
PostgreSQL as a Log database

古永忠 博士
ycku@postgresql.tw



古永忠 (ycku@postgresql.tw)

- 國立臺灣大學
資訊工程研究所 博士
- 資料庫系統、分散式系統
- 國泰人壽 系統資訊部 副理
 - 資料庫架構設計
- 資訊工業策進會 工程師
- 行政院主計處電子處理資料中心
研究員
- PostgreSQL台灣使用者社群
召集人
 - 正體中文使用手冊



關於社群

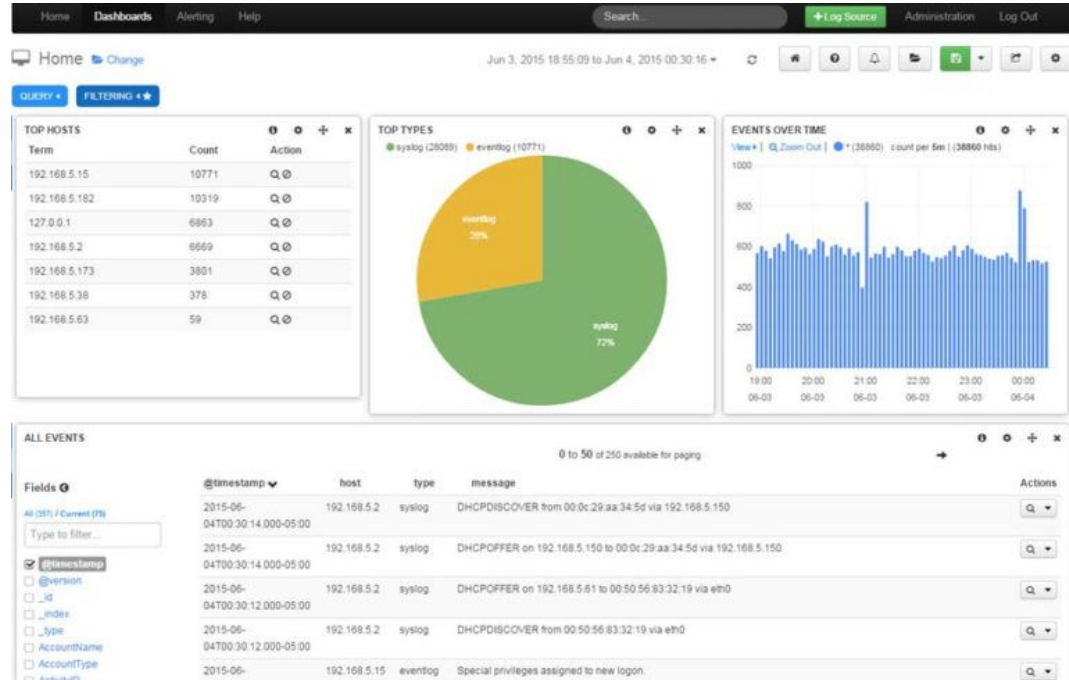
1. DBA - 你不是一個人
2. Developer - 資料庫突顯你的技術能力
3. Manager - 理解資料工程師的價值
4. Business - 資料庫是營利工具

- 持續與日本 PostgreSQL 社群合作
- 舉辦實體社群活動
- 線上協作開源專案



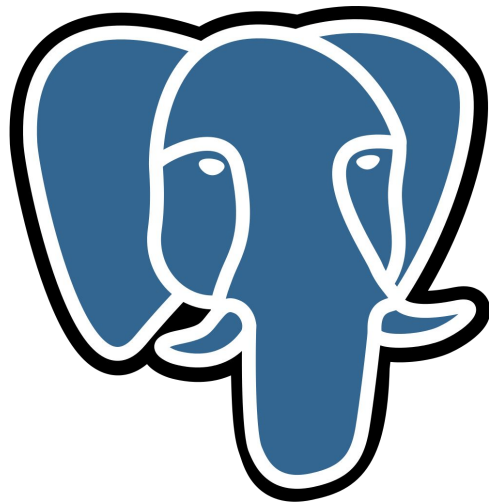
Key features for Log database

1. Response Time
2. Consumed Space
3. Materialized View
4. Partition table
5. BRIN index
6. VACUUM & REINDEX
7. Unlogged Table



Why use PostgreSQL

- 多數程式人員僅熟悉 SQL 開發邏輯。
 - 降低人員門檻。
- PostgreSQL 泛用性高。
- 對 PostgreSQL 的壓力測試。
 - Log 系統的資料量大。
 - 但可容許資料遺失。



Response Time

- 搞清楚你為誰記錄？
- 從需要產生的圖表開始規畫
 - X 無腦每秒查詢
 - X 無腦每秒插入
 - X 無腦 SELECT 全部資料
- 從人員需求定義
 - 總經理
 - 監控人員
 - 程式設計師
- 請程式設計人員以多個短查詢替代長查詢
 - DBA 需要理解組織內的程式文化



Consumed Space

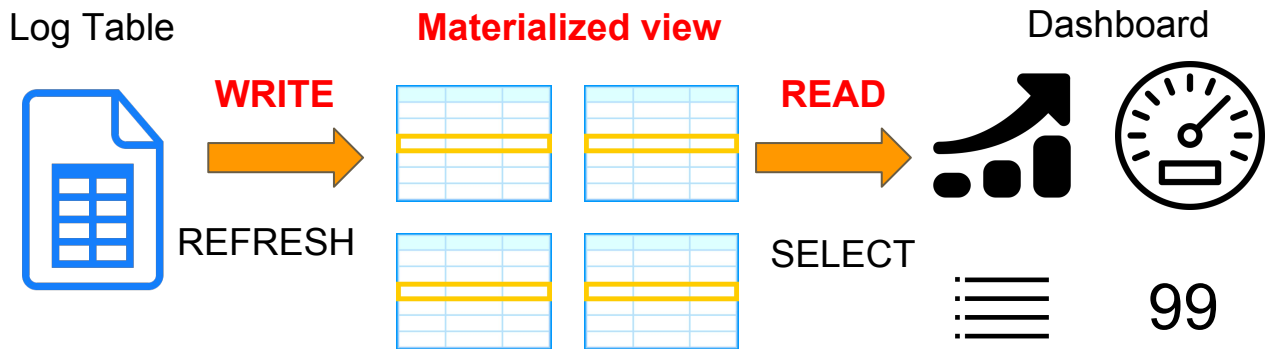
- 空間不是最重要的事
- 多份中繼資料有助於提升 Response Time
 - 異常反應時間 v.s. 額外空間支出？
- 索引規畫
 - 短查詢有利於穩定查詢計畫(確定會用到索引)
 - 讀的資料表 + 寫的資料表
- 快取設計
 - materialized view
- 批次程序
 - 長期 Time-series 趨勢(快取)
 - 非同步作業



Materialized View

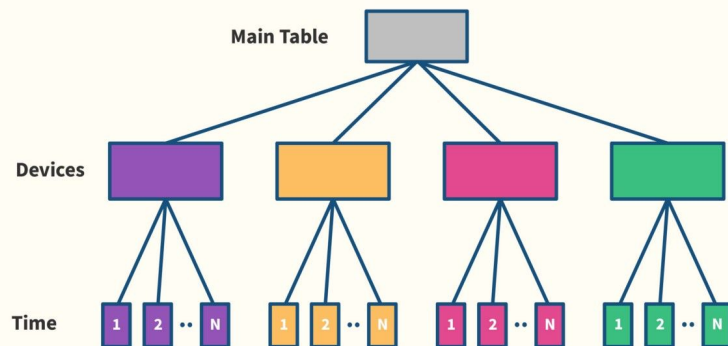
- 即時計算太慢了
- 同時間在同一資料表，記錄資料和產生報表，一定會影響效能。

- 非同步作業
- 針對不同報表產生
- 大量 INSERT 之後再 REFRESH(或重建)
- 原則是讓讀取末端計算量最小化



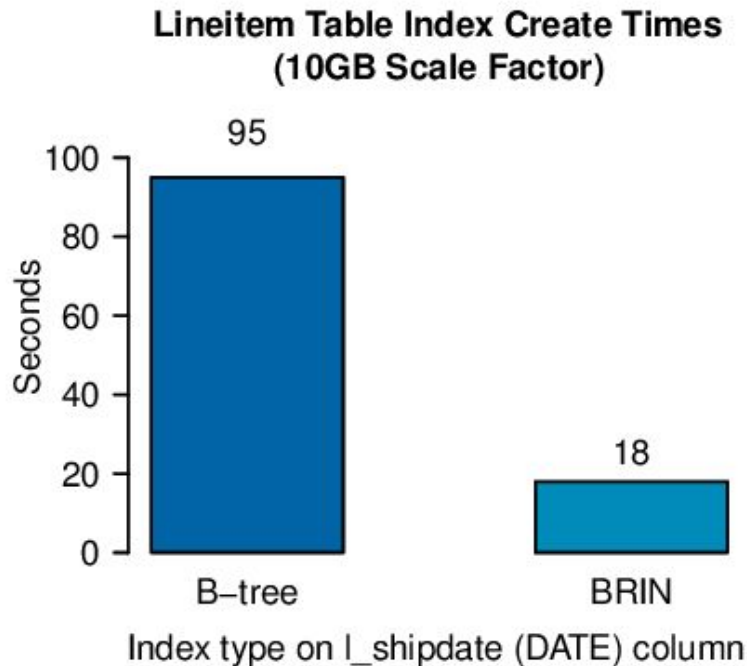
Partition table

- 最好升級至 PostgreSQL 11
- 大型 Table 分割查詢加速
- TRUNCATE or DROP
TABLE 比 DELETE 快！
- VACUUM or REINDEX 快！



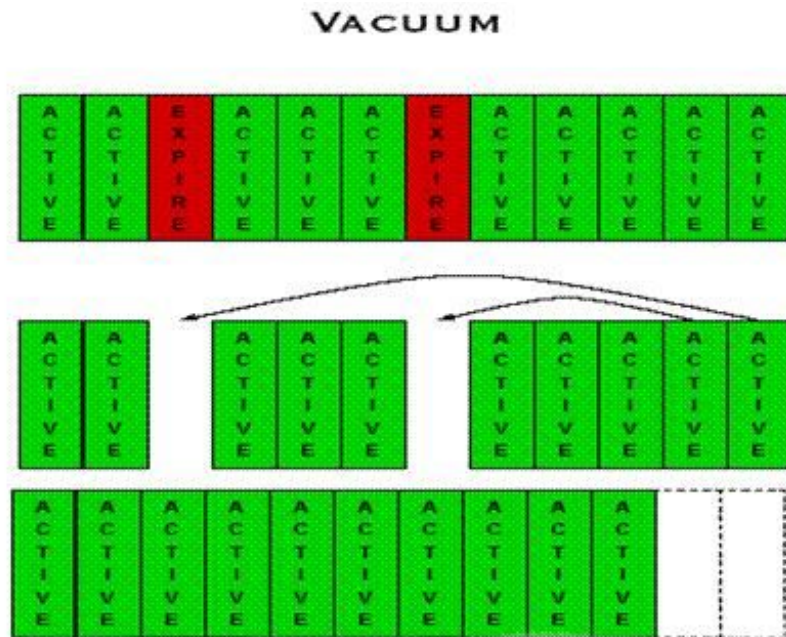
BRIN index

- Log 資料有很多是範圍式查詢
- Log 資料索引都比資料本身大
- 減輕空間負擔, 查詢加速, 維護時間短



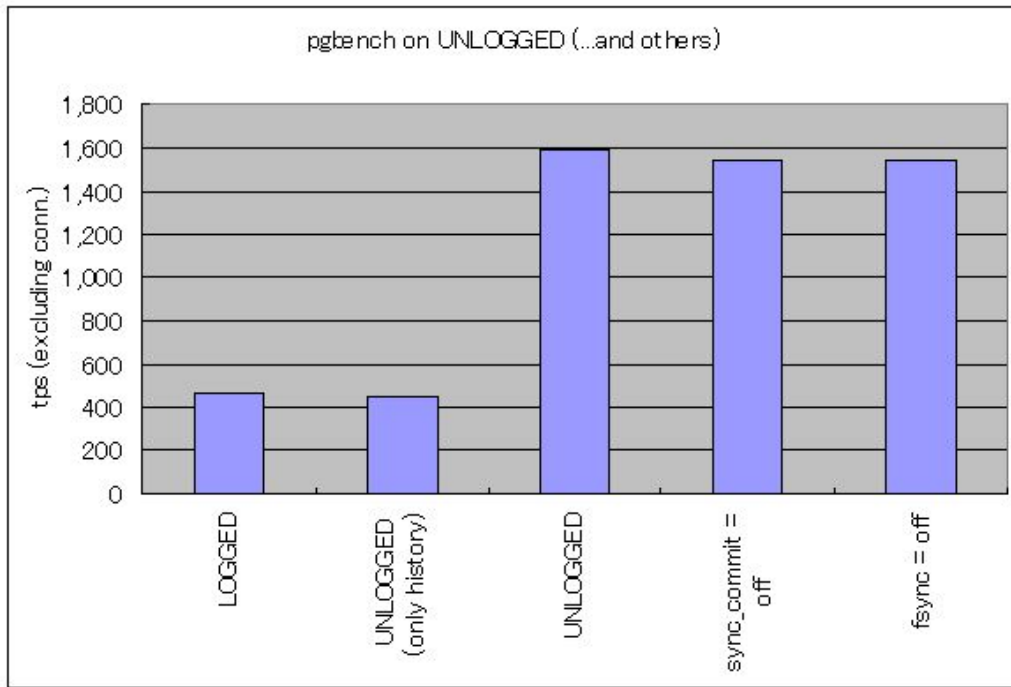
VACUUM & REINDEX

- 需要比一般情況應該更經常做資料庫清理和維護
 - 資料變化較大
 - 回收部份空間
 - 重新排列實體儲存次序
 - autovacuum
- 主機資源要加上每日維護的負載
- REINDEX 不如重新 CREATE INDEX



Unlogged Table

- LOG 資料可允許意外
- 速度倍增
- 不需要調整資料庫結構

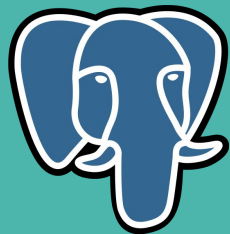


其他 & 未來

- Tablespace 如果處理接近滿載的狀態時, Postgres 會頻繁地發出 "recovery mode" 警告訊息
 - 剩餘空間至少 20% 以上
- Data API 的設計長遠上是需要的。
 - 資料庫專注處理資料邏輯的部分。
 - Data API 處理業務邏輯的部分。



Thanks



postgresql.tw