桌面搜尋與平行化 PageRank 運算

Desktop Search and Parallelized PageRank Calculation

指導教授：涂嘉恆、黃敬群
專題成員：侯志勳
開發工具：GCC、Valgrind、Sanitizer
測試環境：Linux Ubuntu 22.04

一、簡介：

早在 2000 年 Google 就有「Google 桌面」（Google Desktop），一個提供本地端檔案資料搜尋的服務軟體（Desktop Search），如同如今 Google 在雲端提供的搜尋引擎服務，與其類似的本地端的桌面搜尋，可以對於本地端的桌面檔案進行監控與檢索，並且依照特定的優先順序對於檔案作出排序。

然而 Google 桌面並不是開源軟體，並且在 2011 年就已經關閉服務。因此，我們終極的目標是做出一個開源的，並且可運行於 Linux 的背景，並且針對多核，盡可能達到低功耗、高效能，不影響使用者進行中的工作 — 類似於 Google 桌面的精簡版的實作。

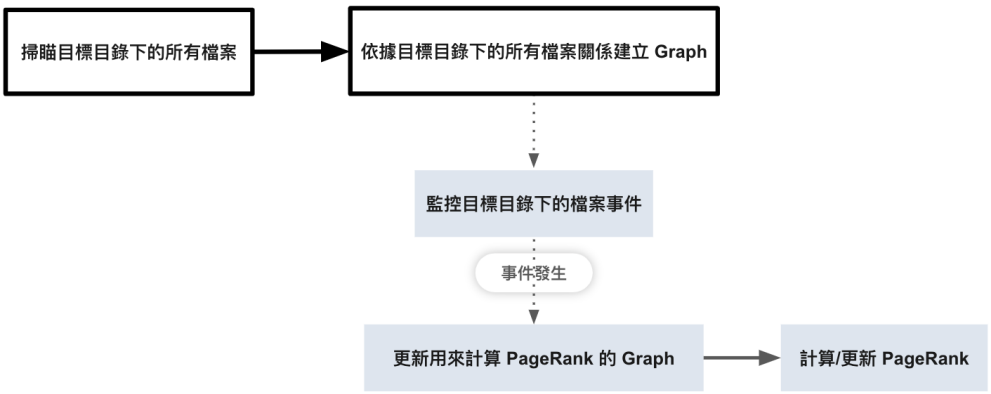
依照 Google 的網頁排序，PageRank 作為主要知名的演算法之一，可以依照檔案連結分析，給予其不同的權重值，進行排序。我們沿用 PageRank 演算法排序，作為在本地端的桌面檔案搜尋的排序參考。

所以要針對多核維持高效，如何有效率的計算 PageRank 便是一大課題，參考 Hadoops 於對於 PageRank 演算法的實作，用到了 Map-Reduce 的方法以及平行化的運算，加速計算 PageRank 的速度，所以我們也效法之，設法對 PageRank 進行平行化的實作。其中探討到 Multithreading，以 POSIX Threads 實作並且針對 PageRank 演算法的平行、並行實作的可能性作進一步的探討。

二、方法：

首先需要掃描目標目錄，藉由掃瞄完該目錄下的所有檔案，建立一個 Graph 用來作為計算 PageRank 所需的資料。之後一個程式在背景監視目標目錄，只要在目標目錄下，任何發生檔案事件，例如檔案的增減，就要更新用來計算 PageRank 的 Graph 的資料，同時發送通知 PageRank 需要重新計算，準備重新計算新的 PageRank，最後將 Graph 的資料做分割，平行化計算 PageRank 得出最後結果。

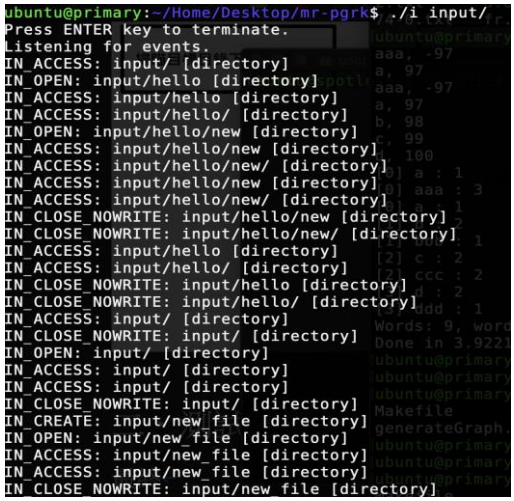
以下為架構圖：



圖一：系統運算流程圖

二、測試結果：

掃描目錄與監控目錄下的檔案與 PageRank 的計算分開實作。圖二是利用 Epoll API 與 nftw 進行實作達到掃描、監控目錄下的檔案事件。



圖二：檔案事件監控

開發紀錄：https://hackmd.io/@qJEZpNvuSHe0kzJAmMSeQg/ByXZQ1nMi