Chapter 14-5 Log-structured File Systems

資工三穆冠蓁 4107056007

* Introduction

1. Motivation : 由於記憶體越來越大，所以可以將更多的data cache到RAM中，但卻只能減少disk read次數，卻很難減少disk write次數。此外，循序存取及隨機存取的效能差異很大。

2. Introduction : 所以我們應該著重在提升write disk的效能，並多利用循序存取的好處

3. Solution : Log-structured file system (LFS)

(1) 如何將所以write變成sequential write?

(2) 先將所有的write寫入memory

(3) 將要寫入的檔案用循序的方式(log-like structure)寫入硬碟

- circular buffer(不要overwrite)

- 將多個random write合併為一個大的sequential write

* Writing To Disk Sequentially

1. Writing To Disk Sequentially

(1) 為了要增加write的效能，循序寫入硬碟(不應該用一次寫一個block的方式，無法達到循序寫入的最高效能)

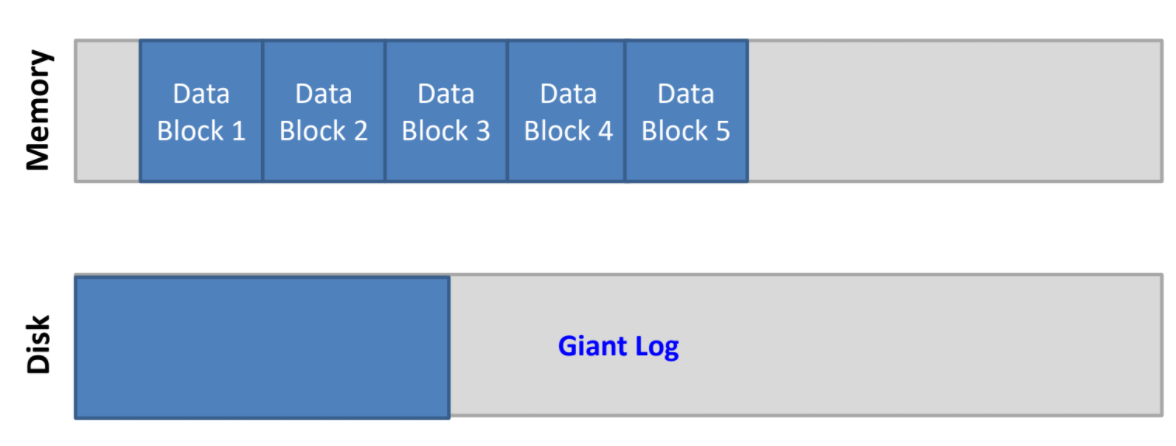
(2) You must issue one large contiguous write to disks

2. Buffering Writes : 先將資料buffer在memory，等存成一個segment再寫入disk

3. Log-Structured File System (LFS)

(1) 先將資料buffer在memory，等存成一個segment再寫入disk

(2) treat the disk as a log(don’t overwrite)，下次寫入以append的方式寫入



* Problem: Finding Inodes

1. 在LFS中，inodes散步在硬碟中(floating inodes)，因此要找到inodes很難

2. Solution : through Indirection

* A New Problem: Garbage Collection

1. Garbage : LFS不停的寫入新的資料到檔案中心的位置，因此會有很多舊的data or metadata(garbage)散步在disk中

2. 當disk滿的時候，就已circular buffer的方式，做garbage collection

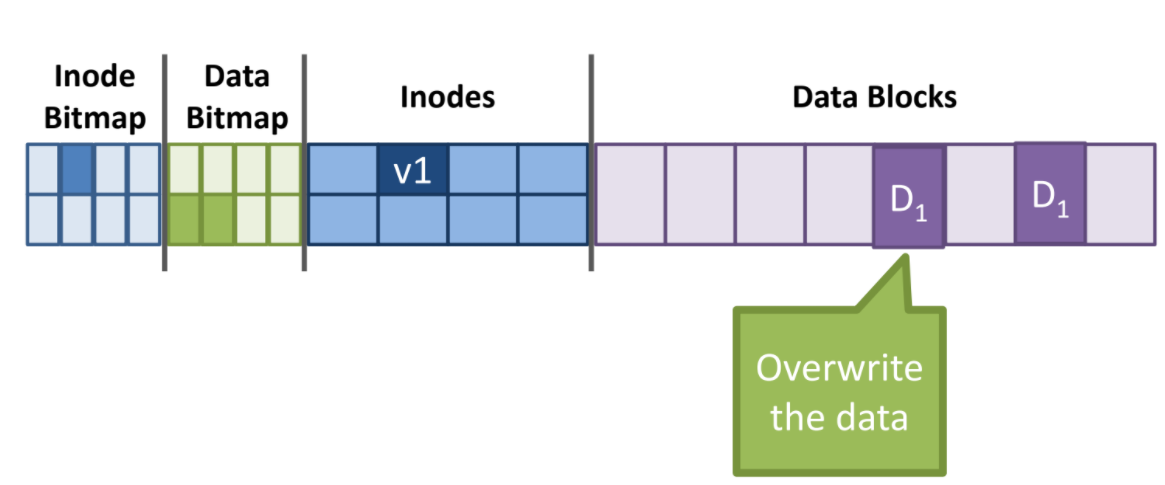
* Summary

1. LFS的觀念，廣泛用在現在的FS中(no overwrite)

2. Copy-on-Write File Systems :

(1) 將要更新的資料，直接寫到disk上空白的地方，而不是直接overwrite

(2) 避免data corruption(保有old data)



3. Versioning File Systems

(1) Old versions of files may be useful!

(2) 如果使用者有需要，可以roll-back來回復到舊版本