Paper Title： Deciding when to forget in the Elephant file system

Student Name：康智詠

Department/Year：資工四甲

Student ID：406261523

Date: 2020/09/30 (2020/10/17補繳)

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

Elephant file system可以自動保存所有使用者重要的檔案版本，主要在闡述它的設計理念、實作、行為

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

第一個提倡版本檔案的file system為Cedar file system 來自 Xerox PARC。寫入檔案會產生新的版本檔案且檔案名稱包含版本的編號，相似的概念也有在

RSX, VMS, and TOPS-10/-20 operating systems from Digital中找到。

近年來檔案系統對於版本控制有不同的解決方式，AFS、Plan-9、WAFL有整個檔案系統高效率的檢查點，可以幫助備份或者提供使用者對檔案意外刪除以及意外覆寫的保護。這些作業系統的不同點在於多久產生一次檢查點。

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

有關於檢查點的檔案系統主要有兩個限制。第一，所有檔案的檢查點都相同，然而檔案有不同的使用情況以及保留需求。將所有的檔案保留所有版本是不適當的，僅對某些檔案而言保存版本是很重要的。不幸的是，我們沒辦法用file system-grain解決這個問題。第二，沒辦法回溯到前一個檢查點。

Q4. What is the authors' approach or solution?

1. 將檔案大致分為read-only, derived, cached, temporary, user-modified讓使用者根據以上檔案類型選擇不同的保留政策
2. 將保留政策主要分為keep one, keep all, keep safe, keep landmarks。Keep one只保存一個版本的檔案，只要檔案被刪除就會馬上被移除； Keep all會保存所有版本的歷史檔案；Keep safe會保存一段時間內檔案系統的所有操作；keep landmarks可以長期保存歷史檔案，只有被設為座標會被保留。而當儲存空間不夠時，沒有被設為座標的歷史版本將會被清除

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

針對不同檔案類型讓使用者選擇認為適合的保留政策使得重要的檔案歷史版本可以被保留，且資源可以被妥善利用

Q6. How does it perform?

**Meta data**

1. 為了保存inumber重要的命名屬性，Elephant重新定義inumber以索引imap
2. 會為每個檔案準備一個imap裡面有entry，每個entry有16 bytes，用來儲存檔案的inode或inode log pointer, temperature, policy, policyGroup。Imap儲存在磁碟以及記憶體中快取，而它的磁碟中會被放在固定的位置
3. Meta-data pointer有兩種，type以及address。Type用來表示是否檔案被存在inode或者inode log
4. Address對inode而言，是在inode file中的移動量；而對inode logs而言，是檔案的inode log磁碟block編號
5. Tempture會隨著版本數量提升而增加，讓系統重新檢視並考慮是否要清除掉此版本檔案
6. Policy是檔案的保留政策
7. policyGroup儲存下一個一樣policy的inumber
8. Elephant將沒有歷史版本的inode儲存在inode file
9. Inode log在磁碟block的granularity中被分配空間以及命名，因此比單個inode還要消耗更多的空間

**Directories**

1. Directories 將名字映射到inumber
2. 每個directory entry儲存名稱的創造時間，如果已經刪除，則儲存刪除時間
3. 在一開始，directory儲存單個inode。如果directory累積足夠多被刪除的名稱，則會有第二個inode來分配給刪除的名稱。必要的時候，系統會定期將被刪除的名稱從現存inode移動到歷史inode，藉此壓縮現存的inode

**System interface**

Elephant允許使用者可以在呈現檔案系統的任何路徑名稱上添加時間標籤，Elephant就可以依據檔案存在的特定時間來取得特定的版本。以下七個system call被添加到Elephant當中

1. setCurrentEpoch用來更改調用進程的當前epoch
2. getHistory取得inodes在檔案的inode log中的摘要
3. setLandmark指定特定的版本檔案成為座標，使此檔案不會被刪除
4. unsetLandmark移除當前的座標
5. setPolicy指派檔案保留政策
6. groupFiles將兩個檔案制定相同的政策，加至相同團體中
7. ungroupFile將檔案從當前團體中移除

**Storage reclamation**

清理儲存空間交給系統清潔者處理。清潔者有權力且信任客製化kernel interface的使用者模式進程跟Elephant file system進行互動，以下為清潔者的kernel interface

1. mapImap用來將所給的路徑之inode map映射到virtual address space，並茄返回mntPt，檔案系統的mountpoint id
2. lockFile和unlockFile用來鎖住以及解鎖檔案。當清潔者鎖住時，其他的進程不被同意操作當前檔案
3. 清潔者使用readBlock和writeBlock直接讀取和同步寫入inode log
4. freeBlock用來清除特定的block

當清潔者在後台中執行時，會掃描在記憶體中被快取temperature高的imap。當清潔者選擇檔案會呼叫lockFile，接著使用readBlcok來將檔案的inode log讀進記憶體中。接著掃描log來建造block的二元樹為檔案分配空間。

**Registering an application policy**

應用程式呼叫registerPolicy藉由路徑名稱註冊特定的程序當作新的政策，或者呼叫unregisterPolicy來移除政策。Kernel指派獨特的policyID給所有的政策並且永久儲存。

Q7. Why is this work important?

Elephant file system可以讓使用者指定的文件自定義保留的策略並且可以回溯到歷史版本

Q8. Can any improvement be done?

1. 使用者僅能一次取得一個版本的檔案，希望可以增加兩個操作，匯出以及匯入歷史檔案。
2. 希望可以合併多個複製檔的歷史檔案，因為多版本共存使被合併的歷史可能有間隔
3. 修訂版本控制系統通常將檔案歷史視為rooted acyclic graph，允許分支或合併點。問題出在命名上，Elephant file system因為使用時間去命名，所以必須擴增或取代才可以解決多個版本共存造成的歧異。策略是為共存的歷史藉由分支增加一個標籤。當需要為共存的歷史檔案消除歧異時可以藉由pathname-time-tag的方式來命名