

# 論文題目Survey方法

1. 首先定好一個研究主題 (e.g., Tree結構在儲存裝置的應用)
2. 針對這個主題搜尋所有相關的論文閱讀 (亂槍打鳥)
3. 當搜尋並看過一定數量的論文, 根據論文的應用情境整理成一個表格
  - 分為兩個維度, 並將讀過的論文map到其中一個欄位中

研究主題 - 以資料結構在儲存裝置的應用為例

不同的資料結構

|                   | Bε tree   | B+ tree | Hash | LSM tree |
|-------------------|-----------|---------|------|----------|
| SSD               | 6,9       |         |      | 9        |
| HDD               | 1,2,3,4,5 |         |      |          |
| SMR SSHD          |           | 10      |      |          |
| GPU               |           | 11      |      |          |
| Persistent Memory |           | 12,13   | 15   |          |
| Transaction       |           | 14      |      |          |
| Network           | 7,8       |         |      |          |

不同的應用情境

| 編號 |   | 備註  |
|----|---|---|
| 1  | An Introduction to Bε-trees and Write-Optimization                                      |   |
| 2  | BetrFS: A Right-Optimized Write-Optimized File System                                   |   |
| 3  | Optimizing Every Operation in a Write-Optimized File System                             | BetrFS於傳統HDD儲存裝置的應用                                 |
| 4  | File Systems Fated for Senescence? Nonsense, Says Science!                              | 探討Bε tree於HDD裝置上的aging程度                            |
| 5  | How to Copy Files   | 以Bε-tree為基礎建立支援快速複製(nimble clone)的資料結構              |
| 6  | Tucana: Design and Implementation of a Fast   | 以減少CPU cycle為切入點, 將Bε-tree套用到                       |
| 7  | Tracking Network Events with Write-Optimized Data Structures                            |   |
| 8  | Lazy Analytics: Let Other Queries Do the Work For You                                   | 主要是基於遠端資料庫的特性開發: 要求的結果不必於提出要求的當下立刻回應; 然而新增的資料必須快速寫入 |
| 9  | SplinterDB: Closing the Bandwidth Gap for NVMe Key-Value Stores                         | 探討以Bε-tree輔助LSM-tree以加速compaction速度的可行性             |
| 10 | Enabling a B+-tree-based Data Management Scheme for Key-value Store over SMR-based SSHD |   |
| 11 | Harmonia: A High Throughput B+-tree for GPUs  |   |
| 12 | Endurable Transient Inconsistency in Byte-Addressable Persistent B+-Tree                | 探討atomic I/O的大小對B+-tree的移植(從硬碟到PM)有何影響              |

# 論文題目Survey方法

4. 根據建好的表, 利用三種不同的觀察法, 尋找可能的題目方向

1. 沒人做過的方向
2. 不同架構之間的異同點
3. 同架構之間的異同點

不同的資料結構

|                   | Bε tree   | B+ tree | Hash | LSM tree |
|-------------------|-----------|---------|------|----------|
| SSD               | 6,9       |         |      | 9        |
| HDD               | 1,2,3,4,5 |         |      |          |
| SMR SSHD          |           | 10      |      |          |
| GPU               |           | 11      |      |          |
| Persistent Memory |           | 12,13   | 15   |          |
| Transaction       |           | 14      |      |          |
| Network           | 7,8       |         |      |          |

不同的應用情境

同樣使用Bε tree在SSD,HDD上的異同點?

LSM Tree在HDD上為何沒有相關的研究?  
1. 不適合?  
2. 有可能性但還沒有人探討?

在Persistent Memory上使用B+ tree和Hash之見的異同點?