

XML 動態索引資料節點標記編碼的字元斷句方法

陳靖國*

朝陽科技大學資訊管理系

jkchen@cyut.edu.tw

徐任鵬

朝陽科技大學資訊管理系

s9314630@cyut.edu.tw

摘要

XML 應用之普及，一些索引技術先後被用來加速 XML 資料的查詢。當 XML 資料有變動時，以往這些方法所建立之索引必須加以重建以便能存取正確又及時的資料。LSDX 可以有效地動態新增標記，不用重建索引，但更新索引時，其節點標記的編碼有字元斷句的問題，無法從某一個子孫(descendant)節點標記的編碼判斷出其父節點或更早祖先(ancestor)節點標記的編碼。在本篇論文中，我們引用杜威碼的概念，保留 LSDX 節點標記編碼規則的特徵，將索引樹節點標記的編碼規則加以改良，提出新的節點標記編碼規則來解決字元斷句的問題。最後，我們舉若干個實例說明 LSDX 方法不足之處，以及我們如何解決此問題。

關鍵詞: XML、索引、標記、字元斷句、杜威碼

1. 前言

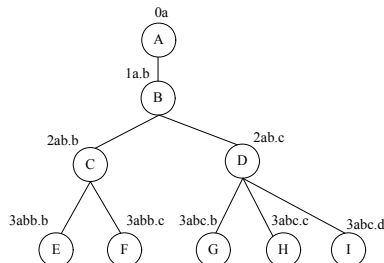
近年來，XML 已經成為網際網路上流行的資訊交換和呈現標準。處理大量 XML 資料，為了增進查詢的速度，最有效的方式之一是建立良好的索引結構。一個好的索引在查詢時不需要搜尋所有資料，即可快速地找到要查詢的資料[7]。在索引樹中每一個節點具有一個唯一的標記(label)，如果這些標記的編碼規則夠明確的話，可以從標記的編碼中清楚的解讀兩個已知節點之間的關係到底是祖孫(ancestor-descendant)關係或兄弟(sibling)的關係。為了要建立快速搜尋的索引，一些方法已經被提出，例如：路徑索引(path indexing)、編號方法(numbering scheme)和標記方法(labeling scheme)。

路徑索引[1,4,6,10]是將 XML 文件的資料用樹狀結構來呈現，針對索引樹中的每個節點，使用路徑命名的方式來為節點編碼。編號方法(numbering scheme)[3,8,9,11]可以分成以區域為基礎(region-based)的編號方法[9]和以字首為基礎(prefix-based)的編號方法[3,8,11]。以區域為基礎

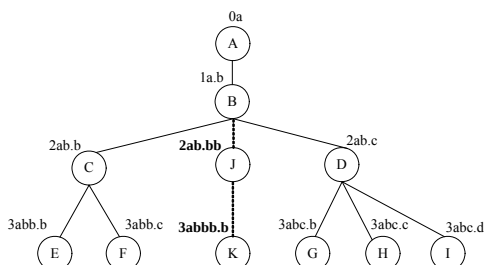
的編號方法對索引樹中每個節點編碼，是使用一個成對的數字，分別表示節點的開始和結束位置，從節點的標記編碼中可以判斷節點之間的關係。以字首為基礎的編號方法是記錄父節點和該節點標記編碼的方式來為索引樹中的每個節點編碼。雖然這些方法可以快速建立索引，但都具有一個共同的缺點，亦即當原始 XML 資料有所變動時，這些方法所建立之索引必須重建，既耗時又降低系統績效。針對此問題，論文[2]和[5]當中的 LSDX 是一個動態索引 XML 資料的標記方法，用來產生 XML 資料索引樹節點標記，若 XML 資料更動頻繁時，只會產生新節點的標記，而舊節點標記不會受到影響。

LSDX 在建索引樹時，子節點標記的編碼為父節點標記的編碼(不包含句點'.')+句點'.'+該節點的新增碼，由於只使用一個句點'.'來區別句點前父節點編碼和句點後的子節點新增碼，當索引樹新增節點和其子節點時，節點標記的編碼可能有字元斷句的問題，無法從某一個子孫節點標記的編碼判斷出其父節點或更早祖先節點標記的編碼。因為 LSDX 正常建立的索引樹其後代節點標記可以正確斷句找到其前代節點，但是索引樹建好之後，後續新增的節點和子孫節點的標記就可能無法正確斷句找到正確的前代節點。舉例如下，使用 LSDX 對索引樹中節點的標記編碼，如圖一(a)所示，根節點'A'的標記編碼為"0a"，其子節點'B'的標記編碼為"1a.b"，節點'B'的第一個子節點'C'的標記編碼為"2ab.b"，第二個子節點'D'的標記編碼為"2ab.c"，其餘節點的標記編碼依此類推，在此圖中任何子孫節點的標記編碼都可以正確斷句找到其前代節點。如圖一(b)所示，若在子節點'C'和'D'之間新增一個節點'J'，其標記編碼為"2ab.bb"，然後在節點'J'下新增一個子節點'K'，其標記編碼為"3abbb.b"，若要從節點'K'標記的編碼"3abbb.b"找到其前一代或更早代的節點是很困難的，因為有正確字元斷句的問題，可能會誤判節點'K'的父節點標記為"2abb.b"而不是節點'J'的"2ab.bb"。在本篇論文中，為了要解決這個斷句問題，我們引用論文[11]所描述的杜

威編碼的概念，來改良 LSDX 建索引樹節點標記的編碼規則，每個節點標記編碼用一個句點'.'隔開其父節點標記的編碼和該節點的新增碼來解決字串斷句的問題，至於節點的新增碼編碼方式則採用原 LSDX 節點標記編碼的規則。



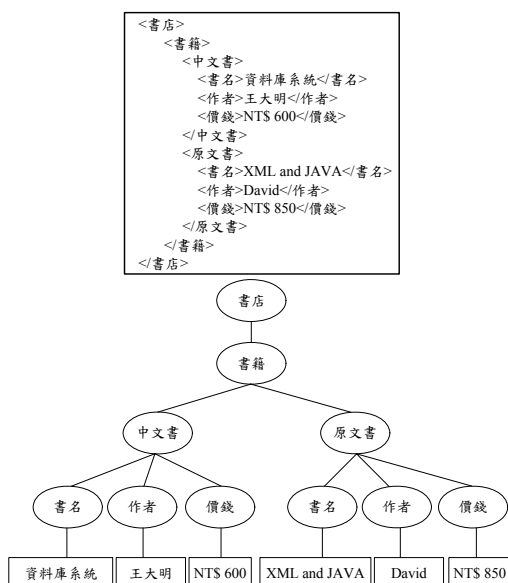
圖一(a): 建立索引時，LSDX 對節點標記編碼



圖一(b): 新增節點時，LSDX 對節點標記編碼

2. 相關工作

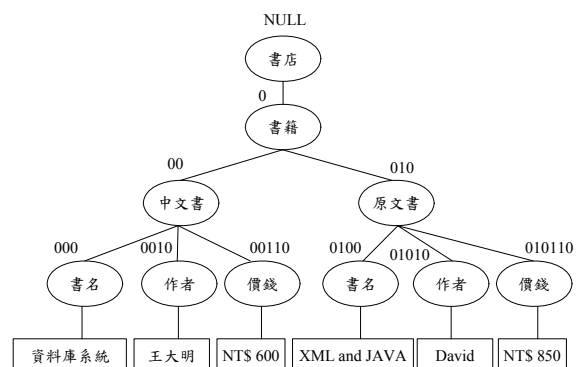
在說明以往相關研究之前，我們先簡單說明一份 XML 文件的格式以及邏輯樹狀圖。如圖二所示，一份 XML 文件可以用樹狀結構來呈現，在樹狀結構中用節點來表示元素、屬性，矩形用以表示元素內容(element content)，用線段來表示節點之間的父子關係。



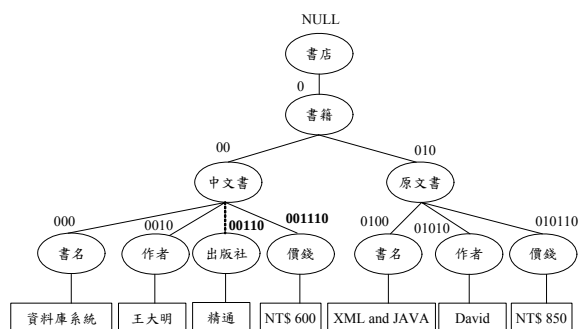
圖二: 一份 XML 文件和 XML 資料樹

XML 之索引研究主要有路徑索引、編號方法和標記方法這幾種。論文[1,4,6,10]使用路徑索引來搜尋 XML 資料。當原始 XML 資料有所變動時，所建立之索引必須被更新，而這些方法不支援動態地更新 XML 資料。在論文[9,11]中以區域為基礎的編號方法和以字首為基礎的編號方法也同樣無法支援索引動態更新。

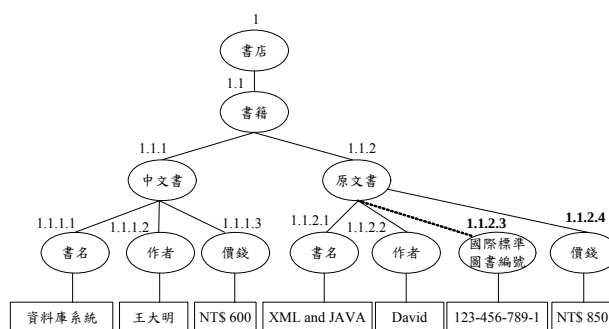
論文[3]中提出了一個簡單字首(simple prefix, SP)標記方法，它是一個以字首為基礎的編號方法，針對一個節點v的每個子節點指定一個特定的標記編碼。節點的標記編碼規則為(1)根節點：其標記為空字串。(2)非根節點：假設 $L(v)$ 代表節點v的標記， $L(v)=L(u)+(1)^{i-1}0$ ，u是v的父節點，i代表節點v是父節點u的第i個兒子。舉例說明，使用SP標記方法，對圖二XML索引樹中節點的標記編碼，結果如圖三(a)所示。”書店”根節點的標記編碼為空字串(NULL)，其”書籍”子節點的標記編碼為”0”，”書籍”節點下的”中文書”子節點的標記編碼為”00”，其”原文書”子節點的標記編碼為”010”，其餘節點的標記編碼依此類推。SP標記方法不是一個動態索引XML資料方法，因為當有新增節點N加入索引樹時，無法給與N一個適當的標記而不影響某些舊節點標記，因為N之後的兄弟節點的標記必須更動，由於SP方法的節點標記編碼長度過長，所以此方法在異動標記時非常耗時間。如圖三(b)所示，若在標記編碼為”00110”的作者子節點後，新增一個”出版社”節點，則其節點的標記編碼應為”001110”，但其右兄弟”價錢”節點的標記編碼已經是”001110”，所以”出版社”節點的右兄弟”價錢”節點的標記編碼要修改為”0011110”。



圖三(a): 簡單字首標記方法

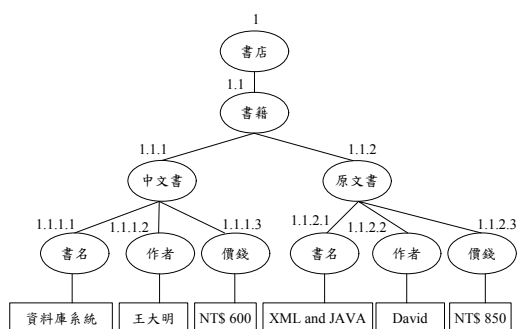


圖三(b): 新增節點時，部份節點標記必須更改



圖四(b): 新增節點時，部份節點標記必須更改

在論文[11]中，使用杜威字首為基礎的編號方法(Dewey prefix-based numbering scheme)，其方法如下。已知一個節點 v 有 n 個子節點， $u_1, u_2, u_3, \dots, u_n$ ， $L(v)$ 表示節點 v 的標記，根節點之標記為'1'，子節點 u_i 的標記為 $L(u_i)=L(v).i$ ， i 表示 u_i 為 v 的第 i 個子節點。舉例說明，使用杜威字首為基礎的編號方法，對圖二XML索引樹中節點的標記編碼，如圖四(a)所示，"書店"根節點的標記編碼為'1'，其"書籍"子節點的標記編碼為'1.1'，"書籍"節點的"中文書"子節點的標記編碼為'1.1.1'，其"原文書"子節點的標記編碼為'1.1.2'，依此類推。此方法同樣無法給與新增節點 N 一個適當的標記而不影響某些舊節點標記，因為當新增 N 時會造成 N 之後的兄弟節點的標記有所更動，如圖四(b)所示，在標記編碼分別為'1.1.2.2'和'1.1.2.3'的"作者"節點和"價錢"節點中間，新增一個"國際標準圖書編號"(ISBN)節點，因為此新節點是"原文書"節點的第三個子節點，因此"國際標準圖書編號"節點的標記編碼應為'1.1.2.3'，但其右兄弟"價錢"節點的標記編碼為'1.1.2.3'，所以其右兄弟"價錢"節點的標記編碼要修改為'1.1.2.4'，如此是非常沒有效率的。



圖四(a): 杜威字首為基礎的編號方法

有別於在論文[1,3,6,9,11,12,13]中，對於XML資料索引樹中每個節點的標記是使用數字的命名方式，論文[2,5]的節點標記使用數字加上英文字母的命名方式。LSDX 在建索引樹時，子節點標記的編碼包含父節點標記的編碼，但由於只使用一個句點'.'來區別父節點標記編碼(不句含句點'.')與子節點的新增碼，某些節點標記的編碼因為有字串斷句的問題，無法判斷出其父節點或更早祖先節點標記的編碼。雖然 LSDX 正常建立的索引樹其後代節點標記可以正確斷句找到其前代節點，但是後續新增的節點和子孫節點的標記就有可能無法正確斷句找到正確的前代節點。只不過，論文[2]主要目的在解決 LSDX 在建立索引樹的過程中，同一階層的節點最多只能標記到二十五個的問題，且在索引樹作更動時，在兩個節點中以向左的方向連續新增兩個節點，造成第二個新增節點的標記無法編碼的另一個問題。

3. 節點標記編碼字元斷句的探討

LSDX在建立索引樹時，節點標記的編碼規則如下：已知一個節點 v 具有 n 個子節點： $u_1, u_2, u_3, \dots, u_n$ ，第一個子節點 u_1 的標記編碼為 u_1 所在的level+節點 v 的標記編碼+'.'+'b'，第二個子節點 u_2 的標記編碼為 u_2 所在的level+節點 v 的標記編碼+'.'+'c'，其他子節點的標記編碼依此類推。由 LSDX建索引樹節點標記的編碼規則，我們發現節點標記編碼只使用一個句點'.'隔開其父節點標記的編碼(不句含句點'.')和該節點的新增碼，某些節點標記的編碼因為有字串斷句的問題，無法判斷出其父節點或更早祖先節點標記的編碼。雖然 LSDX正常建立的索引樹其後代節點標記可以正確斷句找到其前代節點，但是後續新增的節點和子孫節點的標記就可能無法正確斷句找到正確的前代節點。本篇論文，引用杜威字首為基礎的編號方法的概念，每個節點標記編碼用一個句點'.'隔開其父節點標記的編碼和該節點的新增碼，保留 LSDX節點標記編碼的特徵，亦即將索

引樹節點標記的編碼規則加以改良來解決字元斷句的問題。另外將LSDX節點標記編碼中節點所在level數字去掉，是因為我們的節點標記編碼方法用一個句點'.'隔開其父節點標記的編碼和該節點的新增碼，如此可以由節點編碼中句點'.'的個數來判斷該節點所在的level數。我們的節點標記編碼方法如下。已知一個節點 p 有 n 個子節點： $c_1, c_2, c_3, \dots, c_n$ ， $L(p)$ 為父節點 p 的標記。

(1) 根節點的標記為'a'。

(2) 子節點 c_α 的標記為 $L(c_\alpha)=L(p).\textcircled{\alpha}$ ，而 $\textcircled{\alpha} \in \{b, c, \dots, z\}$ ， $\textcircled{\alpha} = \gamma_{\alpha+1}$ ， $\gamma_{\alpha+1}$ 為英文小寫字母第 $\alpha+1$ 個。

從子孫節點的標記編碼可以判斷出其父節點或更早祖先節點標記的編碼，我們的判斷方法如下。已知節點 c 有 n 個祖先節點： $p_1, p_2, p_3, \dots, p_n$ ，節點 p_i 為節點 c 的前 i 層節點，即節點 p_i 為節點 c 的父節點，節點 p_2 為節點 c 的祖父節點，依此類推，節點 p_n 為根節點，而 $1 \leq n < \text{樹的高度}$ 。若要找節點 c 前 i 層節點標記的編碼，是從子孫節點 c 標記編碼的末位字元往左掃描，當碰到第 i 個句點'.'則停止，第 i 個句點'.'左邊(不包含此句點'.')的所有字元即為節點 p_i 標記的編碼。

4. 實例說明

以下用若干實例來說明當原始 XML 資料有所更動，對索引樹新增節點和其子節點時，LSDX 節點標記編碼無法正確字元斷句的問題，以及我們的方法如何解決此問題。如圖二所示，對 XML 資料建立索引樹時，使用我們改良後的方法對每個節點的標記編碼，結果如圖五所示，"書店"是根節點，其標記的編碼為'a'，"書店"根節點下的"書籍"子節點標記編碼為"a.b"，"書籍"節點的"中文書"子節點的標記編碼為"a.b.b"，其"原文書"子節點的標記編碼為"a.b.c"，其餘節點的標記編碼依此類推。

圖六(a)顯示在 XML 資料樹中，"書籍"節點下的第一個"中文書"子節點前新增一個"中文書"節點，然後在"中文書"節點下新增一個"書名"子節點的情況。圖六(b)顯示索引樹新增節點時，LSDX 對新增節點的標記編碼，在 XML 資料索引樹中，"書籍"節點的第一個"中文書"子節點前新增一個"中文書"節點，其標記的編碼為"2ab.ab"，然後在標記編碼為"2ab.ab"的"中文書"節點下新增一個"書名"子節點，其標記的編碼為"3abab.b"，由"書名"子節點標記的編碼"3abab.b"無法判斷出其父節點或更早祖先節點標記的編碼，因為有斷句的問題。圖六(c)說明

我們的方法可以解決 LSDX 節點標記編碼字元斷句的問題，在節點標記編碼為"a.b.b"的"中文書"節點前新增一個"中文書"節點，其標記的編碼為"a.b.ab"，然後在標記編碼為"a.b.ab"的"中文書"節點下新增一個"書名"子節點，其標記的編碼為"a.b.ab.b"，若要由此編碼判斷出其父節點或更早祖先節點標記的編碼，使用我們的判斷方法，可以很容易得知其父節點的標記編碼為"a.b.ab"，祖父節點的標記編碼為"a.b"，曾祖父節點的標記編碼為'a'。

圖七(a)顯示在 XML 資料樹中，"書籍"節點下的"中文書"和"原文書"子節點之間新增一個"中文書"節點，然後在此新增節點下新增一個"書名"子節點的情況。圖七(b)顯示索引樹新增節點時，LSDX 對新增節點的標記編碼，在 XML 資料索引樹中，"書籍"節點下節點的"中文書"("2ab.b")和"原文書"("2ab.c")兩個子節點之間新增一個"中文書"節點，其標記的編碼為"2ab.bb"，然後在此節點下再新增一個"書名"子節點，其標記的編碼為"3abbb.b"，由"書名"子節點標記的編碼"3abbb.b"可能誤判其父節點編碼為"2abb.b"。圖七(c)說明我們的方法可以解決斷句的問題，在標記編碼分別為"a.b.b"和"a.b.c"的"中文書"和"原文書"子節點之間新增一個"中文書"節點，其標記的編碼為"a.b.bb"，然後在此節點下新增一個"書名"子節點，其標記的編碼為"a.b.bb.b"，若要由此編碼判斷出其父節點和祖先節點標記的編碼，可以很容易得知其父節點的標記編碼為"a.b.bb"，祖父節點的標記編碼為"a.b"，曾祖父節點的標記編碼為'a'。

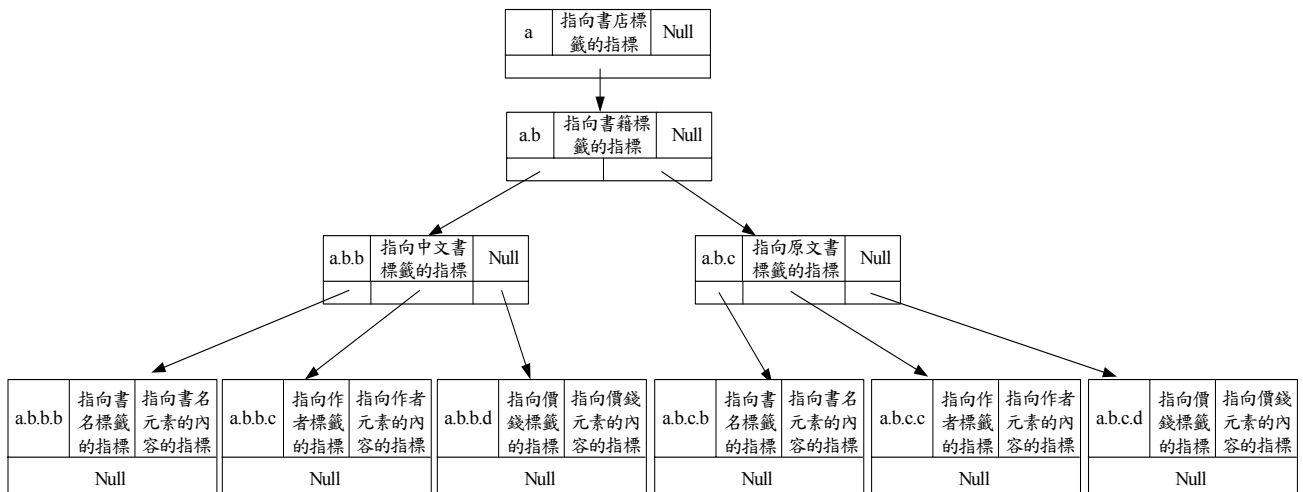
圖八(a)顯示在 XML 資料樹中，第二十五個"原文書"節點後面新增一個"原文書"節點，然後在此新增節點下新增一個"書名"子節點的情況。圖八(b)顯示索引樹新增節點時，LSDX 對新增節點的標記編碼，在 XML 資料索引樹中，"書籍"節點下第二十五個"原文書"節點後面新增一個"原文書"節點，其標記的編碼為"2ab.zb"，然後在此節點下新增一個"書名"子節點，其標記的編碼為"3abzb.b"，由"書名"子節點標記的編碼"3abzb.b"可能誤判其父節點編碼為"2abz.b"。圖八(c)說明我們的方法可以解決斷句的問題，在"書籍"節點下第二十五個"原文書"節點後面新增一個"原文書"節點，其標記的編碼為"a.b.zb"，然後在此節點下新增一個"書名"子節點，其標記的編碼為"a.b.zb.b"，若要由此編碼判斷出其父節點和祖先節點標記的編碼，可以很容易得知其父節點的標記編碼為"a.b.zb"，祖父節點的標記編碼為"a.b"，曾祖父節點的標記編碼為'a'。

5. 結論

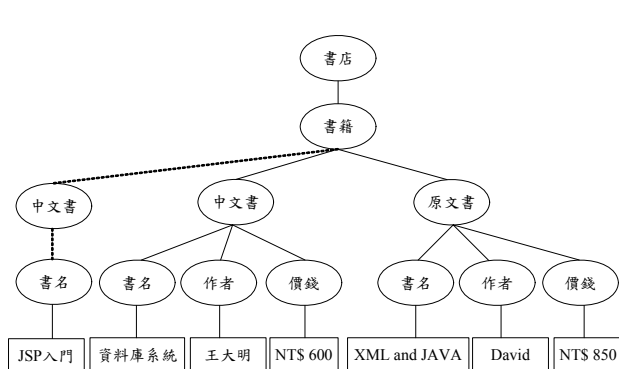
本篇論文，主要解決 LSDX 在對索引樹新增節點和子節點後，從新增節點的標記編碼，可能無法從某一個子節點標記編碼找到其父節點或更早祖先節點的問題。雖然 LSDX 在建立索引樹時，由後代節點標記的編碼可以正確斷句找到其前代節點，但在更新索引樹後，新增的節點和子孫節點，有可能無法從它們標記的編碼正確斷句找到其前代節點。我們改良的方法是引用杜威碼的概念，每個節點標記編碼用一個句點 '.' 隔開其父節點標記的編碼和該節點新增碼，新增碼編碼方式則保留 LSDX 節點標記編碼的特徵。我們引用杜威碼的概念而不直接採用杜威碼編碼規則，是因為當原始 XML 資料更動時須對索引更新，而杜威碼所編出來的碼是無法做動態更新，所以我們保留原來 LSDX 編碼方法以便節點的標記編碼能動態擴充。綜合這兩方法的優點我們提出一個新的節點標記編碼方法。我們的方法與 LSDX 方法相較之下，就節點標記的編碼長度而言，雖然比 LSDX 方法產生的節點標記編碼長，卻可以解決節點標記編碼字元斷句的問題，透過簡單的判斷分析，可由某一個子孫節點標記的編碼得知其父節點或更早祖先節點標記的編碼，這個較長編碼代價是值得的。

參考文獻

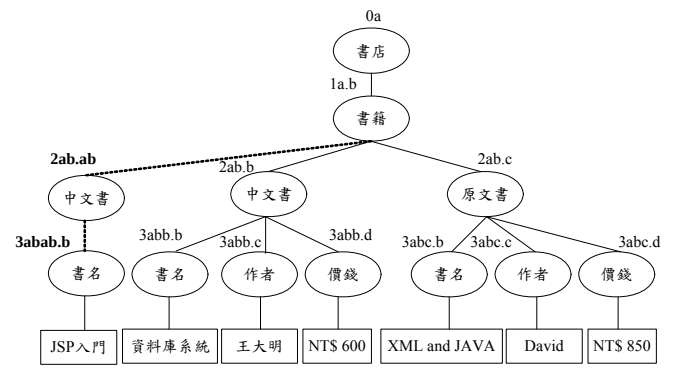
- [1] G. Amato, F. Debole, F. Rabitti, and P. Zezula, "Yet Another Path Index for XML Searching," *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries*, pp: 176-187, 2003.
- [2] J. K. Chen and J. P. Hsu, "Algorithms for Indexing Dynamic XML Data," *Proceedings of the 9th Joint Conference on Information Sciences*, 2006.
- [3] E. Cohen, H. Kaplan, and T. Milo, "Labeling dynamic XML Trees," *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp: 271-281, 2002.
- [4] F. B. Cooper, N. Sample, J. M. Franklin, R. G. Hjalton, and M. Shadmon, "A Fast Index for Semistructured Data," *Proceedings of the 27th International Conference on VLDB*, pp: 341-350, 2001.
- [5] M. Duong and Y. Zhang, "LSDX: A New Labelling Schema Dynamically Updating XML Data," *Proceedings of the 16th Australasian Conference on Database technologies*, pp: 185-193, 2005.
- [6] T. Grust, "Accelerating XPath Location Steps," *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pp: 109-120, 2002.
- [7] M. Kaelin, Database Optimization: Increase query performance with indexes and statistics, TechRepublic, http://techrepublic.com.com/5100-10878_11-5_146588.html?tag=search.
- [8] H. Kaplan, T. Milo, and R. Shabo, A Comparison of Labeling Schemes for Ancestor Queries, http://www.math.tau.ac.il/~haimk/papers/comp_arison.ps.
- [9] Q. Li and B. Moon, "Indexing and Querying XML Data for Regular Path Expressions," *Proceedings of the 27th International Conference on VLDB*, pp: 361-370, 2001.
- [10] H. Meuss and M. C. Strohmaier, "Improving Index Structures for Structured Document Retrieval," *Proceedings of the 21st BCS IRSG Colloquium on IR*, 1999.
- [11] I. Tatarinov, S. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, "Storing and Querying Ordered XML Using a Relational Database System," *Proceedings of the ACM SIGMOD international conference on Management of data*, pp: 204-215, 2005.
- [12] W. Wang, H. Jiang, H. Lu, and X. J. Yu, "PBiTree Coding and Efficient Processing of containment Joins," *Proceedings of the 19th International Conference on Data Engineering*, pp: 391-402, 2003.
- [13] X. J. Yu, D. Luo, X. Meng, and H. Lu, "Dynamically Updating XML Data: Numbering Scheme Revisited," *World Wide Web: Internet and Web Information System*, Vol. 8, Issue 1, pp: 5-26, 2005.



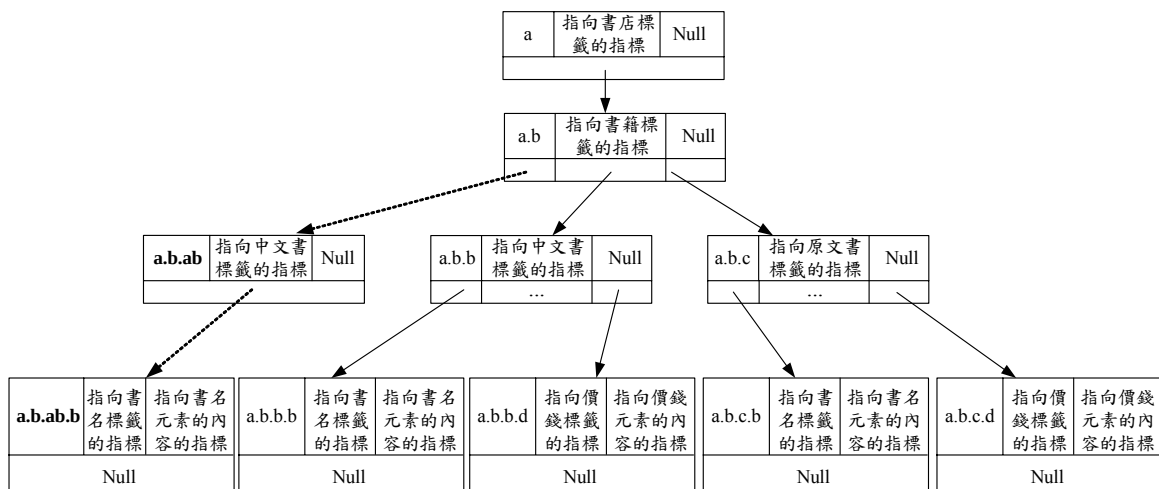
圖五：使用我們的方法對 XML 資料索引樹中節點標記編碼



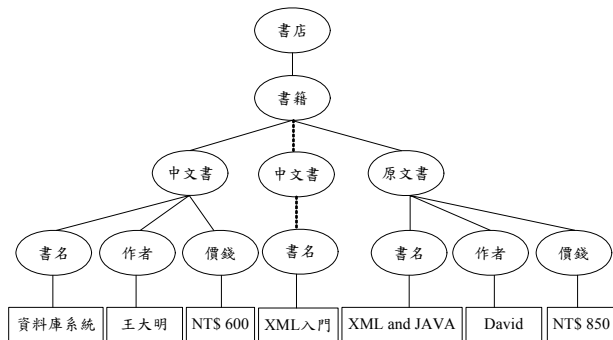
圖六(a): 在 XML 資料樹中，前無節點的位置新增一個節點



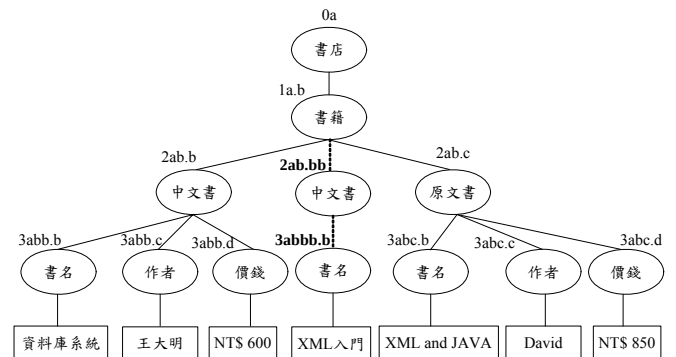
圖六(b): 索引樹新增節點時，LSDX 對節點標記編碼



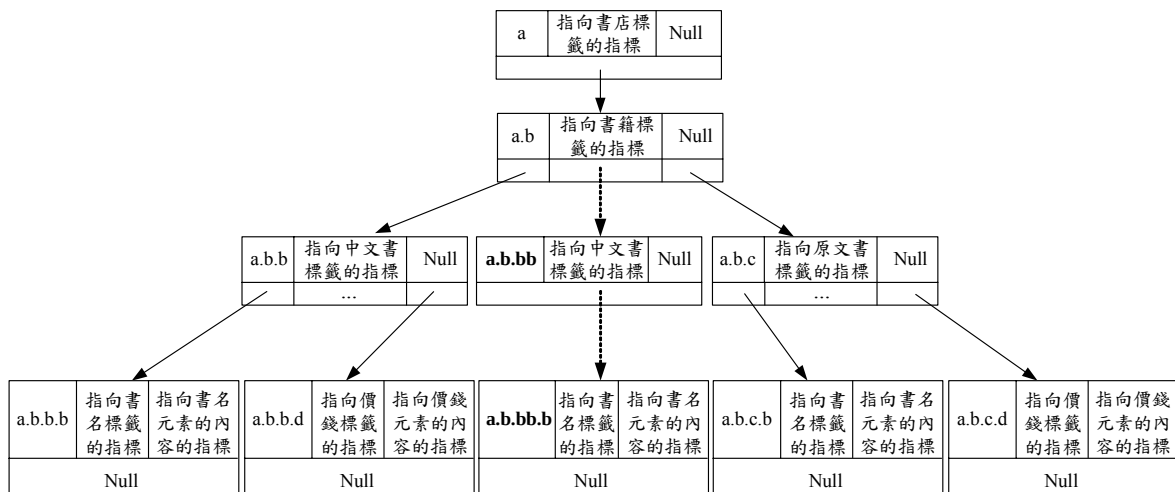
圖六(c): 索引樹新增節點時，使用我們的方法對節點標記編碼



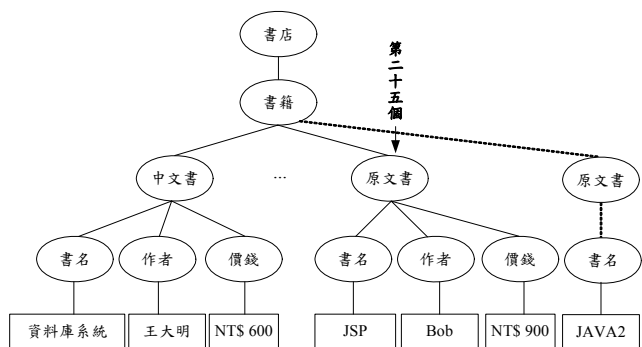
圖七(a): 在 XML 資料樹中，兩個節點之間新增一個節點



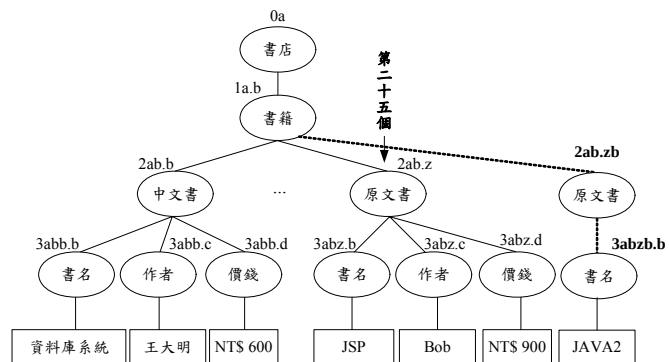
圖七(b): 索引樹新增節點時，LSDX 對節點標記編碼



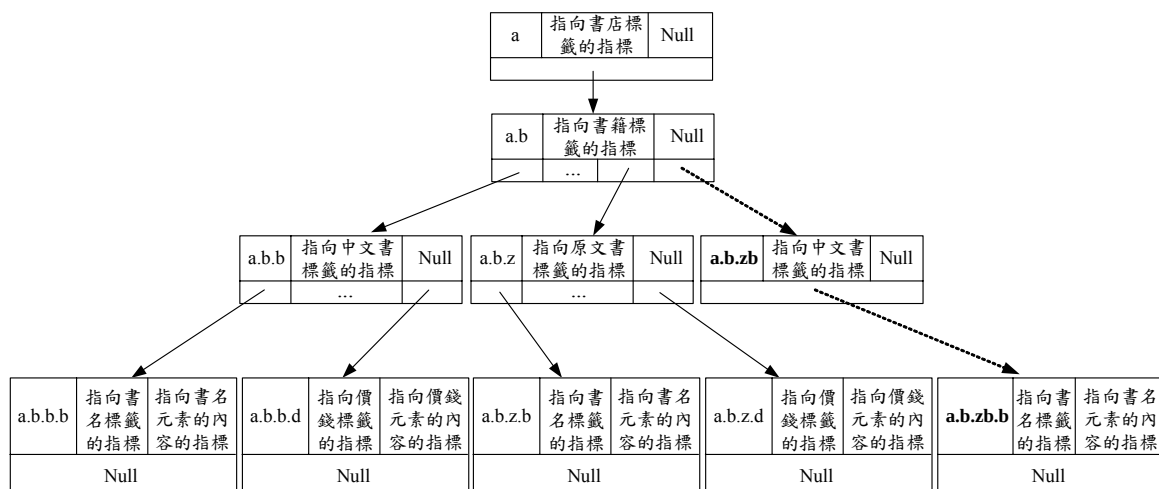
圖七(c): 索引樹新增節點時，使用我們的方法對節點標記編碼



圖八(a): 在 XML 資料樹中，前有節點且後無節點的位置新增一個節點



圖八(b): 索引樹新增節點時，LSDX 對節點標記編碼



圖八(c): 索引樹新增節點時，使用我們的方法對節點標記編碼