

使用多重索引混和鍵樹實作路由表查詢與更新

林家弘¹、許家胤¹、謝孫源^{1,2*}

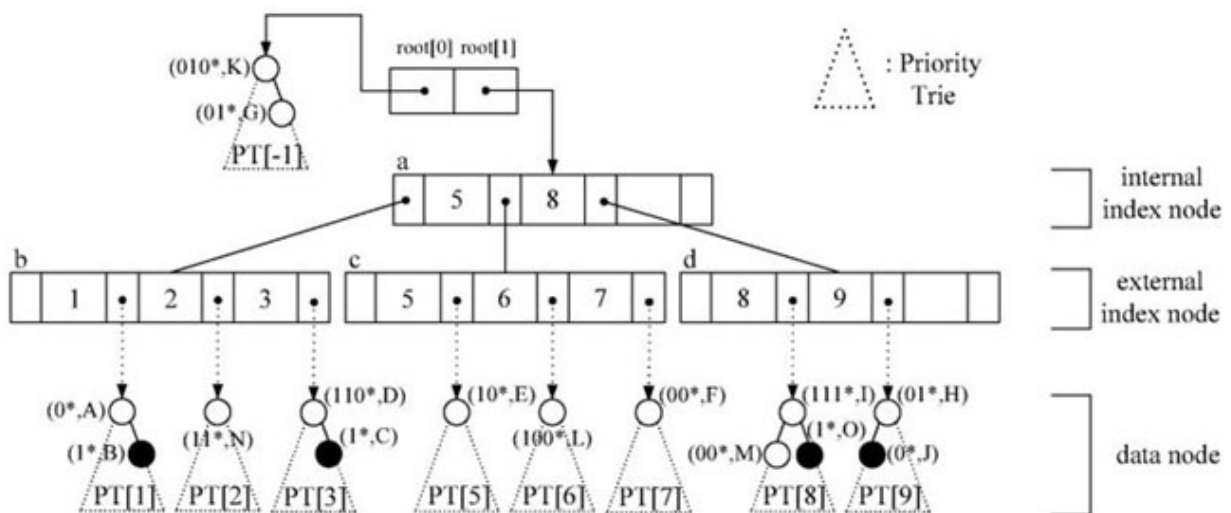
¹ 國立成功大學資訊工程學系

² 國立成功大學製造資訊與系統研究所

hsiehsy@mail.ncku.edu.tw

IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 10, pp. 2486-2498, October 2014.

由於網際網路的普及化及其應用程式常需要巨量資料的傳輸，需要高效率的路由表來幫忙資料的傳輸，以避免網路的壅塞。位址查詢會影響一個進入封包的速度，因此位址查詢是路由表格設計的一個重要角色，由於網路的千變萬化，路由表格必須是動態支援快速地新增與刪除。隨著網際網路流量的快速成長，除了高效率的路由表設計之外，我們亦需要一個有效率之網路位址查詢技術。結合 B*樹的新觀念，提出了一個動態的路由表資料結構，稱之為 Multi-Index Hybrid Trie (MIHT)，圖一為(4,4)-MIHT的架構。



圖一：(4,4)-MIHT

藉由賦予每筆前綴一個鍵值，使其能夠在平衡的搜尋樹上有效率的實施網路位址查詢。此外，由於樹高和需被比對的前綴數量同時減少了，使得動態路由表的各項運作變得更快速。我們的實驗使用實際的網際網路通訊協定第四版(IPv4)的路由資料庫。由表一至表三的實驗結果可以得知，我們所提出的資料結構不但提供更小的記憶體需求，且在路由表的查詢以及更新運作的效能表現良好。在我們的實驗中所使用的四個前綴資料庫是AS1221、AS4637、AS6447和AS65000，它們分別有407,067、219,581、417,995和406,973筆前綴資料。

表一：平均樹高與空間需求的比較

Data Structure	Avg. Tree Height				Storage (KB)			
	AS1221	AS4637	AS6447	AS65000	AS1221	AS4637	AS6447	AS65000
Binary Trie	32	32	32	32	11,519	6,731	11,870	11,513
LC-Trie	23	21	25	23	13,772	8,266	14,011	13,762
Prefix Tree	29	29	32	29	6,360	3,430	6,531	6,358
Priority Trie	28	27	32	29	6,360	3,430	6,531	6,358
DTBM	10	10	10	10	25,291	14,506	25,997	25,285
4-MPT	11	10	10	10	21,484	11,391	22,049	21,501
4-PCMST	10.08	9.72	10.13	10.04	22,172	12,463	23,051	22,043
(16,16)-MIHT	8.04	7.66	8.07	8.04	6,883	3,778	7,064	6,881
(16,16)-PMIHT	4.11	3.47	4.16	4.11	6,875	3,776	7,056	6,873

表二：查詢與記憶體需求的比較

Data Structure	Avg. Lookup Time (# of Clock Cycles)				Avg. # of Memory Accesses			
	AS1221	AS4637	AS6447	AS65000	AS1221	AS4637	AS6447	AS65000
Binary Trie	2,990	2,684	3,018	3,000	23.56	23.40	23.59	23.56
LC-Trie	3,006	2,857	3,020	3,015	9.25	9.75	9.26	9.24
Prefix Tree	3,206	2,793	3,215	3,206	21.71	21.31	21.72	21.71
Priority Trie	2,864	2,481	2,872	2,862	20.09	19.65	20.09	20.09
DTBM	2,094	1,912	2,091	2,114	8.34	8.29	8.35	8.34
4-MPT	2,878	2,509	2,850	2,866	15.69	14.71	15.61	15.70
4-PCMST	2,012	1,805	2,112	2,151	10.16	9.52	9.57	9.16
(16,16)-MIHT	2,222	1,920	2,227	2,221	9.51	9.17	9.51	9.51
(16,16)-PMIHT	1,954	1,657	1,935	1,953	8.31	8.05	8.30	8.30

表三：更新與記憶體需求的比較

Data Structure	Avg. Update Time (# of Clock Cycles)				Avg. # of Memory Accesses			
	AS1221	AS4637	AS6447	AS65000	AS1221	AS4637	AS6447	AS65000
Binary Trie	4,138	4,021	4,216	4,181	26.55	26.70	26.59	26.54
Prefix Tree	2,445	2,395	2,426	2,438	20.87	20.33	20.77	20.85
Priority Trie	3,834	3,830	3,831	3,849	21.42	21.00	21.43	21.42
DTBM	2,755	2,705	2,749	2,754	8.90	8.91	8.91	8.90
4-MPT	2,144	2,419	2,178	2,277	6.24	6.26	6.21	6.24
4-PCMST	1,345	1,517	1,079	1,153	4.04	4.05	3.02	3.24
(16,16)-MIHT	1,907	1,780	1,853	1,900	8.90	8.70	8.90	8.90
(16,16)-PMIHT	1,679	1,693	1,622	1,679	7.56	7.46	7.56	7.57