如輸入一非複製區域相當可能涉及一初始小塊序列，其首三個特別可能複製，而下8或以上特別可能不複製，則該程序可調整在邊界區域內的小塊數目至(例如)三個小塊(而非(例如)四或兩個)。一些實施方案還可包括多次施加第二資料片段化程序，以便進一步改良最終塊大小及/或DER。例如，可修改該程序用以分割；假設一初始小組塊開始於具有複製狀態之區域之一更精細粒度延伸，則可在甚至更精細尺度上想像分割該第一非複製區域以更緊密地終止該複製區域。當與遞增越來越小塊數目相關聯之元資料成本超過該等DER效益(可能基於預定或動態變化目的)時可終止此類程序。

應注意，用於圖3a、3b、4a及4b之輸入資料流可能對應於一原始資料流之相同摘要。在此情況下，可能有益的係考量並比較指派複製/非複製狀態所需之額外處理努力之

主要來源。在圖3a及3b中，用於複製狀態之查詢數目可能上限為每個小塊兩個(對於一塊複製最大一查詢及對於一五個鄰接塊複製一查詢)。故在相同輸入流330及410之情況下，查詢努力係對於圖3a限定為 $30 \times 2 = 60$ ，而在圖4a中為6。一合併程序之其他實施方案可能允許可變大小大型塊，其中若干鄰接塊可合併直至某預定最大值，或直至下面資料之某預定最大數量。此仍產生每單位輸入一恆定數量的查詢，但該常數更大。該些考量可能對於系統設計具有實際影響。實際上，儘管許多程序能夠在使用已指派非複製狀態之鄰接資料區域呈現時產生重新組裝輸出流380及480之輸出，但一有用經驗規則係具有更多"彈性"之程序(即允許大量位置及/或長度用於合併塊)可獲得一更高數量的複製消除。此效能效益可能需要使用額外查詢成本來加以權衡。應注意，不論先前儲存資料之規模如何，此成本大致恆定，而在一歸檔系統內包括重新裝配偵測及增量壓縮的一些技術可能隨著儲存資料數量而線性縮放(或變差)。該等提出系統及方法之此較小額外成本與極佳可縮放性可使該些方案較佳地用於為資料歸檔、資料通信系統等改良複製消除及資料程序速度的其他可用系統及方法。

該等前述程序之各程序可使用一相對簡單的程序、軟體或演算法來執行，因而將僅要求小量程序時間，同時有可能改良複製資料辨識及改良DER。此外，各程序可在各種具體實施例中使用一較短或較小預看緩衝器來實施。在將小量資料插入一大型複製資料流內時，此預看緩衝可能甚

至有助於輔助再同步。在該預看緩衝器不識別在當前資料流內與先前資料流的一偏移情況下，可能需要一主要再同步。此再同步可基於在處理中資料流內的特定資料類型來加以預定。各種具體實施例還可包括一或多個備份切割、斷裂、區塊或塊點以輔助再同步。在任一情況下，現將提供複數個更詳細流程圖及範例以進一步說明本發明之一些具體實施例。還應注意，儘管曾在圖3a、3b、4a、4b所示範例中使用相等長度來繪製各種大型及小塊，但個別塊可包括一變化數目位元組的下面資料流，且與所謂小型及大型塊相關聯之實際資料流長度可能從一小型/大型塊變化至下一此類塊。例如，某些大型塊，其資料內容嚴格大於該等下面非合併塊之任一者之資料內容，可能小於在輸入資料流之其他區域內所產生的個別小塊。因而，該等塊可能實際上係可變大小的且並非如該等圖示所提出那樣係固定大小的。

在產生可變大小塊方面相當多的彈性可能藉由先運行一產生更小塊之程序，隨後將塊合併成大型塊來提供。存在許多可能方式來進行合併決策，且一重要差異係可能需要進行組塊決策的每小塊查詢數目。例如，一簡單方案可能係製造藉由序連一固定或給定數目(N)個連續小塊所產生之大型或大塊。此方案可能限定大塊存在查詢數目至(例如)每小塊1個。另一可能性可能允許大型塊為2..N連續小塊，具有一更高的所需查詢數目限定。使用關於不發射小件(即因為其係某大型塊之部分而不發射的小塊)之資訊可

能不合需要，由於可能存在不知何故需要儲存之大量小塊"元資料"。

一簡單範例性塊合併程序或演算法可能係(例如)：

```

1 void process( SmallChunkLookAheadBuffer buf[0 to
    2k-1]) {
2     if( isDupBig(buf[0]) ) emit big @ buf[0 to k-1];
3     else if( (then=findDupBigStarting(buf[1 to k]) )
4         emit smalls until then, followed by a big;
5     else if( isPrevDupBig ) emit k smalls;
6     else emit big @ buf[0 to k-1]
7 }
```

此類程序藉由允許精確定義組成步驟之各種合理方式來容許彈性。此處，"buf"可能係指一預看緩衝器，其包含(例如)多達 $2k-1$ 個塊。"isDupBig(buf[0])"可檢查以查看一第一小塊是否可能係一先前發射合併塊之部分。在各種具體實施例中，例如，此可能要求檢查包含buf[0]至buf[k-1]之一單一大型塊是否係一重複。若是，則buf[0..k-1]可能再次作為一大型塊發射("emit big @ buf[0 to k-1]")。在其他具體實施例中，isDupBig(buf[0])可能藉由查詢是否針對在k-1至2的某N先前已發射任一合併buf[0..N]來檢查複製狀態，且若是，則可發射包含buf[0]至buf[N]之一合併(大)塊。"findDupBigStarting"可能調用一類似於"isDupBig(buf[0])"之程序，除了可能在該預看緩衝器buf內的增加偏移開始依序調用isDupBig。若發現先前已發射此類預看最長、第

一合併塊，則在行4，可在buf[0]起始發射小塊，隨後發射由行3發現的第一合併塊。"isPrevDup"查詢一狀態變數，其係用於緊鄰buf[0]之前發射之塊之複製狀態。其可實施成為一反映先前發射之指派複製狀態的狀態變數及/或在先前發射係一由多個小塊組成之合併塊之情況下藉由儲存真來實施。其他具體實施例可代替指派一"真"複製狀態給任一複製先前發射(一或多個小塊)。此外，大型塊形成準則可包括將大塊限制成超過在該大塊下面之資料流大小之一預定臨界值者。出於簡化，吾人將在此以後提及由一合併程序期間僅由小塊計數所決定之大塊。使用固定大塊情況下，該程序可能每小塊進最多1個查詢，而對於可變大小塊，可能每塊進行多達k-1個查詢。其他變化方案包括修改是否認為一單一小塊足以產生一(邏輯)複製/非複製轉變之概念。若不使用一預看緩衝器來寫入此類演算法，則可不使用findDupBigStarting()預測。在此情況下，兩個複製輸入流可能不確定地保持不同步。此可藉由引入再同步切割點來加以避免：在一小塊之切割點位準超過某臨界值的任何時候，可允許該小塊終止一大塊，但可不允許在一大塊內部的其他地方出現。依此方式，可在不具有足夠預看以自發進行再同步之演算法中在一再同步切割點之後，可強迫再同步兩個複製輸入流。

現在參考圖5a及5b，提供一內容定義組塊策略之一範例性流程圖500，其可利用一固定大小預看緩衝器來將小塊合併或組合成在長段新非複製資料內的大型塊，同時使用

小塊界定非複製資料之該等邊緣並在較短非複製資料區域內留下小塊。首先，在503，一輸入緩衝器或預看緩衝器包括若干更小塊。接著，在決策步驟506，決定在該緩衝器是否存在足夠小塊以發射一更大資料塊。若否，則在步驟509，可從該緩衝器發射1小塊並在步驟524，可在最後發射塊之後使用下一塊填充該緩衝器。若在該緩衝器存在足夠小的塊以發射一大塊，則在決策步驟512，決定該緩衝器是否開始一先前發射大塊。若是，則在步驟515，可發射該大塊，隨後重新填充緩衝器步驟524。若否，則在步驟518，一決策步驟決定該緩衝器是否具有一足夠靠近該緩衝器開始以防止發射一大塊的先前發射大塊。若是，則在步驟521，該程序將發射在該緩衝器開始與該大型塊之間的小塊並發射該大塊，隨後重新填充緩衝器步驟524。然後，若該緩衝器不具有一足夠靠近該緩衝器開始以防止發射一大塊之先前發射大塊，則該程序進行至步驟決策步驟556。在決策步驟556，在沒有任何先前發射大塊可能足夠靠近該緩衝器開始以防止發射一大型塊(緩衝器開始加上大塊長度)之點之後，決定該緩衝器是否具有一小塊，以允許一隨後大塊之構造。若否，則在步驟568，該程序決定發射的最後塊是否係一先前發射大塊。若此係是，則在步驟571，該程序可發射1個小塊，隨後重新填充緩衝器步驟524。若步驟568的答覆係否，則在步驟565發射一新的大塊。若在步驟556決定可構造一隨後大塊，則執行決策步驟559。在決策步驟559，決定在步驟524所發

射的最後塊是否係一先前發射塊或是否先前發射過隨後大塊。若決策步驟559的答覆係否，則在步驟565發射一新的大塊。若在決策步驟559的答覆係是，則在步驟562在決定下一可能大塊之前發射所有小塊。在任一情況下，重新填充緩衝器步驟524隨後。

參考圖6a、6b及6c，依據至少一具體實施例說明應用一內容定義組塊合併策略至一資料輸入流之一範例性圖解600。首先，顯示一輸入資料流605由一系列範例性小塊雜湊值所組成，其中各範例性雜湊值係表示為一個別字元。範例性輸入資料流605將會係施加一第一標準內容定義組塊演算法之結果。接著藉由使用一預看緩衝器610來交迭查看輸入資料流605，將由依據圖5a及5b之一內容定義組塊合併函數所組成之第二修改選擇函數施加至範例性輸入資料流605。預看緩衝器610具有一八(8)個小塊之範例性容量。在此範例中，所示組塊策略限制一預期大塊由確切四個小塊組成。最初，預看緩衝器610係填充有該等小塊，或更明確而言在輸入資料流605開始處的該等塊之雜湊值"abcdefgh"。在各發射合併小塊或未合併小塊之後，從最後發射塊之後的塊開始將預看緩衝器610填充直至容量。例如，當預看緩衝器610在輸入資料流605開始處最初包含八(8)個小塊"abcdefgh"時，將該等小塊"a"、"b"、"c"及"d"合併成一新的大塊"abcd"615並如此發射，因為該些第一小塊或下一可能大塊"efgh"620在前面均未曾見過。從最後先前發射塊"abcd"615之後的下一塊"e"開始重新填充預看

緩衝器 610，也將該等小塊 "e"、"f"、"g" 及 "h" 合併成一新的大塊 "efgh" 620 並如此發射，因為最後發射塊 "abcd" 615 或下一可能大塊 "ijkl" 625 非複製大塊。從最後先前發射塊 "efgh" 620 之後的下一塊 "i" 開始重新填充預看緩衝器 610，也將該等小塊 "i"、"j"、"k" 及 "l" 合併成一新的大塊 "ijkl" 625 並如此發射，因為最後發射塊 "efgh" 620 或下一可能大塊 "mnop" 非複製大型塊。從最後先前發射塊 "ijkl" 625 之後的下一塊 "m" 開始重新填充預看緩衝器 610 之後，不將該等小塊 "m"、"n"、"o" 及 "p" 合併成一新的大塊並全部作為小塊 "m"、"n"、"o" 及 "p" 630 發射，因為儘管最後發射塊 "ijkl" 625 係一非複製大塊，但下一可能大型塊 "efgh" 640 前面已在 620 看到過。

現在參考圖 6b 中此範例型概念性圖解 600 之延續，從最後先前發射塊 "p" 630 之後的下一塊 "e" 開始重新填充預看緩衝器 610 之後，將該等小塊 "e"、"f"、"g" 及 "h" 合併成一新的大塊 "efgh" 640 並如此發射，因為該些塊作為預看緩衝器 610 開始，構成前面在 620 看到的一先前發射大塊。從最後先前發射塊 "efgh" 640 之後的下一塊 "i" 開始重新填充預看緩衝器 610，將該等小塊 "i"、"j"、"k" 及 "l" 合併成一新的大塊 "ijkl" 645 並如此發射，因為該些塊作為預看緩衝器 610 的開始，構成前面在 625 看到的一先前發射大塊。從最後先前發射塊 "ijkl" 645 之後的下一塊 "a" 開始重新填充預看緩衝器 610 之後，不將該等小塊 "a"、"a"、"a" 及 "b" 合併成一新的大塊並全部作為小塊 "a"、"a"、"a" 及 "b" 650 發射，

因為儘管前面從未看到過下一可能大塊"bbcd"，但最後發射塊"ijkl"645係已經在625前面看到過的一大塊。從最後發射塊"b"650之後的下一塊"b"開始重新填充預看緩衝器610之後，不將該等小塊"b"及"b"合併成一新的大型塊，並作為小塊"b"及"b"655發射，同時將該等小塊"a"、"b"、"c"及"d"合併成一新的大型塊"abcd"655並如此發射，因為先前發射的大塊"abcd"655充分靠近預看緩衝器610之開始以防止發射可能大塊"bbab"。

現在參考圖6c中此範例性概念性圖解600之延續，從最後先前發射塊"abcd"655之後的下一塊"k"開始重新填充預看緩衝器610之後，不將該等小塊"k"、"l"、"m"及"n"合併成一新的大型塊並全部作為小塊"k"、"l"、"m"及"n"665發射，因為儘管前面從未看到過下一可能大塊"opij"，但最後發射塊"abcd"655係在615前面看到過的一大塊。從最後先前發射塊"n"665之後的下一塊"o"開始重新填充預看緩衝器610之後，不將該等小塊"o"及"p"合併成一新的大型塊並作為小塊"o"及"p"670發射，同時將該等小塊"i"、"j"、"k"及"l"合併成一新的大型塊"ijkl"670並如此發射，因為先前發射的大塊"ijkl"670充分靠近預看緩衝器610開始以防止發射可能大塊"opij"。從最後先前發射塊"ijkl"670之後的下一塊"x"開始重新填充預看緩衝器610之後，不將該等小塊"x"、"x"、"y"及"y"合併成一新的大型塊並全部作為小塊"x"、"x"、"y"及"y"675發射，因為儘管前面從未看到過下一可能大塊"zzac"，但最後發射塊"ijkl"670係前面已經在

625看到過的一大塊。從最後先前發射塊"y"675之後的下一塊"z"開始重新填充預看緩衝器610之後，將該等小塊"z"、"z"、"a"及"c"合併成一"zzac"680並如此發射，因為預看緩衝器610已開始減少資料(使得在預看緩衝器610從可能大塊"zzac"之後的"a"開始情況下在下一交迭無法發射另一大塊)且最後發射塊"y"675並非一先前發射大塊。最後，在使用最後發射塊"zzac"680之後的剩餘小塊"a"重新填充預看緩衝器610之後，小塊"a"無法合併成一大塊，因為沒有任何塊與其一起合併，故作為一小塊"a"685發射。因此，施加該第二修改選擇函數(其由依據圖5a及5b之一內容定義組塊合併函數組成)至輸入資料流605已導致將根據該第一標準組塊演算法提供的塊修改成在較長非複製資料區域內較大，在較長非複製資料區域之邊界處較小以及在較短非複製資料區域內較小。

考量在病態情況下的再同步可能性會引起一決不允許一大塊橫跨某較高位準切割、區塊、斷裂或塊點之選項。此選項可能最終在一理論上可能長期保持不同步之複製流內強迫再同步，只因為始終以一偏移方式來始終選擇該等大塊。依據至少一具體實施例，圖7a及7b提供一內容定義組塊策略之一範例性流程圖700，該策略利用一固定大小預看緩衝器以及備份(例如再同步)塊點以在長段非複製資料內將小塊合併成大型塊，只要該大型塊不橫跨一再同步點，同時還使用小塊界定該非複製資料之該等邊緣並在較短非複製資料區域內留下小塊。首先，在703，一輸入緩

衝器或預看緩衝器包括若干更小塊。接著，在決策步驟709，決定在該緩衝器是否存在足夠小塊以發射一更大資料塊。若否，則在步驟712，可從該緩衝器發射一小塊並在步驟706，可使用在該最後發射塊之後的下一塊來填充該緩衝器。若在該緩衝器存在足夠小的塊以發射一大塊，則在決策步驟715，決定在發射一大塊時是否交叉一再同步點。若否，則該程序進行至步驟718，以決定該緩衝器是否開始於一先前發射大塊。若是，則將會交叉一再同步點，接著該程序進行至步驟712並可發射一小塊，隨後重新填充緩衝器步驟706。若是，則在步驟515，可發射該大塊，隨後重新填充緩衝器步驟706。若在步驟718，決定該緩衝器開始於一先前發射大塊，則該程序進行至步驟721以發射該大塊，隨後重新填充緩衝器步驟706。若否，則在步驟724，一決策步驟決定該緩衝器是否具有一先前發射的大塊，其不交叉再同步點並足夠靠近緩衝器開始以防止發射一大塊。若是，則在步驟727，該程序將發射在該緩衝器開始與該大型塊之間的小塊並發射該大塊，隨後重新填充緩衝器步驟706。然而，若該緩衝器不具有足夠靠近該緩衝器開始以防止發射一大塊之先前發射大塊，則該程序進行至決策步驟760。在決策步驟760，在沒有任何先前發射大塊可能足夠靠近該緩衝器開始以防止發射一大塊(緩衝器開始加上大塊長度)之點之後，決定該緩衝器是否具有足夠小的塊，以允許構造一不會交叉再同步點之隨後大塊。若否，則在步驟780，該程序決定在步驟706期間

該最後發射塊是否係一先前發射大塊。若此係是，則在步驟785，該程序可發射1個小塊，隨後重新填充緩衝器步驟706。若步驟780的答覆係否，則在步驟775發射一新的大塊。若在步驟760決定可構造一不交叉再同步點之隨後大塊，則執行決策步驟765。在決策步驟765，決定該最後發射塊是否係一先前發射塊或考量發射一大塊，隨後大塊係一先前發射大塊。若決策步驟765的答覆係否，則在步驟775發射一新的大塊。若在決策步驟765的答覆係是，則在步驟770在決定隨後大塊之前發射所有小塊。在任一情況下，隨後進行重新填充緩衝器步驟706。

下面顯示一簡單塊分割程序或演算法範例，其在偵測到一複製大塊之前或之後重新組塊一非複製大塊。

```

1  for (each big chunk) {
2      if (isBigDup) emit as big;
3      else if (isPrevBigDup II isNextBigDup)
4          rechunk as smalls; // dup/nondup transition
5      else emit as big;
6  }
```

此處，可能使用一大型平均塊大小來進行在資料上的主要趟次，且可以更小平均區塊大小來重新組塊在一複製大塊之後或之前的一單一非複製資料塊。該程序可修改此類演算法以偵測複製/非複製轉變之更複雜定義；例如當N個非複製可能相鄰D個複製時，使用更小平均大小來重新組塊R個更大塊。或者，該程序可與該等塊之實際長度一起

工作以決定並處理複製/非複製轉變點。

參考圖 8，提供一內容定義組塊策略 800 之一範例性流程圖，該策略利用一固定大小預看緩衝以在短段非複製資料內將大塊分割成小塊，同時在長段非複製資料內留下大塊並沿該等長段非複製資料之該等邊界將大塊分割成小塊。在步驟 805，一輸入緩衝器可包括複數個大塊。接著在決策步驟 815，該程序可決定該緩衝器是否開始於一先前發射大塊。若是，則在步驟 820 發射該大型塊，隨後在步驟 810 從該最後發射塊之後的下一塊開始重新填充該緩衝器。若否，則該程序進行至決策步驟 825。在步驟 825，該程序決定該最後發射塊是否係一先前發射大塊或一先前發射大塊之下一塊。若該些之任一者均真實，則該程序可進行至步驟 830 並將該大的新塊分割或重新組塊成複數個更小塊且其可在該緩衝器開始與下一大塊之間發射。此步驟之後進行緩衝器重新填充步驟 810。然後，若在步驟 825 之問題之答覆係否則在步驟 835 作為一大塊來發射該塊，隨後進行緩衝器重新填充步驟 810。

參考圖 9a、9b 及 9c，依據本發明之至少一具體實施例，提供施加由依據圖 8 之一內容定義組塊分割函數所組成之所提出第二修改選擇函數至一輸入資料流之一範例性圖解 900。現在參考圖 9a、9b 及 9c，表示概念性圖解 900、935 及 960，顯示一輸入資料流 905 由一系列範例性逗號劃界大塊雜湊值組成，其中各範例性小塊雜湊值係表示一個別字元且該等大塊係表示為序連字元。儘管曾將大多數塊描述

成各包含四個小塊，但在各大塊內的小塊數目實際上可變。範例性輸入資料流905將會係施加一設計以使用一大型平均塊大小來組塊之第一標準內容定義組塊演算法(例如圖2之220)之結果。接著藉由使用一預看緩衝器910來交迭查看輸入資料流905，將由依據圖8之一內容定義組塊分割函數(例如圖2之240)所組成之第二修改選擇函數施加至範例性輸入資料流905。預看緩衝器910具有一二(2)個大塊之範例性容量。最初，預看緩衝器910係填充有該等大塊，或更明確而言在輸入資料流905開始處的該等塊之雜湊值"abcd"與"efgh"。在各發射分割小塊或大塊之決策之後，預看緩衝器910係從最後發射塊之後的塊開始而填充直至容量。例如，當預看緩衝器910在輸入資料流905開始處最初包含二(2)個大型塊"abcd"與"efgh"時，該等小塊"a"、"b"、"c"及"d"仍保留作為一大型塊"abcd"915並如此發射，因為該些第一大型塊或下一大型塊"efgh"920在前面均未曾見過。從最後先前發射塊"abcd"915之後的下一大塊"efgh"開始重新填充預看緩衝器910之後，該等小塊"e"、"f"、"g"及"h"還保留作為一大型塊"efgh"920並如此發射，因為最後發射塊"abcd"915或下一大塊"ijkl"925未曾在前面見過。從最後先前發射塊"efgh"920之後的下一大塊"ijkl"開始重新填充預看緩衝器910之後，該等小塊"i"、"j"、"k"及"l"還保留作為一大塊"ijkl"925並如此發射，因為最後發射塊"efgh"920或下一大塊"mnop"未曾在前面見過。從最後先前發射塊"ijkl"925之後的下一塊"mnop"開始

重新填充預看緩衝器910之後，將該等小塊"m"、"n"、"o"及"p"從一大塊"mnop"分割完並全部作為小塊"m"、"n"、"o"及"p"930發射，因為儘管前面未曾見過最後發射塊"ijkl"925，但下一大塊"efgh"已經在920前面看到過。

現在參考圖9b中此範例性圖解900之延續，從最後先前發射塊"p"930之後的下一塊"efgh"開始重新填充預看緩衝器910，該等小塊"e"、"f"、"g"及"h"保留作為一大塊"efgh"940並如此發射，因為該些塊構成一已在920前面看到的先前發射大塊。從最後先前發射塊"efgh"940之後的下一大塊"ijkl"開始重新填充預看緩衝器910之後，該等小塊"i"、"j"、"k"及"l"還保留作為一大塊"ijkl"945並如此發射，因為該些塊作為預看緩衝器910開始，構成一已在925前面看到的先前發射大塊。從最後先前發射塊"ijkl"945之後的下一塊"aaab"開始重新填充預看緩衝器910之後，將該等小塊"a"、"a"、"a"及"b"從一大塊"aaab"分割完成小塊"a"、"a"、"a"及"b"950發射，因為儘管前面從未看到過下一可能大塊"bbab"，但最後發射塊"ijkl"945較大且已經在925前面看到過。從最後先前發射塊"b"950之後的下一大塊"bbab"開始重新填充預看緩衝器910之後，該等小塊"b"、"b"、"a"及"b"仍保留作為一大塊"bbab"955並如此發射，因為最後發射塊"b"950或下一大塊"cdkl"965係已在前面見過的大塊。讀者應瞭解，一塊分割演算法不求助於偵測並重新發射透過大塊"bbab"955中途開始的大型複製塊"abcd"。此在一進行一增加數量查詢用於複製大塊的塊合

併演算法中有可能。

現在參考圖9c中此範例性圖解900之延續，從最後先前發射塊"bbab"955之後的下一塊"cdkl"開始重新填充預看緩衝器910，該等小塊"c"、"d"、"k"及"l"保留作為一大塊"cdkl"965並如此發射，因為最後發射塊"bbab"955前面從未見過且下一大塊"mnop"前面從未作為一大塊發射過。從最後先前發射塊"cdkl"965之後的下一塊"mnop"開始重新填充預看緩衝器910之後，將該等小塊"m"、"n"、"o"及"p"從一大塊"mnop"分割並全部作為小塊"m"、"n"、"o"及"p"970發射，因為儘管前面未曾見過先前發射塊"cdkl"965，但下一大塊"ijkl"前面已在945看到過。從最後先前發射塊"p"970之後的下一大塊"ijkl"開始重新填充預看緩衝器910之後，該等小塊"i"、"j"、"k"及"l"保留作為一大塊"ijkl"975並如此發射，因為該些塊作為預看緩衝器910開始，構成一前面在945已看到的先前發射大塊"ijkl"。從最後先前發射塊"ijkl"975之後的下一塊"xxyy"開始重新填充預看緩衝器910之後，將該等小塊"x"、"x"、"y"及"y"從一大塊"xxyy"分割完並全部作為小塊"x"、"x"、"y"及"y"980發射，因為儘管前面未曾見過下一塊"zzac"，但先前發射塊"ijkl"前面已在975看到過。從最後先前發射塊"y"980之後的下一塊"zzac"開始重新填充預看緩衝器910之後，該等小塊"z"、"z"、"a"及"c"仍作為一大塊"zzac"985並如此發射，因為隨後大塊"a"前面從未見過，且先前發射塊"y"980係複製但不係一複製大塊。最後，在使用最後發射塊

"zzac"985之後的剩餘小塊"a"重新填充預看緩衝器910之後，僅如此發射小塊"a"。施加該第二修改選擇函數(其係由依據圖8之一內容定義組塊分割函數組成)至輸入資料流905已導致將根據該第一標準組塊演算法提供的塊修改成在較長非複製資料區域內較大，在較長非複製資料區域之邊界處較小以及在較短非複製資料區域內較小。

圖10係依據本發明之至少一具體實施例一內容定義組塊合併策略於一歸檔或備份系統1000之一範例性應用。在本發明之此範例性應用中，可提供在(例如)一電腦系統中的一源檔案系統1010並可包含資料集或資料流。此源檔案系統1010可耦合至一備份程序1020。本發明可併入源檔案系統1010及/或備份程序，並可操作以提供更佳資料組塊並更佳複製消除。備份程序1020可接著將該修改資料輸入一目標儲存裝置1030，例如一磁帶、磁碟、遠端磁碟儲存系統等。一範例性儲存系統可能係(例如)一土星完全備份系統。現在將提供用於本發明之各種範例性具體實施例之一些試驗資料。

現在吾人將轉向論述本發明之試驗結果及分析。本文該(等)系統及方法與各種模擬裝置(例如工具)允許吾人研究一類技術、程序或演算法，其中吾人使用關於未發射小塊(即因為其係某大塊之部分而未發射的小塊)的資訊。由於將會存在大量不知何故需要儲存的小塊"元資料"，一基於此類資訊之策略較不合需要。

在測試碼中，還可允許吾人一些理論興趣的非實用演算

法。吾人單獨維持用於許多不同類型的塊發射的布魯姆過濾器：小塊及大塊，同時發射與未發射。一益處(例如)係允許"複製"資料區域概念包括先前發射小塊以及未發射小塊(其過去作為一些先前大塊發射之部分而發射)二者。一修改以查詢未發射小塊之演算法可在一更精細顆粒位準上偵測複製資料。該碼還可注入主動錯誤至該等塊存在應答中，以便測量大致回答存在查詢之效果。

吾人使用在一4個月期間完全備份數百使用者目錄的由大約1.16太位元組的所組成之一測試資料集。儘管吾人無法確定檔案位元組的分佈情況，但僅確定其係一大型實際資料集且相當代表實際中可能所見者。

曾發展過許多工具，其方便線上模擬各種技術及/或演算法，使得分析極大型輸入資料集成為可能。關鍵思路在於產生輸入資料的一二進制"概述"。對於在位準9(理論上每塊平均512個位元組)所產生的每一小塊，吾人儲存該塊的SHA-1雜湊以及塊大小及實際切割點位準[s]。此塊資料足以重新組塊大多數輸入資料集，除了根本不產生任何塊點的每一非隨機資料集外。吾人公用程序之稍後版本還儲存局部壓縮估計，其係藉由透過一LZO壓縮演算法來運行每一(4k、8k、16k、32k)固定大小塊並使用原始塊大小百分比來儲存一單一位元組。接著給定目前流偏移及塊大小，吾人可估計在流內任意點處的壓縮。依此方式，該1.1太位元組輸入資料可作為一更可攜60 Gb的概述資訊集來加以分析。(此類重新分析花費數小時而非數天，很大

程度上係由於減小磁碟讀取。)吾人還可在針對一分離檔案碰到每一位準9塊時儲存其複製/非複製狀態。

在一給定演算法內，存在數個參數(例如最小及最大塊大小)及可產生不同表現的觸發位準。打散及合併演算法還具有其他參數，例如k(在一大塊內的小塊數目)、是否允許可變大小的塊等。當一演算法在整個1.1太位元組資料集或其壓縮概述上運行時，吾人測量DER，通常作為輸入位元組與儲存壓縮塊內位元組之比率。測量不同演算法在吾人資料集上的效能。

依據至少一具體實施例，圖11顯示對於各種組塊技術，壓縮因數對塊大小之實驗結果。圖11表示對於三個演算法一DER取樣對塊大小測量。對於各演算法，實線係該等儲存塊之平均大小，而虛配對者係相同塊之平均壓縮大小。平均塊大小越大，壓縮便越出色。基線演算法1120及1125改變最小及最大塊大小以及平均塊大小。一般而言，吾人保持組塊位準設定以產生大致在最小塊大小與最大塊大小中間的平均塊長度，且吾人使用三個備份切割點位準。在"簡單"跡線1130及1135中內顯示一簡單合併組塊程式，其使用固定大小的塊來運行。"想像"演算法1140及1145具有最佳效能，但此使用布魯姆過濾器查詢用於先前曾遭遇但僅作為一先前大塊之部分發射的塊。

參考圖12，依據至少一具體實施例，顯示對於各種組塊技術，壓縮因數對塊大小之實驗結果。圖12顯示相同三個演算法在一比圖11更大的平均塊大小範圍上的更多測量：

基線 1220 及 1225、實際 1245 及 1250 及最佳 1235 及 1240。吾人還顯示主動錯誤率設定在 0、1、2、5、10、20 及 40% 時塊存在查詢的主動錯誤應答效果。水平線係對於相同輸入資料 (DER=5.67) 一商用 (DataDomain) 備份器具之複製消除效能。對於此測量，平均塊大小未知，但可能在 4 至 10k 位元組之間的某處。標記 "實際" 及 "最佳" 的跡線係與圖 11 內相同的演算法，但在一更寬參數範圍上。在一對數尺度上，趨勢在一從 1k 至 80k 之塊大小範圍上顯現線性。然而，此並非一般規則，由於限制性表現肯定存在。隨著該等塊大小變得極大 (圖示最下面部分)，故修改該基線演算法幾乎沒有任何效益。為了具競爭力，吾人希望與具有一大約 8k LBFS 樣式塊大小之系統競爭，該系統產生一大約 5.7 之 DER。吾人能夠獲得一平均儲存塊大小在 20k 與 30k 之間的類似 DER。正嘗試研究來自一局部布魯姆過濾器之大致查詢是否可形成一實用且可縮放歸檔機制之基礎。為了使此與巨型 (例如 Pb) 資料集成為可能，吾人研究一些演算法對錯誤識別複製之靈敏度。在所有情況下，最終證明 DER 及塊大小極快速地劣化，甚至對於塊存在查詢具有 1% 主動錯誤。所發生的係該些錯誤產生的複製/非複製轉變不正確地產生小塊。最佳的係，應在一些受限的過去將此類 "局部" 布魯姆過濾器限制於高品質的存在查詢，使得不會引起任何主動錯誤。然而，若在該局部布魯姆過濾器內未發現一物件，則仍應從後端儲存機制獲得一確定性回答。對於網路傳輸，一局部布魯姆過濾器方案可完全足

夠。

吾人研究一些更複雜演算法，其試圖在出現大塊時在優先化大塊相等選擇所能做的方面更加機智，但該些演算法效果不甚出色。例如，一變化方案使用一些相當複雜的邏輯來優先化在再同步點開始或結束的大塊並使用測試以使用相等優先權的最長大塊。顯示沒有任何顯著改良。實際上，若干此類嘗試在實際資料上運行時經常出於相當微妙的原因而效果較差。

參考圖 13，依據至少一具體實施例，顯示對於各種組塊技術，打散複製消除比對塊大小之實驗結果。在圖 13 中，比較基線演算法 1320，吾人呈現使用一打散演算法之結果。系列 1 1325、系列 2 1330、系列 3 1335 及連接最小查詢 1340 僅具有超過基線 1320 之名義改良。變化用於該些列的參數係重新組塊將會產生的期望小塊數目 k 。然而，為了獲得較佳效能，必須小心設定該等打散演算法之該等參數。該等打散演算法似乎在平均塊大小 40 至 50 k 下特別出色。不幸的係，此體系具有某些無競爭力的 DER。吾人還顯示對於 "簡單" 合併演算法資料之一些資料的未壓縮 DER。吾人看到該打散演算法可能幾乎與該等最簡單合併演算法實施方案一樣出色。

吾人比較在塊點選擇(參見表 1)基於拉賓、乘以一大素數及一組合箱車 + CRC32c 方案、MLCG 及組合 -CRC32c 雜湊時使用該基線演算法之吾人資料集之 DER。吾人發現幾乎相同的複製消除值與平均塊大小。此確證 Thaker 觀察，即

對於一般資料，甚至使用一平常箱車和來選擇組塊點會產生一合理隨機狀塊大小分佈。他解釋此為一在輸入資料自身內存在在足夠位元位準隨機性之指示，故對於他的資料不需要一高品質隨機化雜湊。

現在吾人將論述(例如)最佳的"始終在一起"塊之資料特性。對於吾人的 1.1 Tb 資料集，還感興趣的係考量小塊的可能最佳配置。最終證明，在塊概述之 $A=512$ 位準處，僅涉及大約 360 百萬個不同塊。小心並轉譯不同訊息摘錄成整數識別符，有可能寫入其他工具以決定一些簡單塊合併，其使用"未來知識"來決定一些組塊效能的理論限制。

一組簡單的最佳化行動係始終合併始終在一起出現的兩個塊。此根本不會影響 DER，但會增加平均塊大小。交迭此會產生始終共同出現的最長可能串的塊。此類合併不影響 DER，但會增加平均塊大小。

實務中，合併經常在一起或始終在一起的塊可能係一用以最佳化儲存之有用背景任務。然而，一歸檔後端可能不支援刪除，並可能不提供隨後塊資訊，故吾人無法遵循此途徑。相反此實驗提供一方法以判斷基於複製/非複製轉換區域之吾人簡單演算法正在執行的程度。發現，在原"位準 9"概述流上的此程序將平均未壓縮儲存塊大小從 576 增加至 5855 個位元組(即對於此資料集，平均始終共同出現小塊數目大約為 10)。此 6k 大小類似於一般在低帶寬網路檔案系統(LBFS)樣式組塊中所使用之塊大小。(例如參見 A. Muthitacharoen、B. Chen 及 D. Mazieres，"一種低帶

寬網路檔案系統"，(2001)，第174至187頁)

吾人還分析使用平均大約8k塊大小之一更大原始組塊。合併每一對始終在一起塊導致一75397位元組的平均儲存塊大小。再次，此程序將平均儲存塊大小增加一大約10的因數。

現在參考圖14，圖表1400提出觀察基線組塊演算法1430之先前提及"理論"限制、若干其他可變大小合併列1440及1450(一不同於先前圖示種所呈現之該等集合之集合，但具有大約相同的表現)及各種理論塊大小限制1460及1470。在此情況下，目的係增加DER與塊大小二者。如所示，圖表1400說明藉由合併每一組在吾人1.1太位元組資料集內始終共同出現之塊所決定的理論塊大小。此程序可能不影響複製消除比(DER)(忽略元資料成本)，但可能增加平均儲存塊大小。圖表1400對於吾人1.1太位元組資料集，將具有可變大小大塊的吾人合併演算法(例如1440)大約放置於此特定理論組塊限制中途。

儘管基於簡單原則P1及P2之考量來最初公式表述，但較重要的係判斷大多數吾人測試資料輸入脫離此類簡化資料模型之程度。故，考量在吾人測試資料集內的實際特性及變更之一分析。吾人發現，實際輸入測試資料相當程度上實質偏離一附著至P1及P2之"理想"資料集。一附著至P1之最簡單可能資料集可能在一第一備份會話期間具有長序列的鄰接非複製資料，跟隨在後序列期間的長序列複製資料。

吾人藉由使用塊的"目前"複製狀態之一位元流概述，詢問在512位元組位準所組塊的概述流來開始評估吾人測試資料集。鄰接非複製塊數目對鄰接複製隨後塊數目(反之亦然)之實際直方圖顯示一具有一單一非複製塊，隨後一單一複製塊之壓倒性且平滑變化偏好。給定用於產生涉及激發本發明之試探式的簡單原則，此可能係令人吃驚的。圖表繪製用於最終鄰接複製/非複製塊數目(在14個全備份會話)之直方圖(未顯示)。而且，令人吃驚的係，在首個"完全"備份之後的該等直方圖特性上類似於該等直方圖。然而，值得注意的係，此類直方圖不足以估計不同策略之複製估計，由於未計數一塊的複製次數。結合吾人測量，即平均始終在一起塊大小大約為6k，此評估暗示著，該測試資料集之統計表現之一更深理解可產生其他試探式，其可能甚至能夠勝過本文所發明之該等程序，特別係對於涉及複製與非複製塊之高度散布之輸入資料集。雖然如此，使用此1.1 Tb測試資料所發現之該等改良暗示著本發明還可在具有非理想統計行為之無數其他實際生活資料集上獲得健固效能。

如上所述，在各種具體實施例中，本發明可用於資料識別與複製資料消除。如此，在決定較佳雜湊、切割、邊界、塊或斷裂點之後，可產生一用於決定資料塊之雜湊值並與先前儲存雜湊值進行比較。在各種具體實施例中，本發明可特別適用於資料複製消除。此外，也如上所述，本發明可平等應用於各種電子系統，包括無線網路、網際網

路、企業內部網路、程序、儲存及/或傳輸一串資料及可證明使用雜湊有用之電腦系統及網路。下面提供在可操作本發明之各種系統之一些系統之一說明。

如上所述，在各種具體實施例中，本文所提供之該(等)系統及方法可使用一計算裝置來實施，例如一個人電腦、一伺服器、一微型主機電腦及/或一主機電腦等，其係程序化以執行一指令序列，該等指令序列組態該電腦以執行如本文所述之操作。在各種具體實施例中，該計算裝置可能係(例如)一可購自任一數目商業製造商(例如德州奧斯汀的 Dell Computer)之個人電腦，其運行(例如)Windows™ XP™與Linux作業系統並一組標準周邊裝置(例如鍵盤、滑鼠、顯示器、印表機)。

圖 15 係一計算裝置 1500 之一具體實施例之一功能性方塊圖，其可用於收錄實施本文所述之該(等)系統及方法之軟體應用程序。現在參考圖 15，計算裝置 1500 可包括一處理單元 1505、通信介面 1510、儲存裝置 1515、一使用者介面 1520、作業系統指令 1535、應用可執行指令/API 1540，全部均提供於功能性通信並可使用(例如)一指令匯流排 1550。計算裝置 1500 還可包括系統記憶體 1555、資料及資料可執行碼 1565、軟體模組 1560 及(多個)介面埠。該(等)介面埠 1570 可耦合至一或多個輸入/輸出介面 1575，例如印表機、掃描機、一體式印表機/掃描機/傳真機等。該(等)處理單元 1505 可能係一或多個微處理器或微控制器，其係經組態用以執行實施本文所述之該等功能之軟體指

令。應用可執行指令/API 1540及作業系統指令1535可使用計算裝置1500而儲存在(多個)儲存裝置1515及/或可包括非揮發性及揮發性記憶體之系統記憶體1555上。應用可執行專利/API 1540可包括實施本發明系統及方法之軟體應用程序。作業系統專利1535可包括軟體指令，其可操作以控制基本操作及控制處理器1505。在一具體實施例中，作業系統指令1535可包括(例如)購自華盛頓Redmond Microsoft Corporation的XP™作業系統。

可將指令讀取至一來自另一電腦可讀取媒體(例如一儲存裝置)之主記憶體。本文所使用之術語"電腦可讀取媒體"可能係指任一參與提供指令至處理單元1505以供執行之媒體。此類媒體可採取許多版本，包括(但不限於)非揮發性媒體、揮發性媒體及傳輸媒體。非揮發性媒體可包括(例如)光碟或磁碟，拇指或跳躍磁碟機及儲存器件。揮發性媒體可包括動態記憶體，例如一主記憶體或快取記憶體。傳輸媒體可包括同軸電纜、銅導線及光纖，包括包含匯流排1550之連接。傳輸媒體還可採取聲波或光波版本，例如在射頻(RF)及紅外(IR)資料通信期間所產生之該等波。電腦可讀取媒體之常見版本包括(例如)軟碟、一撓性碟、硬碟、磁帶、任何其他磁性媒體、通用序列匯流排(USB)memory stick™、一CDROM、DVD、任何其他光學媒體，一RAM、一ROM、一PROM、一EPROM、一快閃EPROM、任何其他記憶體晶片或匣、一載波(如下文所述)或任何其他一電腦從其可讀取之媒體。

電腦可讀取媒體之各種版本可能涉及遞送一或多個指令之一或多個序列至該(等)處理單元1505以供執行。例如，該等指令可最初搭載在一(多個)遠端電腦1585(例如一伺服器、一PC、一主機等)之一磁碟上。該(等)遠端電腦1585可將該等指令載入其動態記憶體內並使用(例如)一連接至一資料機之電話線在一或多個記憶體介面1580發送該等指令，該資料機可能係一類比、數位、DSL或纜線數據機。該網路可能係(例如)網際網路及企業內部網路、一對等網路等。計算裝置1500可經由通信介面1510，透過其他電腦之一網路來發送訊息並接收資料，包括(多個)程序碼，該通信介面可透過(多個)網路介面1580而耦合。一伺服器可透過該網際網路傳輸一要求一應用程序之請求碼以獲得一下載應用程序。該接收碼可在接收其及/或儲存於一儲存裝置1515或其他非揮發性儲存器1555以供稍後執行時由該(等)處理單元1505來執行。依此方式，計算裝置1500可以一載波版本獲得一應用程序碼。

本系統及方法可駐留於一單一計算裝置或平臺1500或多個計算裝置1500上，或不同應用程序可駐留於分離計算裝置1500上。應用可執行指令/API 1540及作業系統指令1535可載入至計算裝置1500揮發性記憶體之一或多個配置碼片段以供運行時間執行。在一具體實施例中，計算裝置1500可包括系統記憶體1555，例如揮發性記憶體之512 MB與非揮發性記憶體儲存器之80 GB。在至少一具體實施例中，本發明系統及方法之軟體部分可使用(例如)C程序化語言

源碼指令來執行。其他具體實施例亦可行。

應用可執行指令/API 1540可包括一或多個應用程序介面(API)。本發明之該(等)系統及方法可使用API 1540用於程序間通信並請求及返回應用程序間函數呼叫。例如，可結合一資料庫1565來提供一API，以便促進發展(例如)SQL指令碼，其係用以引起該資料庫依據該(等)指令碼內所指定之該等指令來執行特性資料儲存或接取操作。一般而言，API可用於促進發展應用程序，其係程序化以完成本文所述之該等功能之一些功能。

該(等)通信介面1510可向計算裝置1500提供在網際網路上傳輸並接收資訊之能力(包括但不限於電子郵件、HTML或XML頁)以及檔案傳送能力。為此目的，通信介面1510可進一步包括一網頁瀏覽器，例如但不限於Microsoft Corporation所提供的Microsoft Internet ExplorerTM。該(等)使用者介面1520可包括一電腦終端顯示器、鍵盤及滑鼠裝置。一或多個圖形使用者介面(GUI)還可包括以提供顯示及操作包含於互動式HTML或XML頁內之資料。

現在參考圖16，說明一網路1600，在其上可操作該(等)系統及方法。如上述，本專利申請案之該(等)系統及方法可在一或多個電腦上操作。網路1600可包括耦合至一或多個用戶資料儲存器1610的一或多個用戶1605。該一或多個用戶可透過一通信網路(例如光纖、電話線、無線等)耦合至通信框架1630。通信框架1630可能係(例如)網際網路及企業內部網路、一對等網路、一LAN、一特定電腦至電腦

網路等。網路1600還可包括一或多個伺服器1615，其係耦合至通信框架1630並耦合至一(多個)伺服器資料儲存器1620。該(等)本發明系統及方法還可具有多個部分，其可在網路1600內的該等元件之一或多個元件上操作，以便作為一完整作業系統及方法來操作。

儘管上面已說明本發明之具體實施例，但顯然，習知此項技術者應明白許多替代、修改及變更。一般而言，多個具體實施例可能關於其中執行資料分析之該些及其他商務程序之自動化。據此，如上面所提出，本發明之該等具體實施例意在說明性，不應視為限制本發明之範疇。可進行各種變化而不脫離本發明之精神及範疇。據此，本發明之範疇應不僅由上示具體實施例，還由其隨附申請專利範圍及其合法等效內容來加以定義。

本文所引用之所有公告案、專利案及專利申請案全部內容均出於全部目的以引用版本併入本文。

【圖式簡單說明】

結合附圖，考量本發明之具體實施例之下列上述詳細說明應容易地瞭解並明白本發明實用性、目標、特徵及優點，且

圖1係依據本發明之至少一具體實施例之一資料管理系統之一範例性概略，其可在一輸入流內識別資訊並可有時在產生輸出之前修改該資訊；

圖2係依據至少一具體實施例之一資料識別程序之一範例性概略，該資料識別程序利用一第一選擇函數並接著利

用一第二修改選擇函數，該第二修改選擇函數由一可合併及/或打散特定現有塊之內容定義組塊策略組成；

圖3a及3b係顯示依據至少一具體實施例之一範例性內容定義組塊技術或策略之資料流，該範例性內容定義組塊技術或策略可在長段非複製資料內一起添加或合併小塊成大型塊及/或可能不合併在非複製資料邊緣處的小塊以更佳識別在此類轉變區域內的可能複製；

圖4a及4b係顯示依據至少一具體實施例之一範例性內容定義組塊技術或策略之資料流，該範例性內容定義組塊技術或策略可在非複製資料邊緣處將更大型塊打散或分割成小塊以在此區域內更佳識別潛在複製；

圖5a及5b係依據至少一具體實施例之一範例性內容定義組塊技術或策略之一流程圖，該範例性內容定義組塊技術或策略可將小塊合併成大型塊及/或使用小塊來界定非複製資料之邊緣及/或在較短非複製資料區域內留下小塊；

圖6a、6b及6c係依據至少一具體實施例用於一資料集或一資料輸入流之一內容定義組塊合併技術或策略之時序圖；

圖7a及7b係依據本發明之至少一具體實施例之一範例性內容定義組塊技術或策略之一流程圖，該範例性內容定義組塊技術或策略可包括再同步將小塊合併成大型塊；

圖8係依據至少一具體實施例之一範例性內容定義組塊技術或策略之一流程圖，該範例性內容定義組塊技術或策略可將大塊分割成小塊；

圖 9a、9b及 9c係依據至少一具體實施例用於一資料集或一資料輸入流之一內容定義組塊分割技術或策略之時序圖；

圖 10係依據至少一具體實施例之一可施加資料合併及/或資料分割技術或策略之備份系統之範例性應用；

圖 11顯示依據至少一具體實施例對於各種組塊技術壓縮因數對塊大小之實驗結果；

圖 12顯示依據至少一具體實施例對於各種組塊技術壓縮因數對塊大小之實驗結果；

圖 13顯示依據至少一具體實施例對於各種組塊技術打散複製消除比對塊大小之實驗結果；

圖 14顯示依據至少一具體實施例對於理論塊大小限制 DER對塊大小之實驗結果；

圖 15係依據至少一具體實施例之一計算裝置之一範例性功能方塊圖；以及

圖 16係依據至少一具體實施例說明一網路之一範例性功能方塊圖。

【主要元件符號說明】

100	輸入資料流
120	資料識別系統
130	資料操縱系統
140	輸出資料流
301-326	鄰接塊
330	摘要

340	輸 出 流
341	大 型 塊
342	大 型 塊
343	大 型 塊
344	大 型 塊
345	大 型 塊
346	大 型 塊
350	資 料 流
351	資 料 流
352	塊
353	資 料 區 塊 或 塊
354	資 料 區 塊 或 塊
355	資 料 區 塊 或 塊
356	資 料 區 塊 或 塊
357	塊
360	塊 點
370	塊 點
381	小 塊
382	小 塊
383	小 塊
384	小 塊
385	大 型 塊
390	大 型 塊
391	大 型 塊

395	大型塊
396	大型塊
397	大型塊
400	資料流
410	組塊
420	範例性輸出
450	資料流
460	組塊
461	複製
462	複製/初始塊
463	複製
464	複製
465	複製
466	複製
470	切割、斷裂、區塊或塊點
475	切割、斷裂、區塊或塊點
480	範例性輸出
481	小區塊或塊
482	小區塊或塊
483	小區塊或塊
484	小區塊或塊
485	小區塊或塊
491	區域
492	區域

493	區域
494	區域
500	資料流
605	輸入資料流
610	預看緩衝器
615	大塊 "abcd"
620	大塊 "efgh"
625	大塊 "ijkl"
630	最後先前發射塊 "p"
640	大塊 "efgh"
645	大塊 "ijkl"
650	小塊 "a"、"a"、"a"及 "b"
655	大塊 "abcd"
665	小塊 "k"、"l"、"m"及 "n"
670	小塊 "o"及 "p"
675	小塊 "x"、"x"、"y"及 "y"
680	大塊 "zzac"
685	小塊 "a"
905	輸入資料流
910	預看緩衝器
915	大塊 "abcd"
920	大塊 "efgh"
925	大塊 "ijkl"
930	小塊 "m"、"n"、"o"及 "p"

940	大塊 "efgh"
945	大塊 "ijkl"
950	小塊 "a"、"a"、"a"及 "b"
955	大塊 "bbab"
965	大塊 "cdkl"
970	最後先前發射塊 "p"
975	大塊 "ijkl"
980	小塊 "x"、"x"、"y"及 "y"
985	大塊 "zzac"
1000	歸檔或備份系統
1010	源檔案系統
1020	備份程序
1030	目標儲存裝置
1120	基線演算法
1125	基線演算法
1130	"簡單"跡線
1135	"簡單"跡線
1140	"想像"演算法
1145	"想像"演算法
1220	基線
1225	基線
1235	最佳
1240	最佳
1245	實際

1250	實際
1320	基線演算法
1325	系列 1
1330	系列 2
1335	系列 3
1340	連接最小查詢
1400	圖表
1430	基線組塊演算法
1440	可變大小合併列
1445	可變大小合併列
1460	理論塊大小限制
1470	理論塊大小限制
1500	計算裝置
1505	處理單元
1510	通信介面
1515	儲存裝置
1520	使用者介面
1535	作業系統指令
1540	應用可執行指令/API
1550	指令匯流排
1555	系統記憶體
1560	軟體模組
1565	資料及資料可執行碼
1570	介面埠

1575	輸入/輸出介面
1580	網路介面
1600	網路
1605	用戶
1610	用戶資料儲存器
1615	伺服器
1620	伺服器資料儲存器
1630	通信框架

十、申請專利範圍：

1. 一種資料管理方法，該方法實現於利用一儲存器件之一計算器件中且由該計算器件執行，該方法包含：

使用一第一資料片段化程序與一第二資料片段化程序之一組合，將一資料流分段，其中該第一資料片段化程序與該第二資料片段化程序之一期望平均資料群組大小不同，

其中該第一資料片段化程序將該資料流打散成複數個資料群組，

其中該第二資料片段化程序自該複數個資料群組將至少二個較小資料群組組合在一起，以形成具有比該等較小資料群組的平均大小還大之一平均大小之一更大資料群組，且

其中回應於判定至少二個另外較小資料群組係在該資料流中是新資料且非先前在該資料流中出現過的複製資料，且回應於判定依序相鄰該至少二個另外較小資料群組之一資料群組係先前出現在該資料流之一複製資料，該第二資料片段化程序避免將該至少二個另外較小資料群組組合在一起，其中在該複數個資料群組中該至少二個另外較小資料群組係連續資料群組，且其中回應於判定該至少二個另外較小資料群組係在該資料流中是新資料且非先前在該資料流中已出現過的複製資料，該第二資料片段化程序將該至少二個較小資料群組組合為該更大資料群組。

2. 如請求項1之方法，其中當已到達或處理該資料流之一最大預定長度時，將新資料的一或多個較小資料群組組合成一更大資料群組。
3. 如請求項1之方法，其中依序相鄰該至少二個另外較小資料群組之該資料群組係在該資料流中之該至少二個另外較小資料群組之前依序相鄰。
4. 如請求項1之方法，其中依序相鄰該至少二個另外較小資料群組之該資料群組係在該資料流中之該至少二個另外較小資料群組之後依序相鄰。
5. 如請求項1之方法，其中依序相鄰該至少二個另外較小資料群組之該資料群組係在該資料流中之該至少二個另外較小資料群組之前或之後依序相鄰。
6. 如請求項1之方法，其進一步包含以下步驟：

發射一或多個較小資料群組或一或多個更大資料群組。
7. 如請求項6之方法，其中在不交叉一再同步點時，僅發射一更大資料群組。
8. 如請求項1之方法，其中當不將任何以該更小資料群組結束之更大資料群組視為一重複時、當一緊接隨後數目的資料群組係一預定類型或將一緊接隨後數量的資料係視為複製資料時，不將該第二資料片段化程序施加至至少一更小資料群組。
9. 一種資料管理方法，該方法實現於利用一儲存器件之一計算器件中且由該計算器件執行，該方法包含：

使用一第一資料片段化程序與一第二資料片段化程序之一組合，將一資料流分段，其中該第一資料片段化程序與該第二資料片段化程序之一期望平均資料群組大小不同，

其中該第一資料片段化程序將該資料流打散成複數個資料群組，

其中回應於判定一已知資料群組非先前在該資料流中出現過的複製資料，該第二資料片段化程序避免將該複數個資料群組之該已知資料群組打散，且

其中回應於判定該複數個資料群組的至少一者非先前在該資料流中出現過的複製資料，且回應於判定依序相鄰該複數個資料群組的該至少一者之一資料群組係先前出現在該資料流之一複製資料，該第二資料片段化程序進一步將依序相鄰該已知資料群組之該複數個資料群組的該至少一者打散為較小資料群組。

10. 如請求項9之方法，其中該第一片段化程序具有一大於該第二片段化程序之平均資料群組大小。

11. 如請求項9之方法，其進一步包含以下步驟：

發射一或多個較小資料群組或一或多個更大資料群組。

十一、圖式：

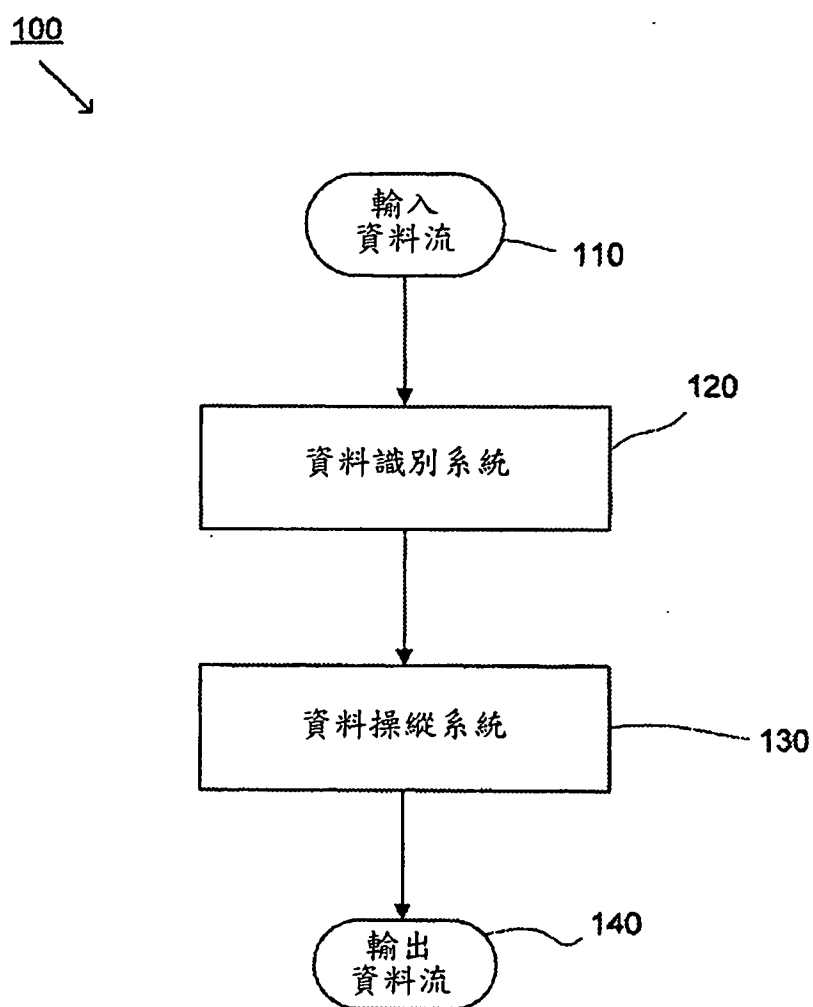


圖 1

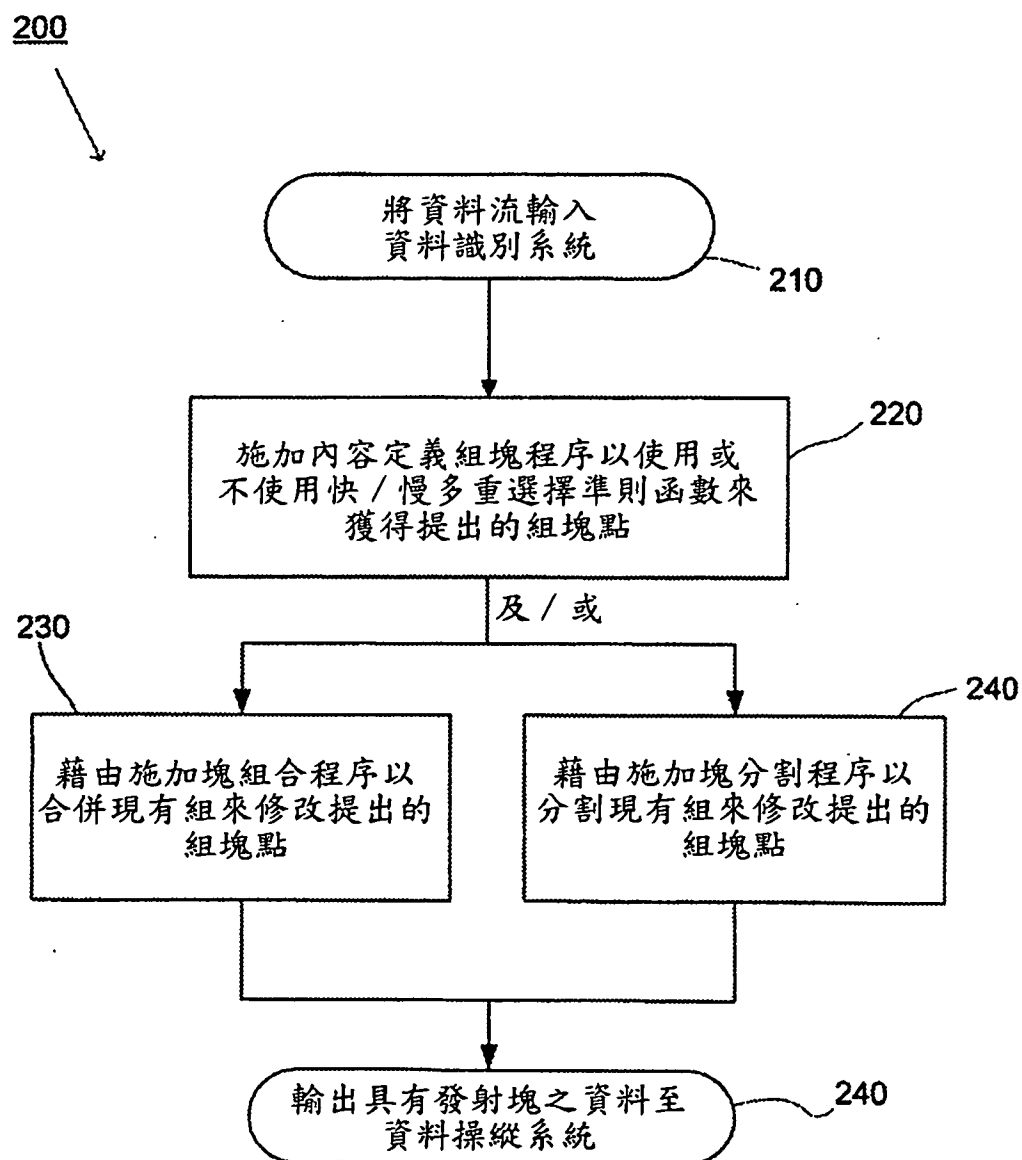


圖 2

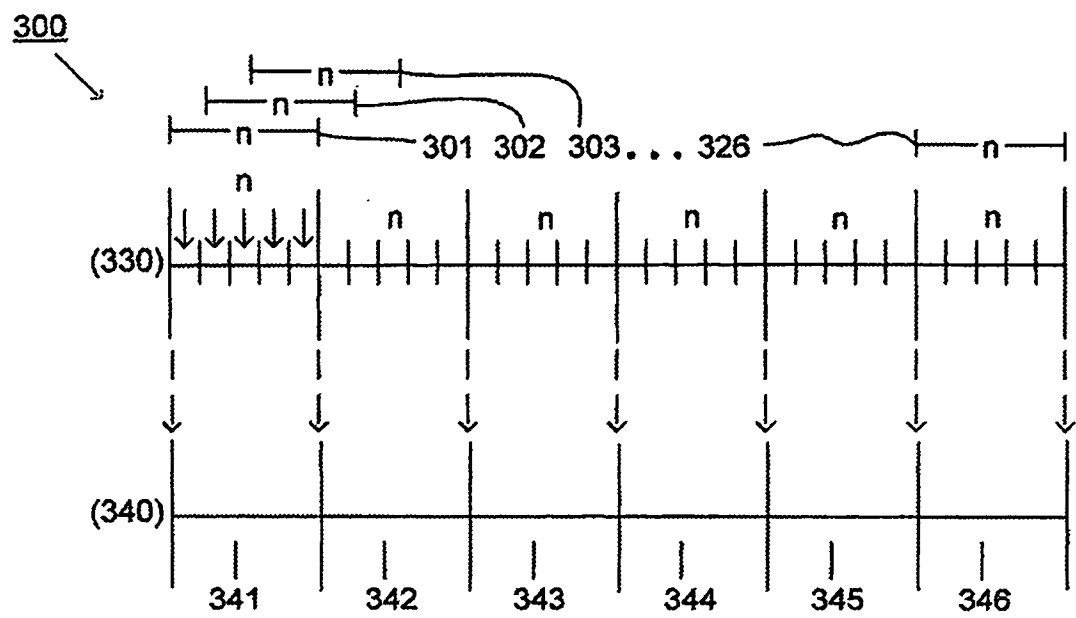


圖 3a

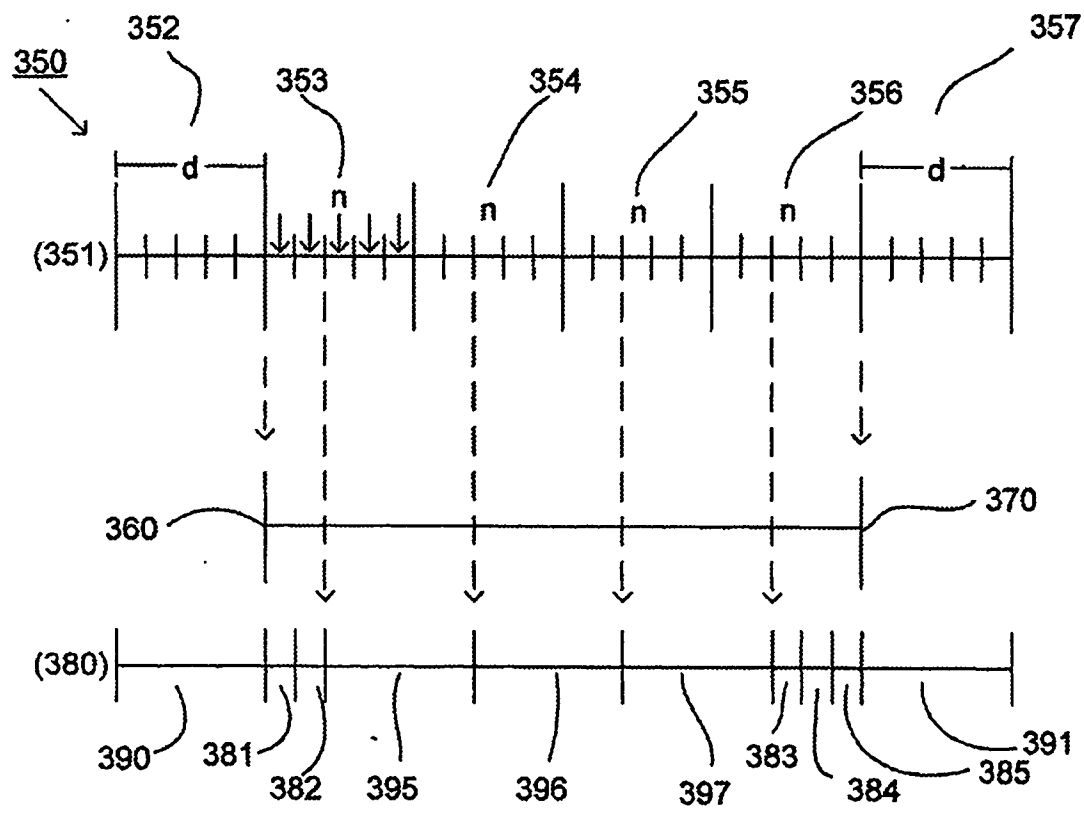


圖 3b

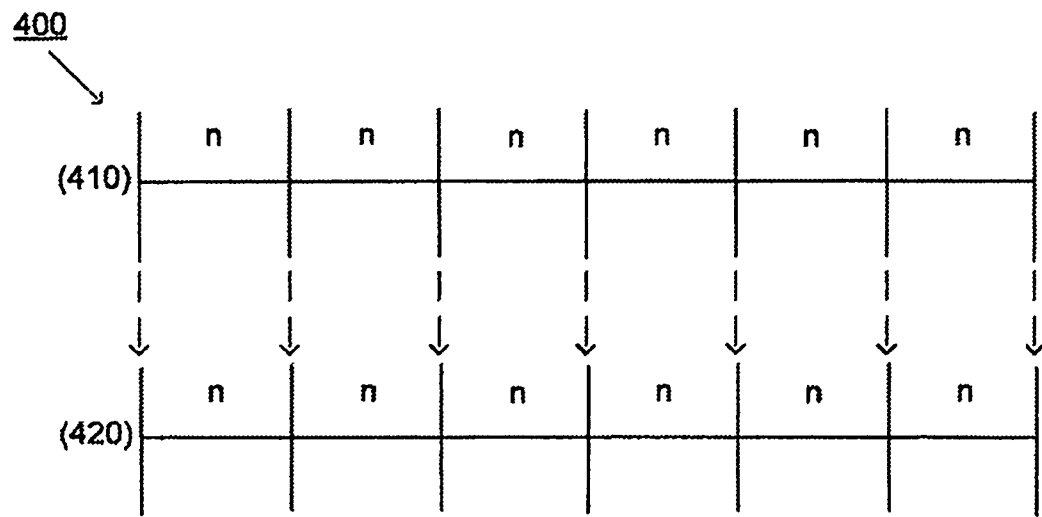


圖 4a

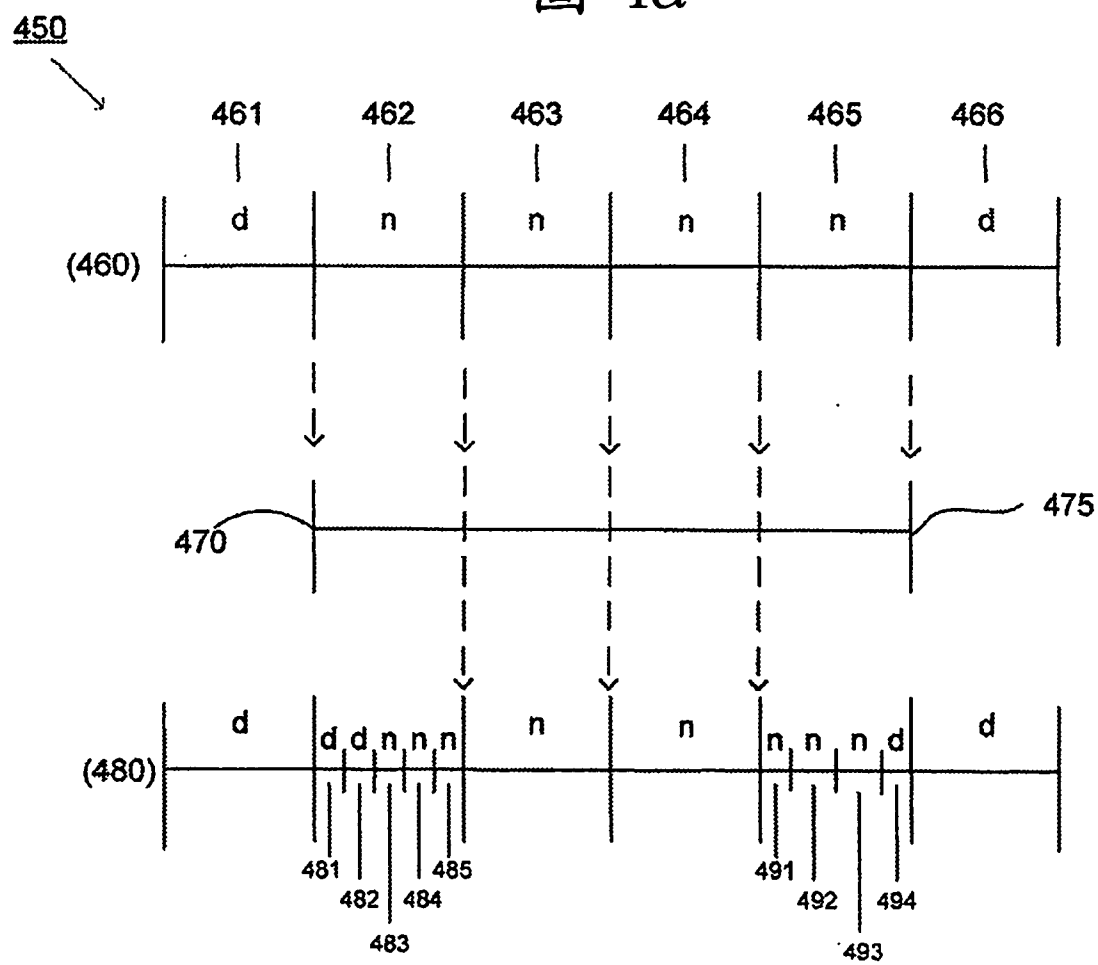


圖 4b

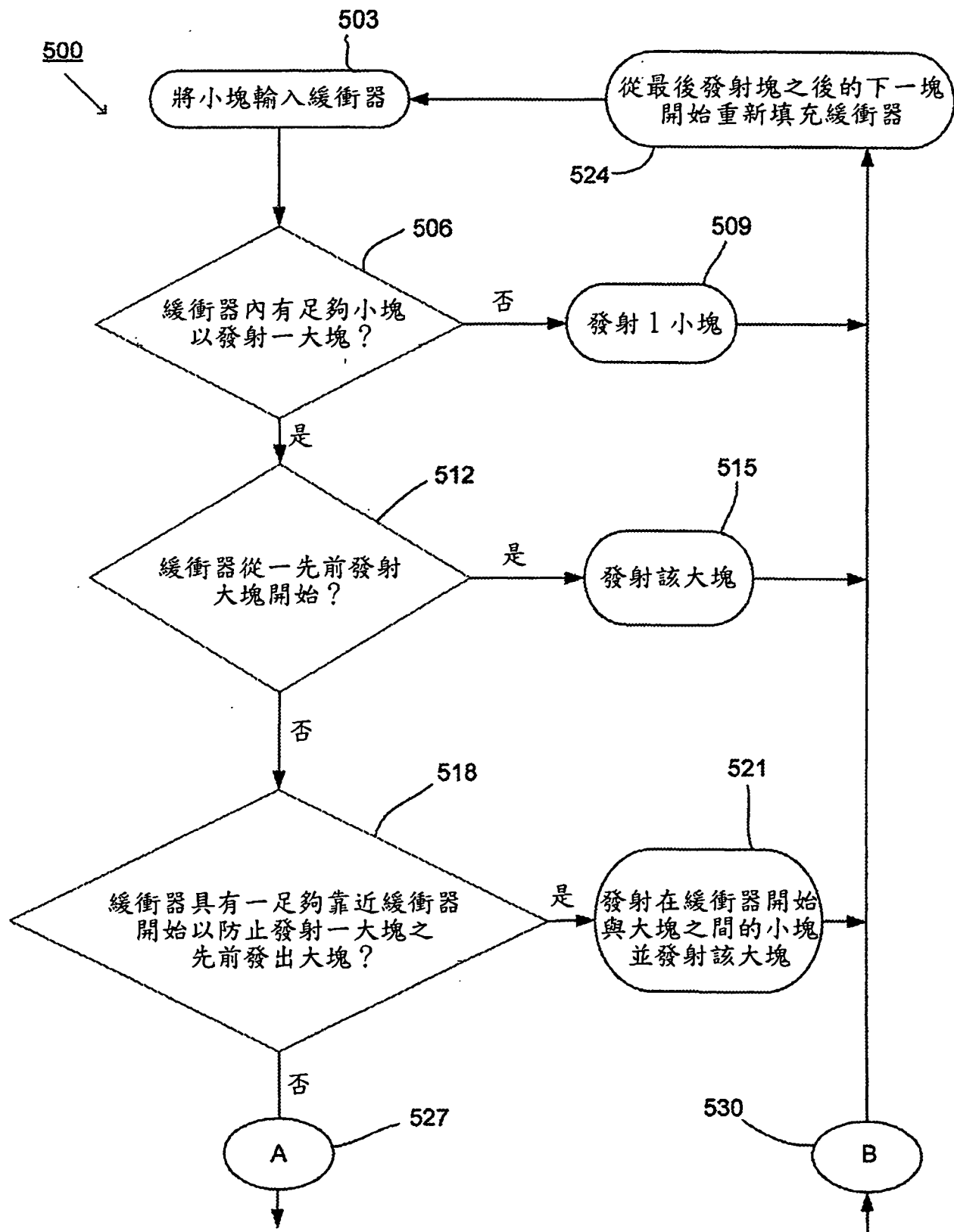


圖 5a

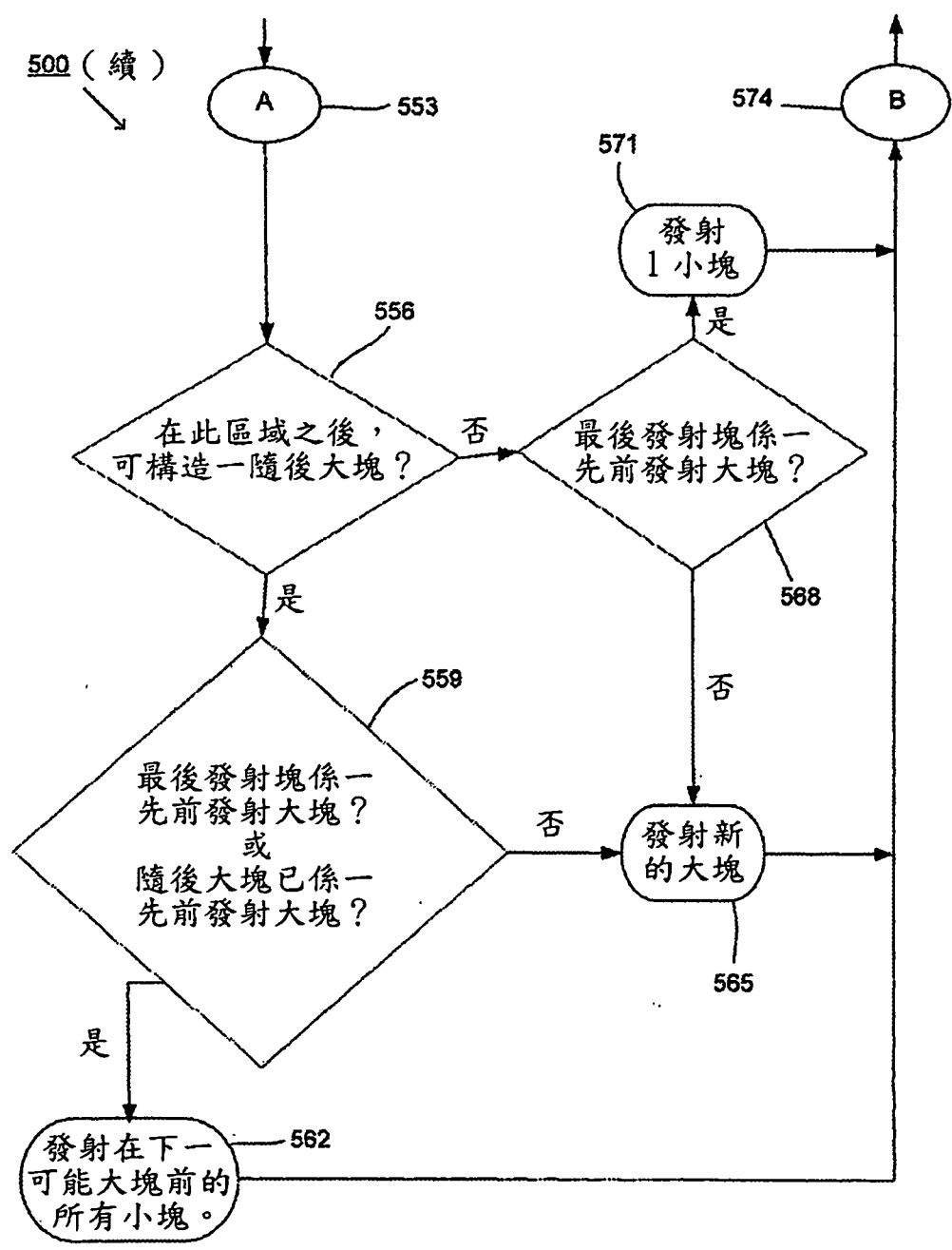


圖 5b

605——輸入資料： abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzca 600 (續)

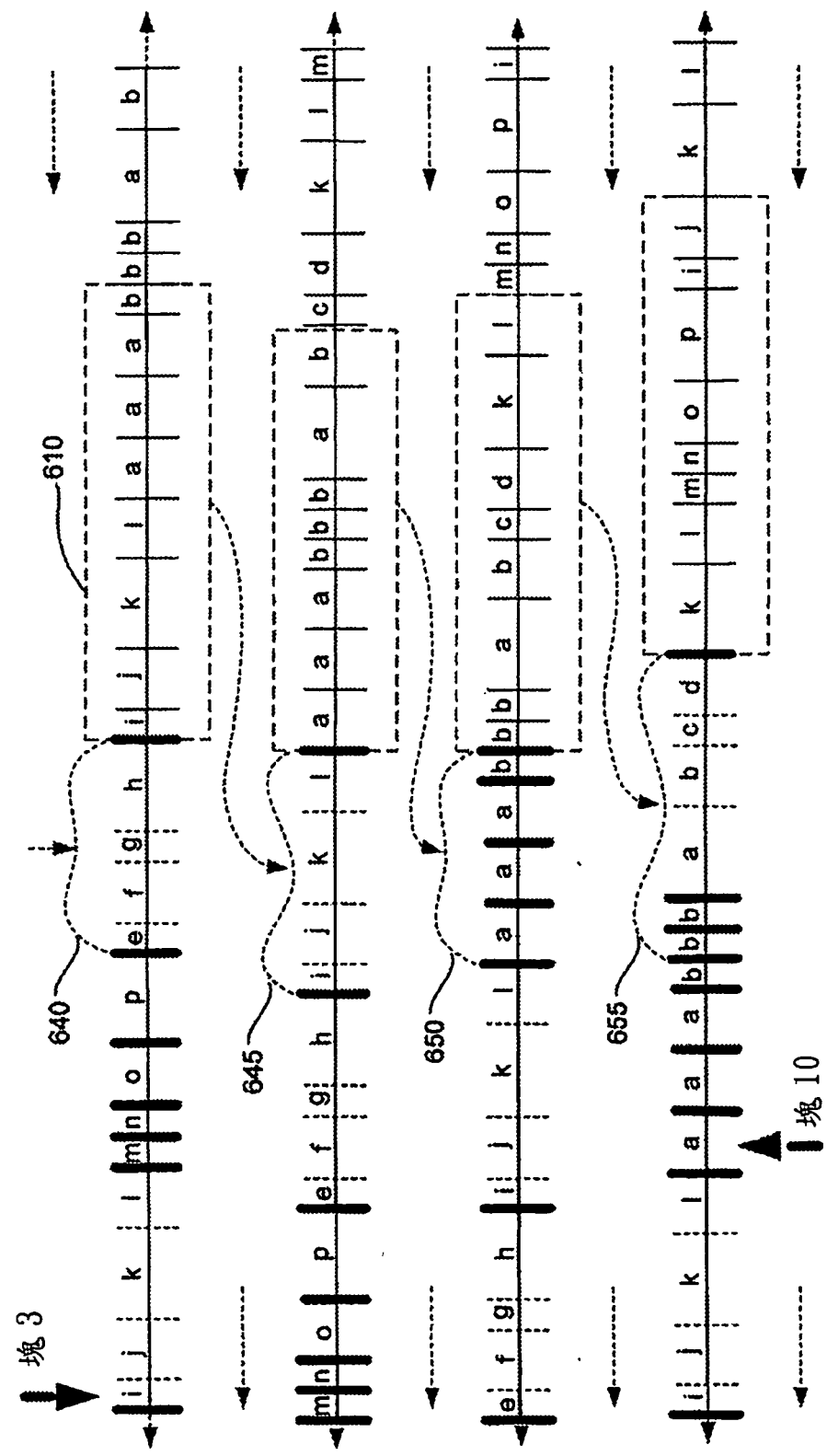


圖 6b

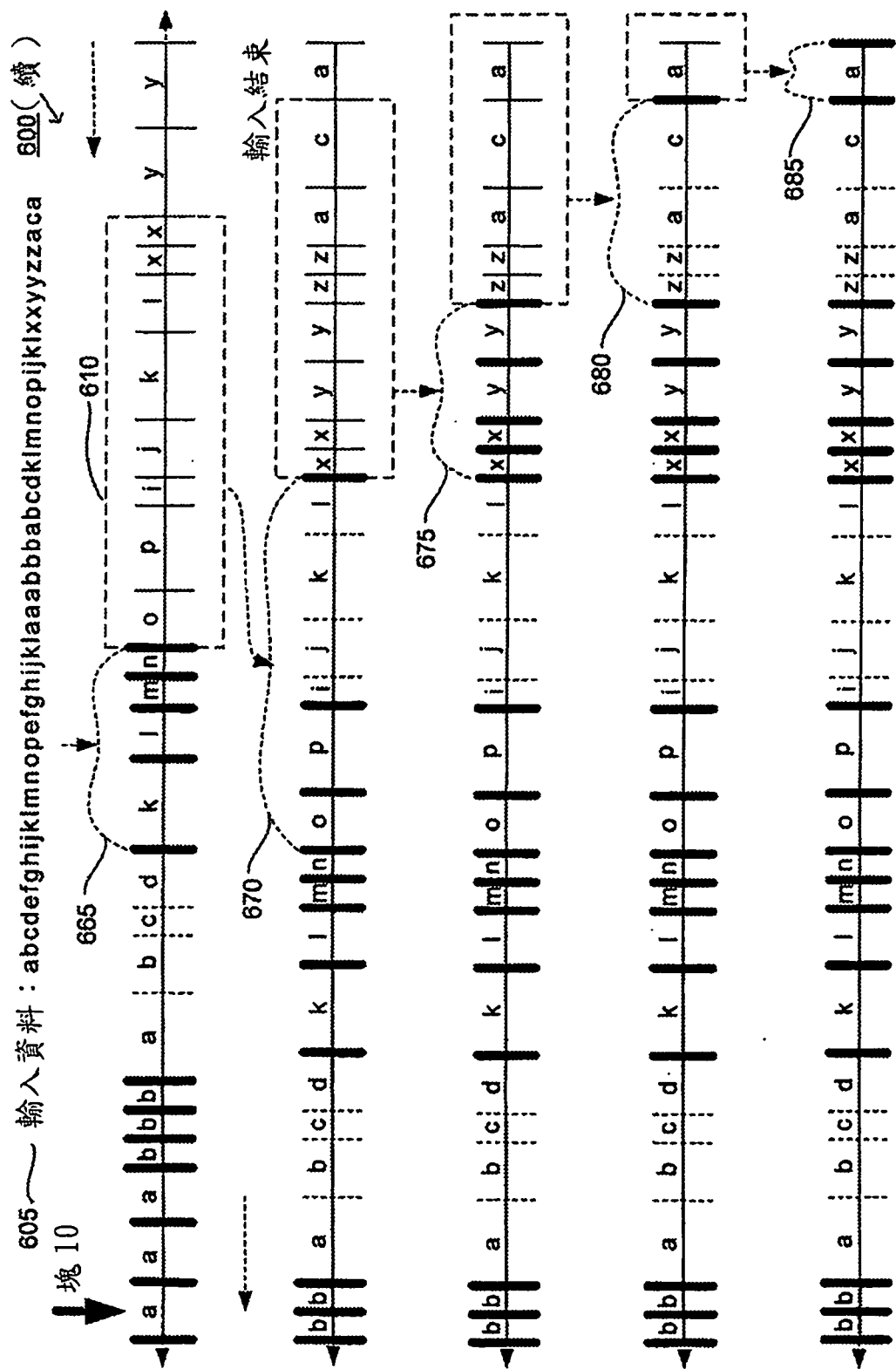


圖 6C

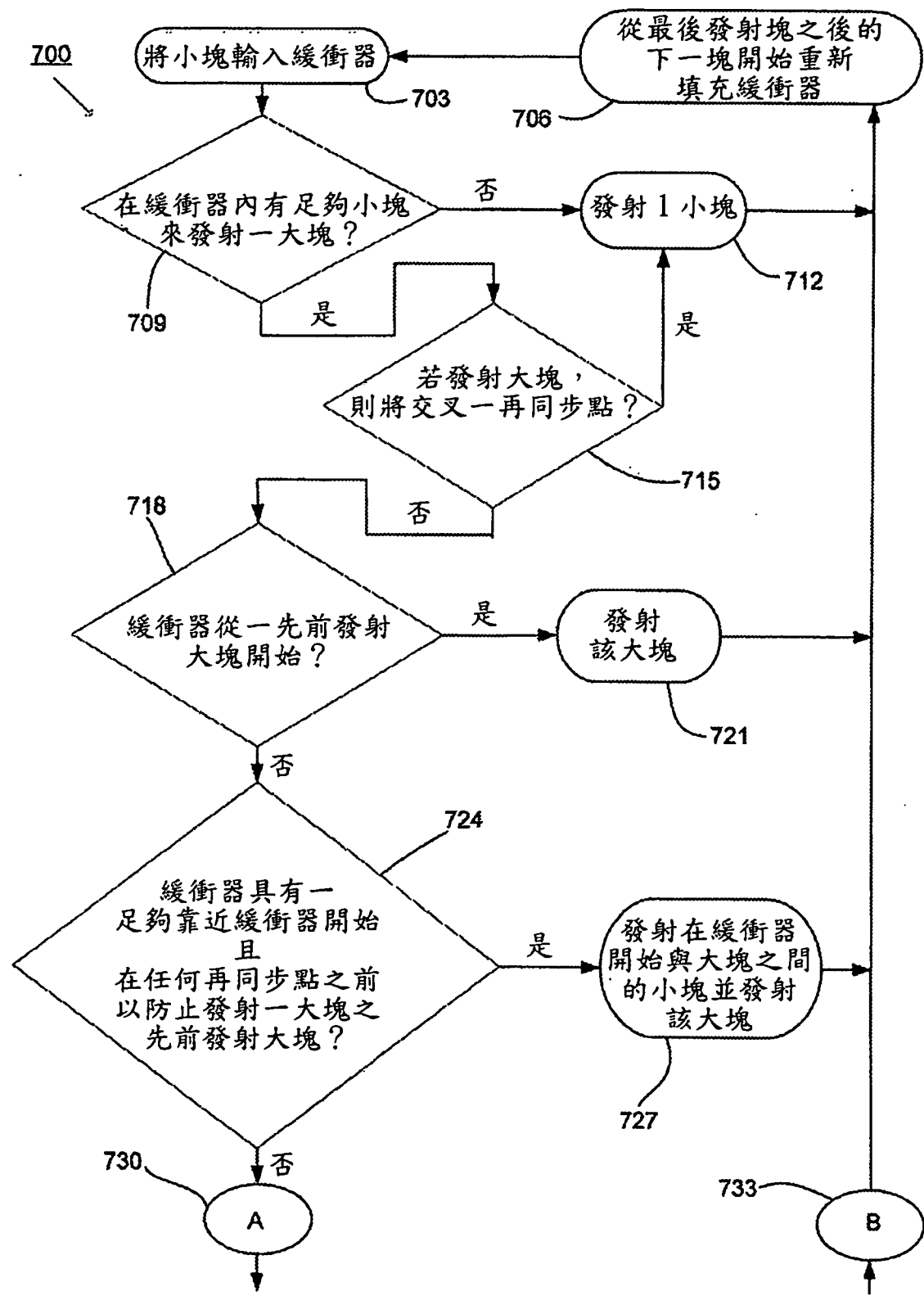


圖 7a

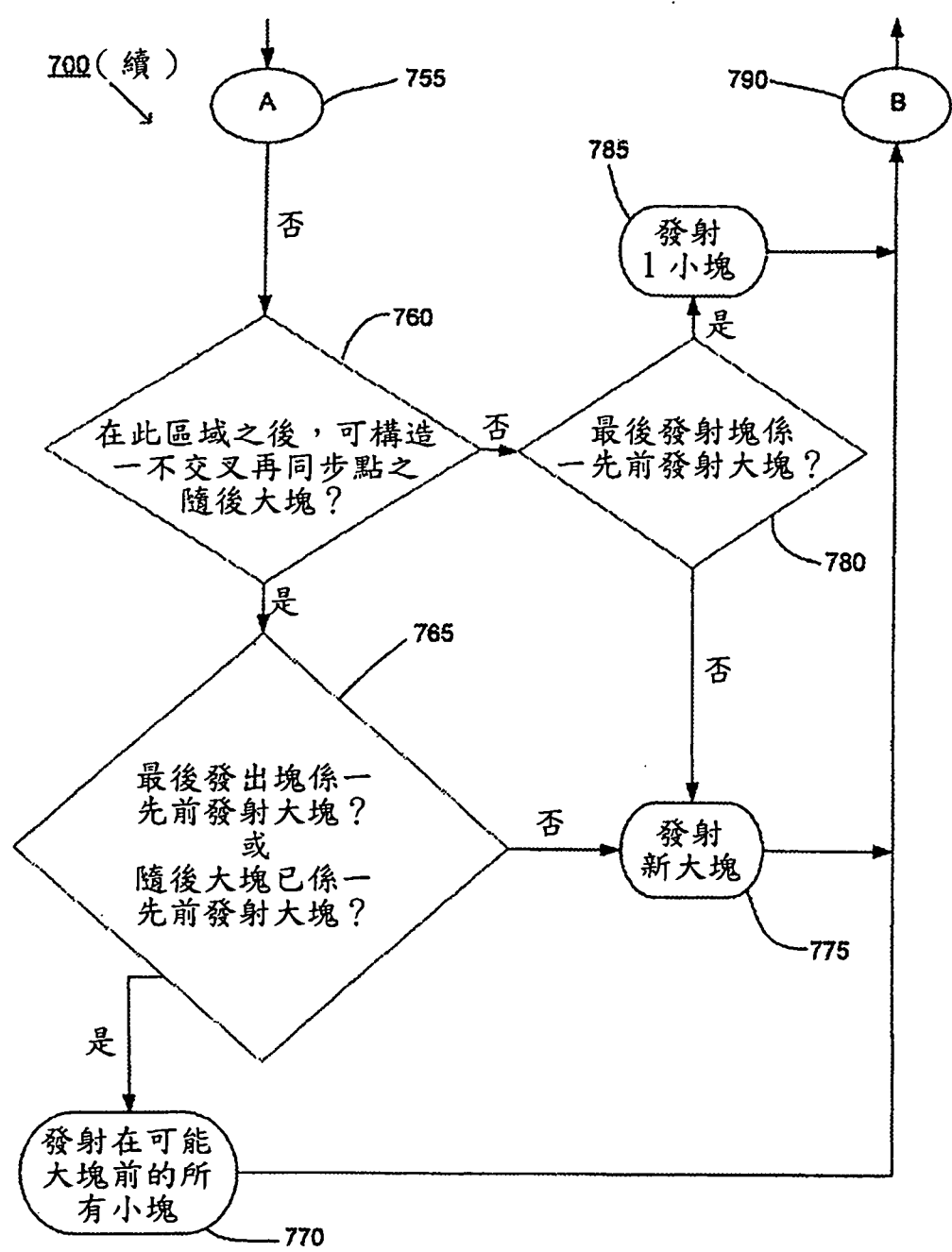


圖 7b

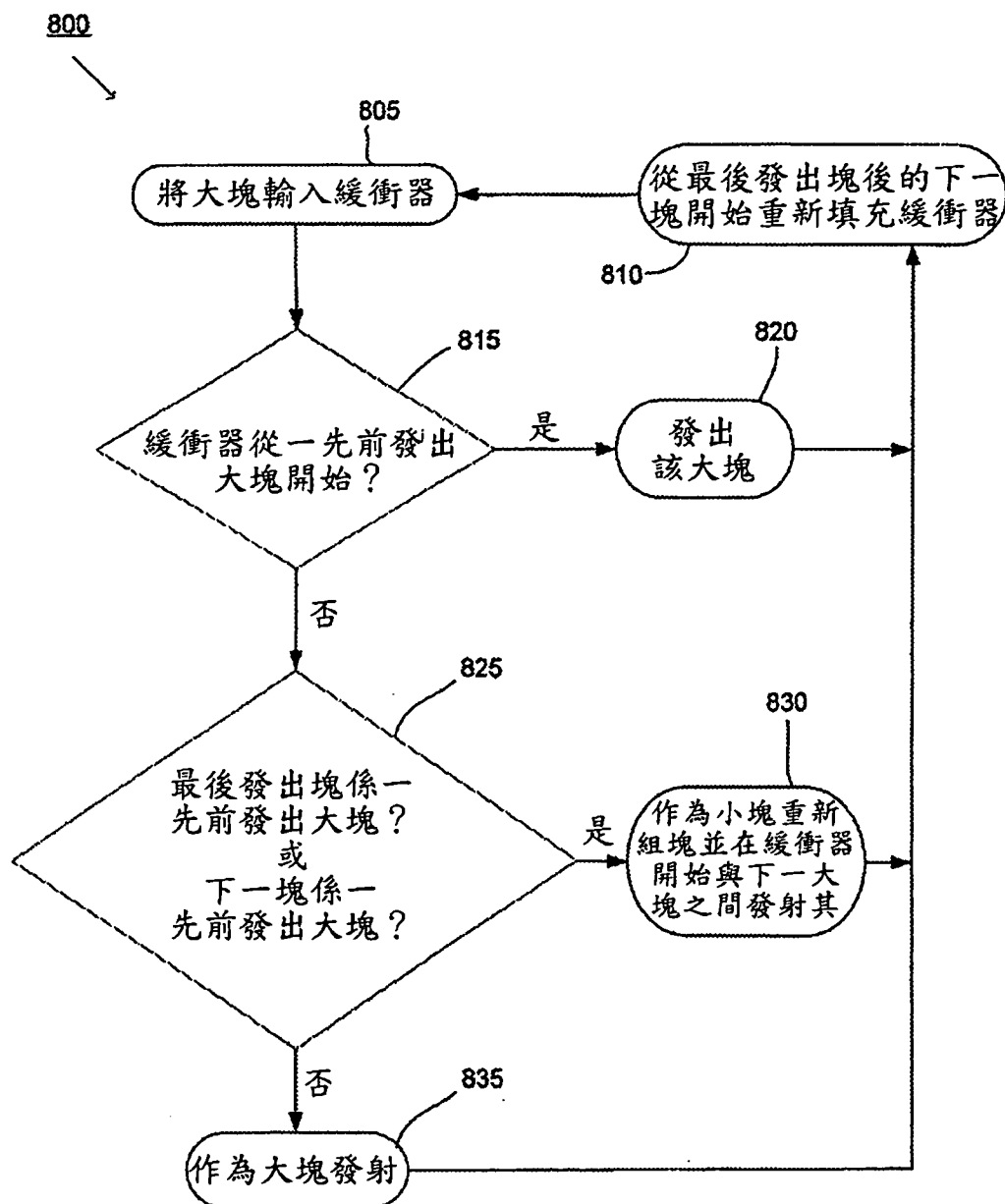
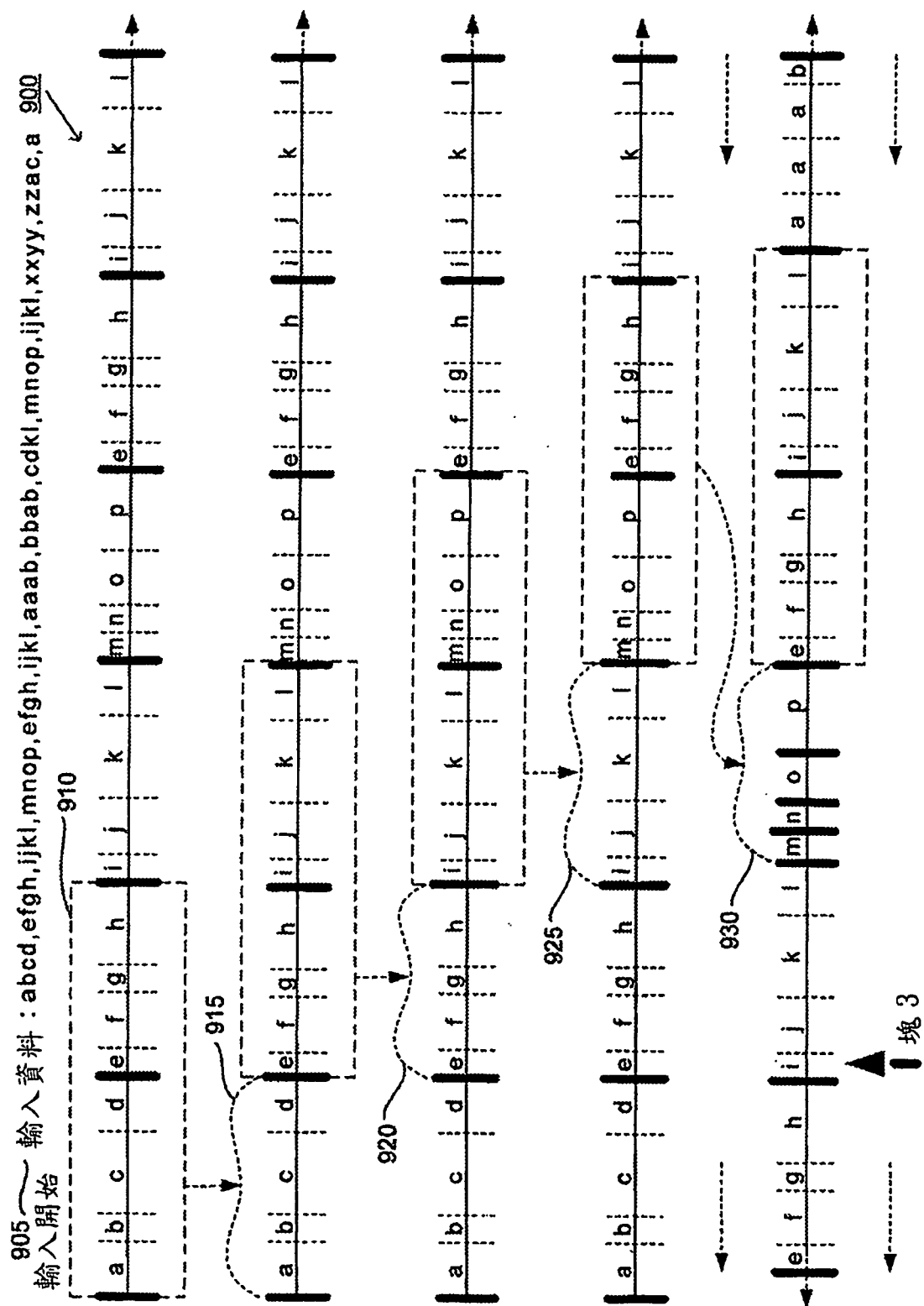


圖 8



905 — 輸入資料 : abcd,efgh,ijkl,mnop,efgh,ijkl,aaab,bbab,cdkl,mnop,ijkl,xyyy,zzac,a 900
(續)

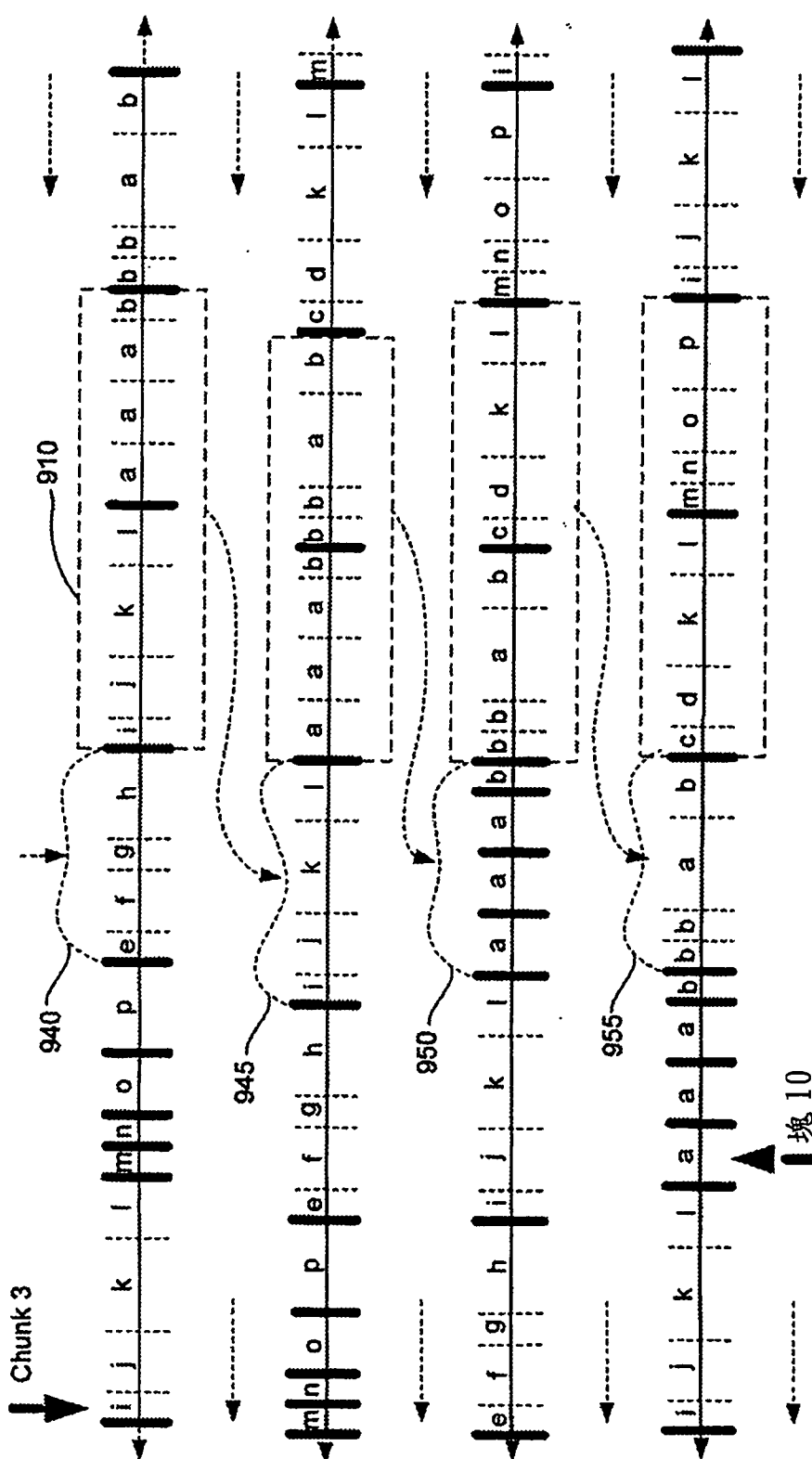


圖 9b

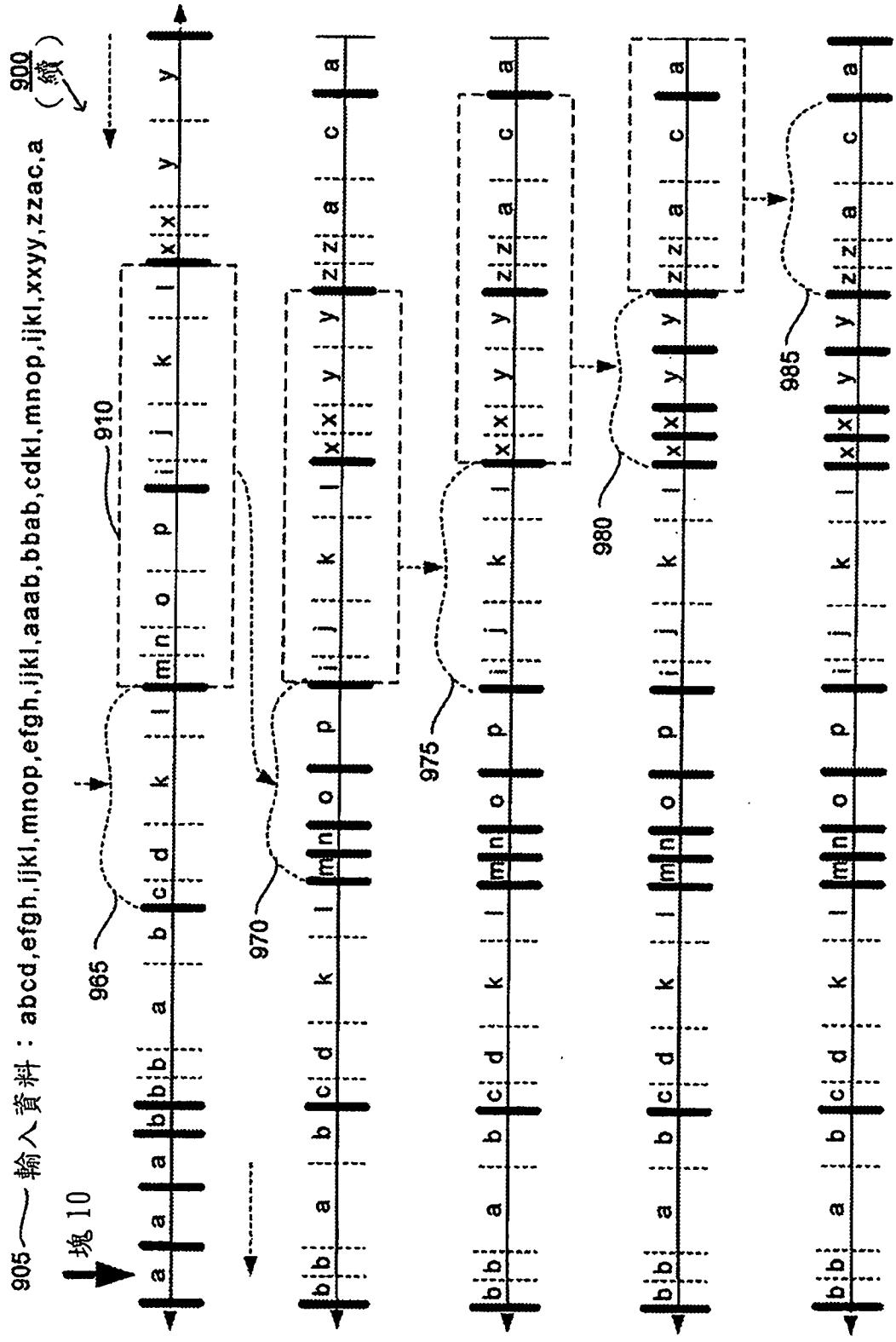


圖 9c

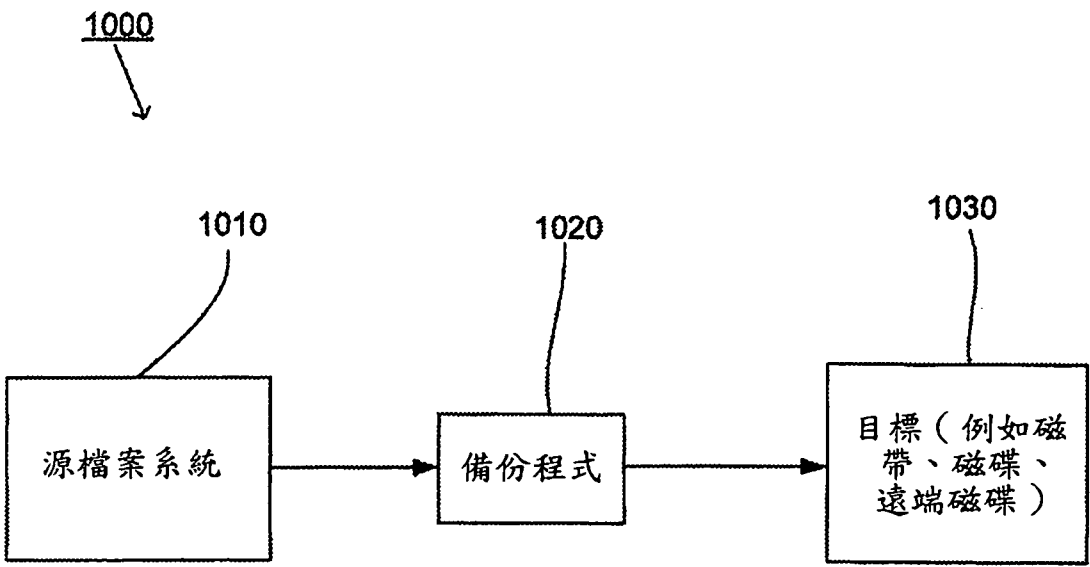


圖 10

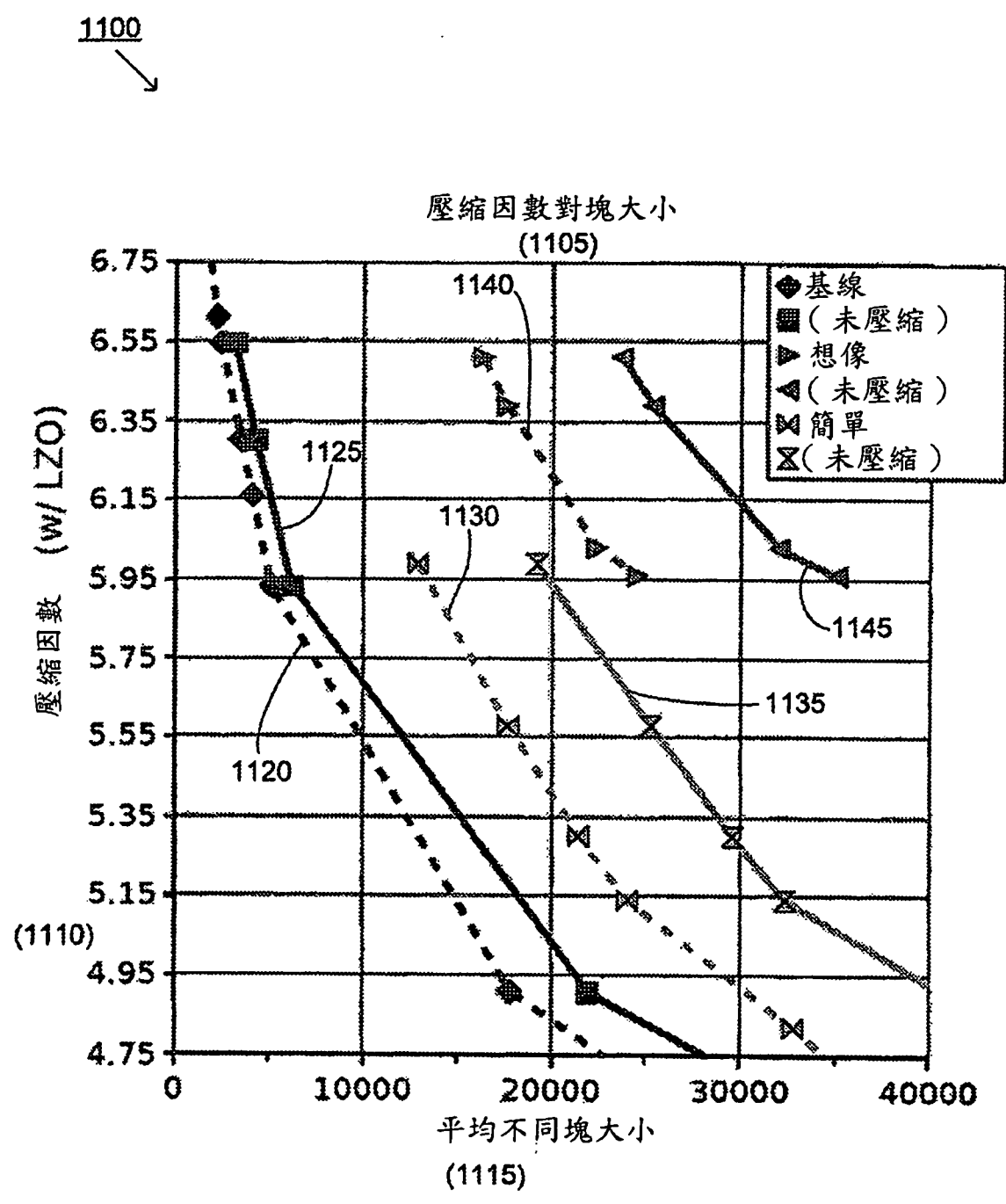


圖 11

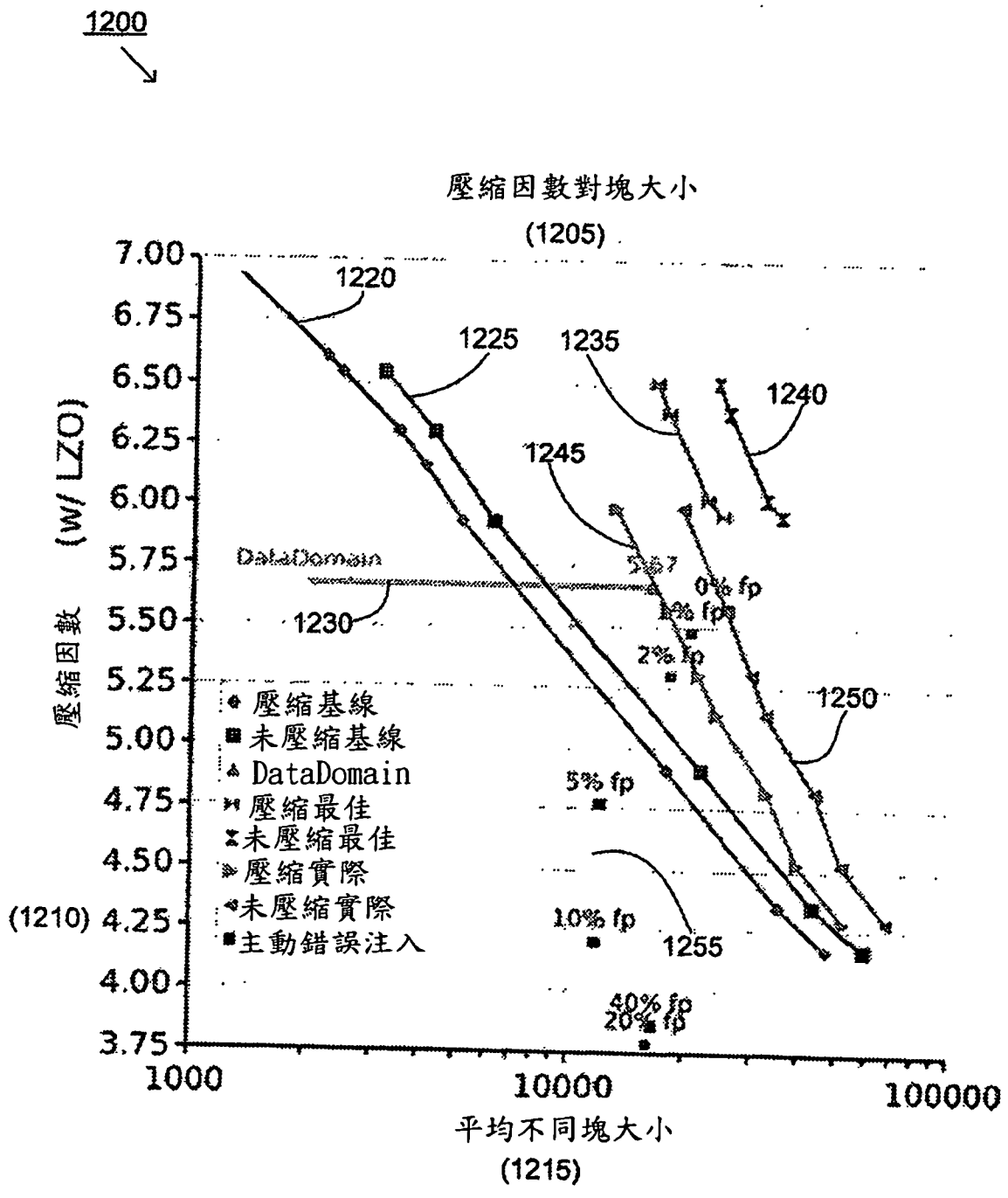


圖 12

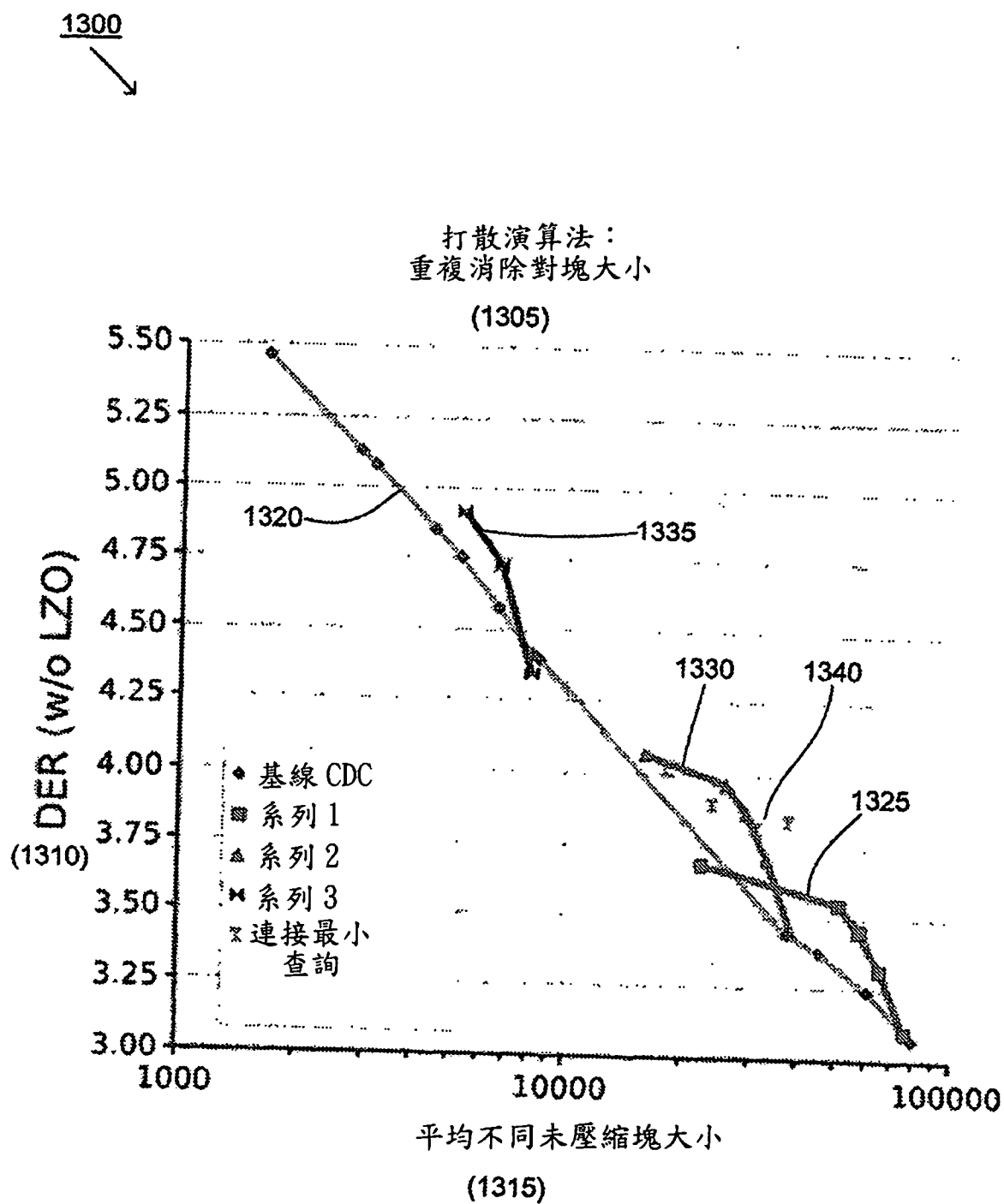


圖 13

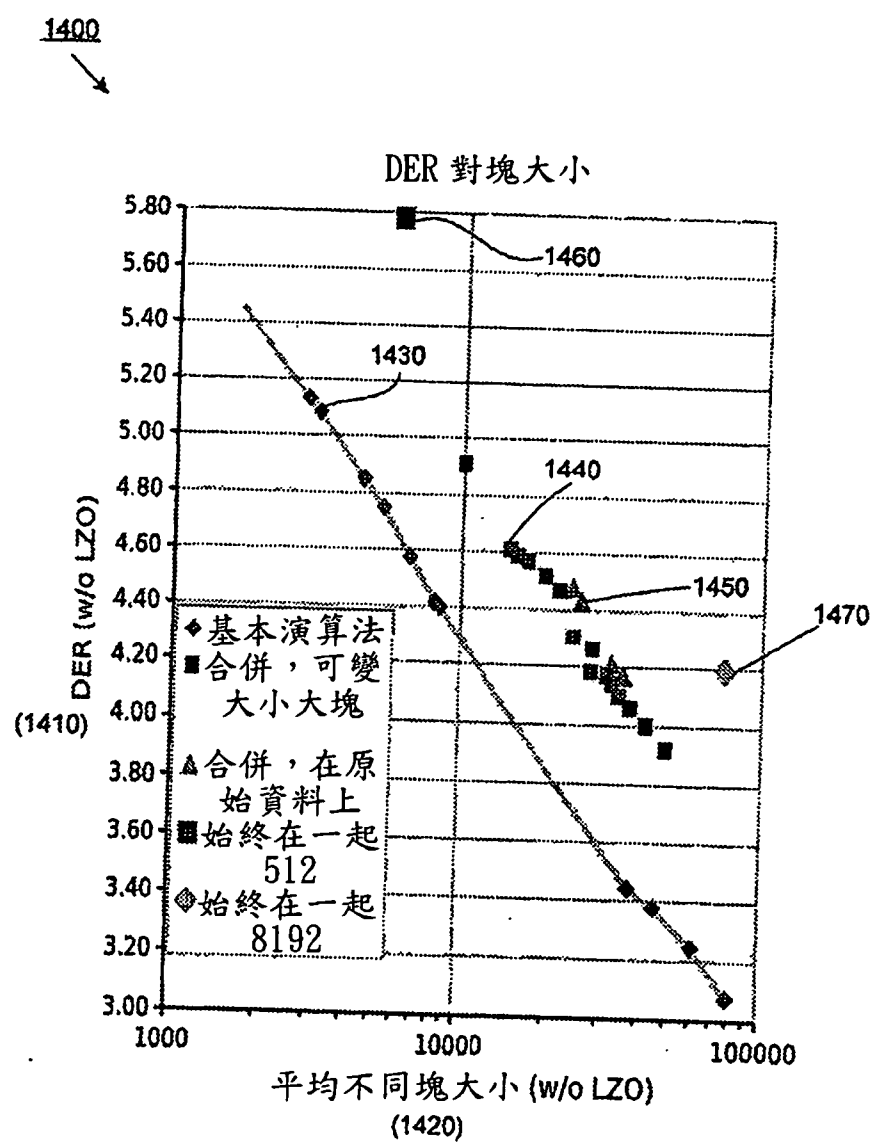


圖 14

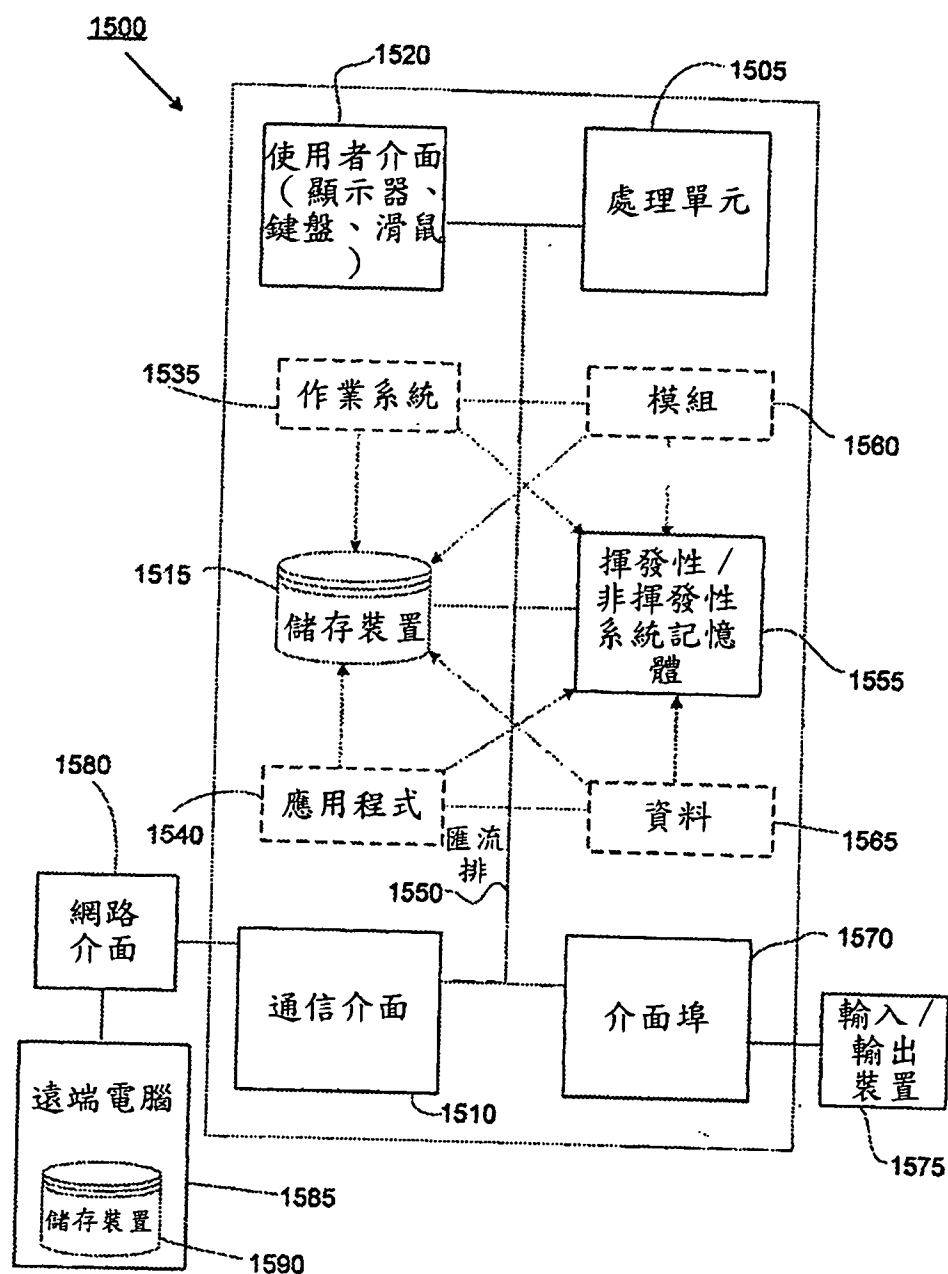


圖 15

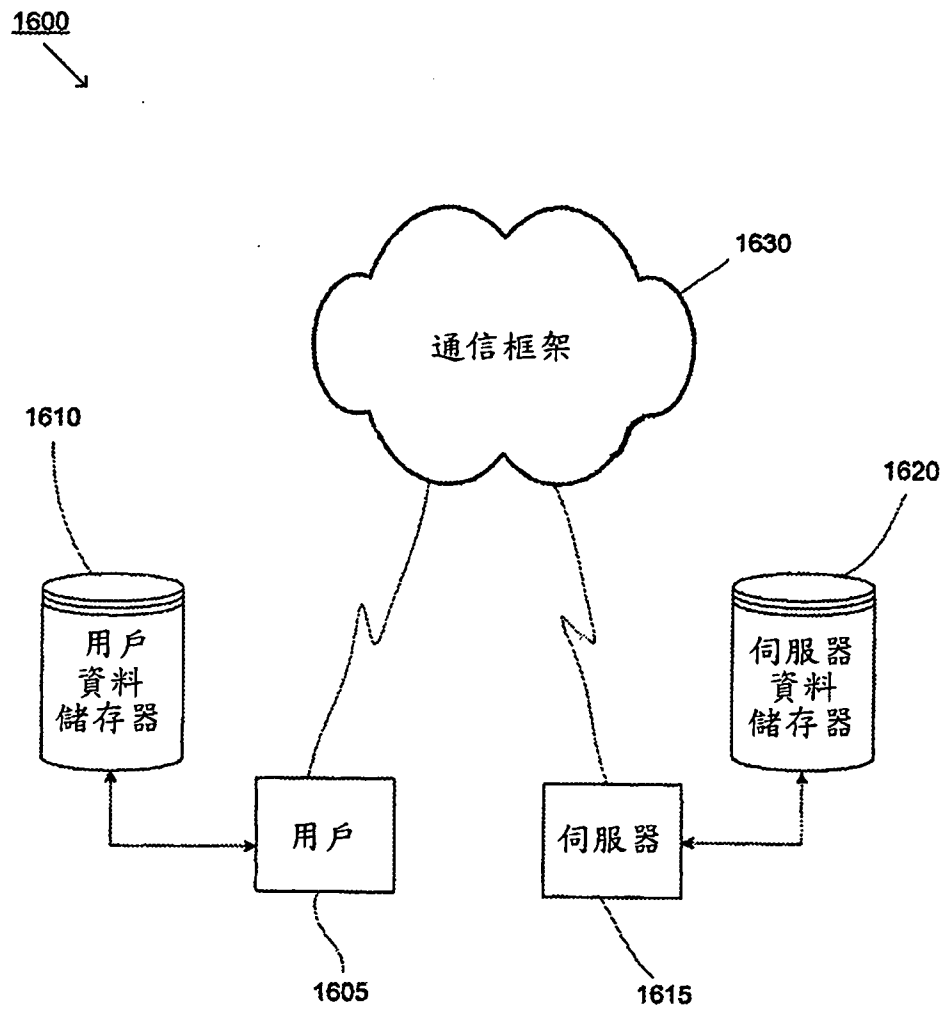


圖 16