Paper Title： File System Design for an NFS File Server Appliance

Student Name：康智詠

Department/Year：資工四甲

Student ID：406261523

Date: 2020/10/10

Q1. What is the problem the authors are trying to solve?

主要在介紹WAFL(Write Anywhere File Layout) implement Snapshots的演算法以及資料結構

Q2. What other approaches or solutions existed at the time that this work was done?

Fast File System and TransArc’s Episode file system

Q3. What was wrong with the other approaches or solutions?

Episode’s fileset clones 需要耗費大量的硬碟I/O並且需要很多的磁碟空間

Q4. What is the authors' approach or solution?

1. 特殊的Snapshots稱作consistency point
2. 根據不同大小的檔案使用不同的鏈結方式指到blocks。檔案小於64 KB使用16個block pointer指向data blocks ;檔案小於64 MB, Inodes point to indirect blocks which point to actual file data;

檔案大於64 MB, Inode point to doubly indirect blocks.

1. Meta-Data lives in files. Use inode file which contains the inodes for the file system、Block-map file identifies free blocks、inode-map file identifies free inodes
2. Tree of blocks 的結構保存inode file.
3. 使用NVRAM保存NFS的請求

Q5. Why is it better than the other approaches or solutions?

1. 根據不同大小的檔案選擇不同方式記錄檔案資料可以減少浪費的資源
2. Write-anywhere 使得WAFL可以有效率操作RAID規劃多次寫入一樣的RAID stripe
3. WAFL創造Snapshot快速且disk I/O很快
4. WAFL回復NFS的請求快速
5. WAFL 可以在磁碟的任何地方寫入任何的file system block
6. WAFL 可以以任何排列方式將blocks寫入磁碟
7. WAFL可以在一次single write episode分配許多磁碟空間給許多NFS操作

Q6. How does it perform?

1. **File System Consistency**

WAFL每幾秒創造一種特別的Snapshot稱為consistency point.

Consistency point並沒有特別標記名字也並不需要通過訪問NFS.

1. **Non-Volatile RAM**

WAFL使用non-volatile RAM來保存NFS的要求。當不明的終止發生後，WAFL回復RAM中所保存的要求，以防止漏掉任何的要求

**3. The Block-Map File**

WAFL的每個4 KB disk block中的 block-map file包含32-bit entry 來記錄Snapshot的結果

當block被active file system reference時bit設成0

當block有被first Snapshot reference時bit設成1

**4. Creating a Snapshot**

為了保持Snapshot data資料的一致性，會將在cacah中的dirty data標記”IN\_SNAPSHOT”，如果資料並沒有被標記為”IN\_SNAPSHOT”則必定不會在磁碟中更新。

為了避免把NFS的請求拒之門外，WAFL必須快速更新有被標記為”IN\_SNAPSHOT”的資料，因此WAFL制定了以下步驟：

1. 分配磁碟空間給所有有被標記為”IN\_SNAPSHOT”的檔案，之後WAFL將inode更新的資訊從inode cache複製到inode file disk buffer並把in-core中的inode的“IN\_SNAPSHOT”清除
2. 更新block-map file
3. 將在cache中所有被標記為”IN\_SNAPSHOT”的disk buffer寫入在磁碟中被分配到的空間，一旦寫完，WAFL繼續回應NFS的要求
4. 複製root inode給新的Snapshot使用，並將root inode的”IN\_SNAPSHOT”清除

Q7. Why is this work important?

設計一套可以穩定接收並回應請求、快速寫入磁碟、有效紀錄資料的Server

Q8. Can any improvement be done?

WAFL只能接收來自NFS client的請求，如果能夠處裡來自其他client的請求會更好