**高效能路由器的架構與設計**

作業二

P76121437

李漢威

# A IP lookup methods, Trie

透過執行範例程式binary\_trie.c以及4bit\_trie.c各十次，取平均之後得到的值，當中有小小更改一些地方，shuffle這個function在binary\_trie無法執行，目前是不知道問題所以就沒有執行這個function。

以下是binary\_trie的分析

|  |  |  |
| --- | --- | --- |
| Metric | Value(Average) | Unit |
| Build time | 1578.2 | Seconds/CPU cycle |
| Update time | 277 | Seconds/CPU cycle |
| Query time | 442.1 | Seconds/CPU cycle |
| Memory usage | 37362 | KB |

以下是4bit\_trie的分析

|  |  |  |
| --- | --- | --- |
| Metric | Value(Average) | Unit |
| Build time | 245.9 | Seconds/CPU cycle |
| Update time | 365 | Seconds/CPU cycle |
| Query time | 37 | Seconds/CPU cycle |
| Memory usage | 69594 | KB |

下圖是兩個不同演算法的搜尋時間比較，可以看到4bit\_trie的搜尋時間大幅縮短，大多在0~300cycle就可以搜尋到目標，而binary\_trie的搜尋時間不等，主要會需要200~1000cycle才可以搜尋到目標。

一張含有 文字, 圖表, 行, 繪圖 的圖片

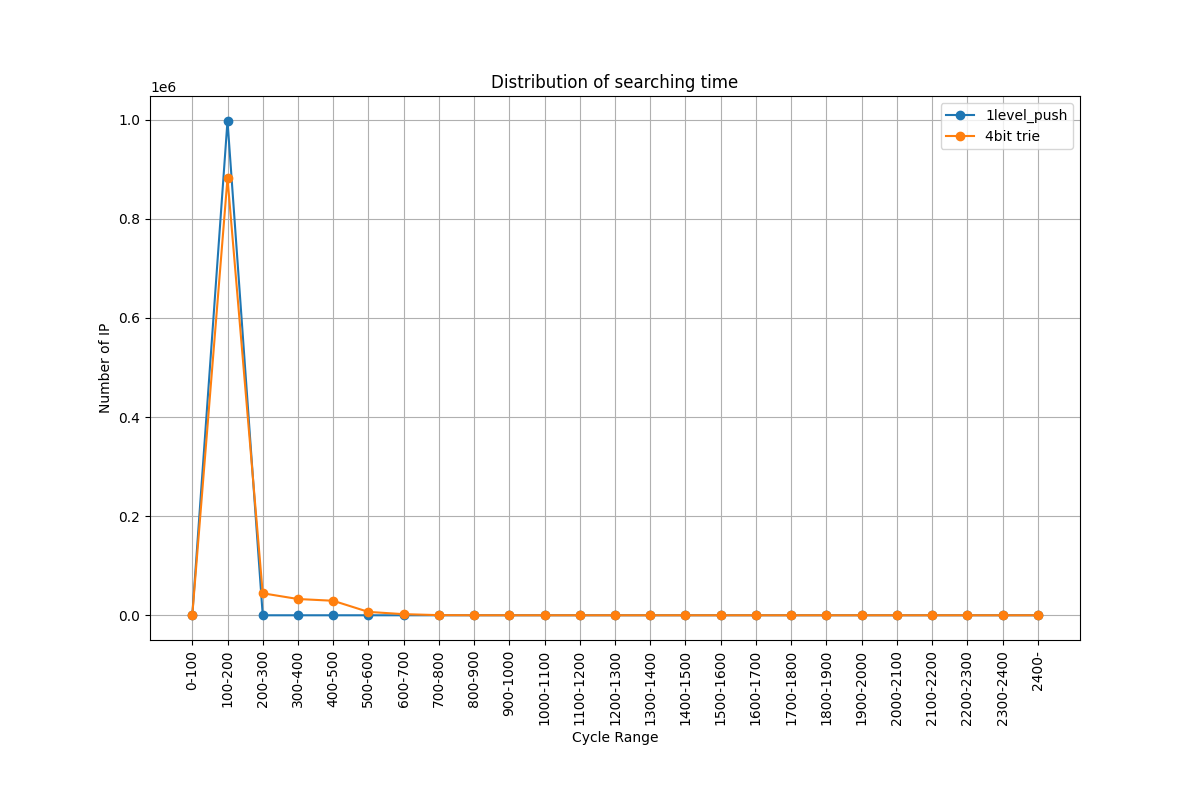
自動產生的描述

# B. Analysis of Leaf Push Algorithm for Memory Optimization

執行1level\_push.c十次以後，取平均之後得到的值。

|  |  |  |
| --- | --- | --- |
| Metric | Value | Unit |
| Before one level push prefix nodes | 997952 |  |
| After one level push prefix nodes | 959645 |  |
| Average Build time | 988.4 | Seconds/CPU cycle |
| Average Insert time | 810.8 | Seconds/CPU cycle |
| Average Search time | 12 | Seconds/CPU cycle |
| Nodes in binary trie | 3258148 |  |
| Total Memory requirement | 26935 | KB |

以下是搜尋的cycles數以及IP數目的折線圖



可以看到與4bit\_trie相比，幾乎都落在100~200cycle之間，搜尋的速度是足夠快速的，但build time 及 Insert time都比4bit\_trie還要長。

# C. Tree bitmap

根據已經建好的binary\_trie去建立其相對應的bitmap，亦是執行十次取平均得到的結果，以下是其分析。

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | 4bit\_trie | Binary\_trie | Tree bitmap |
| Average build time | 245.9 | 1578.2 | 99.3 |
| Average Insert time | 365 | 277 | x |
| Average Search time | 37 | 442.1 | 20 |
| Number of nodes | 3188276 | 1192284 | 13 |
| Total memory requirement | 37362 | 75682 | 1 |

其中bitmap的nodes 及 memory不確定算法是不是對的，13也許是13個bitmap(內含7個點)，故可能為91個點，memory就會是10KB。

以下是其搜尋的cycles數以及IP數目的折線圖

一張含有 文字, 行, 圖表, 繪圖 的圖片

自動產生的描述

其實跟1level\_push得到的結果是相似的，都是在100~200cycle之間就能找到目標。

# D. Conclusion

* Binary Trie :
  + 優點 : Insert time是當中較快速的
  + 缺點 : 建置的時間及查詢的時間都很長
* 4bit Trie : 適合需要頻繁搜尋
  + 優點 : 建置時間及搜尋時間都很短
  + 缺點 : memory 使用的空間較大
* 1level Push : 適合需要頻繁搜尋且不常更新
  + 優點 : 查詢時間最短
  + 缺點 : Insert time需要較長時間
* Bitmap : 總體的表現最好，但是建置時間會需要將前置的Trie先建立好，如果每次有增刪都需要將Trie的增刪時間納入考量。