國立交通大學

資訊科學與工程研究所

碩 士 論 文

固 態 硬 碟 的 效 能 評 測 與 分 析 方 法

SSD Benchmarks and Performance Analysis

研 究 生：楊明毅

指導教授：張立平　教授

中 華 民 國　九 十 八　年 七 月

固態硬碟的效能評測與分析方法
SSD Benchmarks and Performance Analysis

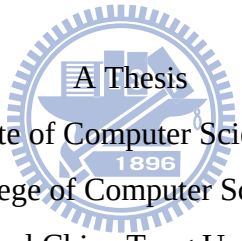
研 究 生：楊明毅　　Student：Ming-Yi Yang

指導教授：張立平　　Advisor：Li-Pin Chang

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文



A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

July 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年七月

固 態 硬 碟 的 效 能 評 測 與 分 析

學生：楊明毅　　指導教授：張立平

國立交通大學資訊科學與工程研究所碩士班

摘要

固態硬碟使用快閃記憶體作為儲存媒體,因此具有省電,耐震,以及優異的隨機存取能力,近年來已經展現出取代傳統磁碟的趨勢.固態硬碟的廠商經常宣稱其產品有優異的效能,但是使用者的感受卻不是如此而抱怨不斷,問題的原因是廠商沿用了傳統磁碟的測試工具,其典型的循序與隨機存取得到的測試結果,並沒有辦法反應固態硬碟的管理成本,因此並不能代表固態硬碟實際的效能.我們提出了一個基於實際存取行為的固態硬碟效能評測與分析方法.我們收集了8種在個人電腦上常見的Workload進行特徵分析,將具有相似特徵的Workload集合成Benchmark Suite,並且歸納這些特徵對固態硬碟管理議題造成的影響,使用者可以直接選用一組Benchmark Suite評估固態硬碟 的實際效能.

關鍵字:NAND 快閃記憶體(NAND Flash Memory),固態硬碟(Solid-State Disk),效能評測(Benchmark)

SSD Benchmark and Performance Analysis

student：Ming-Yi Yang　　Advisors：Dr. Li-Pin Chang

Department（Institute）of Computer Science
National Chiao Tung University

ABSTRACT

Solid State Drives (SSD) adopt NAND flash memory as storage media, and therefore have the characteristics of low power consumption, shock resistance, and low random access latency. SSD vendors promote their products by providing outstanding performance statistic, but there is a gap between the realistic performance users perceive and the measured performance. The main problem is that the statistics are measured using benchmarking tools designed for magnetic disks, and SSD management cost cannot be revealed by typical sequential and random access patterns. We have developed a real-workload based method to benchmark and analyze SSD performance .We collect eight common workloads in PC environment and characterize their behaviors. We classify workloads with similar characteristics into a benchmark suite and analyze the impact on SSD performance. Users can therefore select a benchmark suite to test realistic performance.

Keyword: NAND Flash Memory, Solid-State Disk(SSD), Benchmark

誌謝

能夠順利完成這篇論文並且通過口試, 心中充滿無限的感激. 首先要感謝的是我的好同學, 郭郡杰, 蘇宥全, 黃士庭以及廖秀芬. 郡杰經常和我討論各種研究相關的主題, 激發我的靈感, 這兩年也有多次合作成功完成各種工作的經驗, 是共患難的好夥伴. 宥全是我心目中的好榜樣, 我經常以宥全為標準督促自己. 當遇到不順利的事情時, 我總是會找士庭和秀芬聊聊, 心情會變得很好, 他們是非常熱心的人. 接下來我要感謝我的老師張立平教授, 老師對我的耐心與細心指導, 不論是專業知識或是做人處事的態度, 我都謹記在心. 特別要感謝老師給我各種磨練的機會, 包括論文報告, 指導專題生, 還有國科會計畫, 雖然當下感覺有點辛苦, 但事後回想我其實在這些鍛鍊中成長了不少, 謝謝老師這兩年來的指導, 我真的受益良多, 向您致上最高的敬意以及謝意. 最後我要感謝我所有的家人, 特別是我的父母以及妹妹, 因為你們給我的支持與鼓勵以及細心照顧, 讓我可以克服各種困難, 謹以此篇論文獻給我最親愛的家人.

目　　錄

中文摘要 …… i

英文摘要 …… ii

誌謝 …… iii

目錄 …… iv

表目錄 …… vi

圖目錄 …… vii

1. Introduction…… 1

2 Related Work…… 3

3 SSD Management…… 5

3.1 SSD Overview…… 5

3.2 Flash Memory Management…… 6

3.2.1 Address Mapping…….. 7

3.2.2 Garbage Collection 7

3.3 Buffer Management…… 7

3.4 SLC vs MLC…… 8

4. Benchmarking SSD…… 8

4.1 Benchmark Methodologies…… 9

4.2 Performance Metrics…… 9

4.3 Symptoms of SSD performance issues…… 10

5. Workload Characterization and Benchmark Suites…….. 12

5.1 Macroscopic Characterization…… 12

5.2 Microscopic Characterization…… 15

5.2.1 Seek Distance…… 16

5.2.2 Life Cycle…….. 16

5.3 Benchmark Suites …… 16

6 Experiments…… 17

6.1 Environment Setup…… 18

6.1.1 Test Bed and Testees…… 18

6.1.2 Workloads……. 19

6.1.3 Trace-Collection and Trace-Replay ……... 19

6.2 Macroscopic Characterization Result…… 20

6.3 Microscopic Characterization Result…….. 21

6.4 Benchmark Suites……. 27

6.5 Benchmark Results……. 28
6.5.1 Results of Current Storage Benchmarks……. 28
6.5.2 Results of Benchmark Suites…… 30
6.5.3 Discussion…… 34
7. Conclusion…… 35
Reference 36

表目錄

表(1)常見的 Micro-Benchmark 整理 5
表(2)SLC 與 MLC 的規格比較 9
表(3)測試平台的規格 18
表(4)待測物的規格 19
表(5)Workload 的使用者情境 19
表(6)巨觀的 Workload 分析結果 21
表(7)Benchmark Suite 分類結果 28

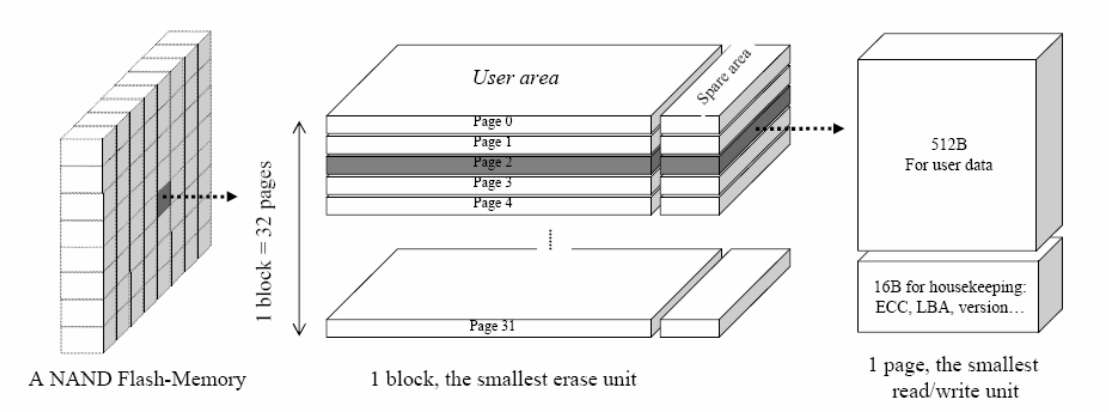
圖目錄

圖(1)快閃記憶體的物理結構 1
圖(2)部署快閃記憶體的方式 2
圖(3)固態硬碟的硬體架構 6
圖(4)系統架構 10
圖(5)使用回應時間判讀管理活動 11
圖(6)使用 PBR 判讀管理活動 11
圖(7)Mapping 機制不良 12
圖(8)回收機制不良 13
圖(9)寫入緩衝不良 13
圖(10)寫入與更新 14
圖(11)循序與隨機 15
圖(12)對齊與偏移 15
圖(13)Seek Distance 示意圖 16
圖(14)Life Cycle 示意圖 17
圖(16) PCMARK05 的微觀分析 22
圖(17)Browser 和 EMAIL 的寫入分佈 23
圖(18)安裝 Windows 和 Linux 的寫入分佈 24
圖(19)Windows 以及 Linux 的 Life Cycle 和 Seek Distance 分佈圖 25
圖(20)BT 和 eMule 的寫入分佈 26
圖(21)eMule 的 chunk 下載行爲 26
圖(22)eMule 的 Life Cycle 和 Seek Distance 27
圖(23)IOMeter 循序寫入測試:X 軸爲傳輸長度,Y 軸爲 MB/sec 29
圖(24)IOMeter 的隨機寫入測試:X 軸爲傳輸長度,Y 軸爲 MB/sec 30
圖(25)Buffer Suite 的測試結果 30
圖(26)Mapping Suite 的測試結果 32
圖(27)GC Suite 的測試結果 32
圖(28)BT 的測試結果 33

1. Introduction

NAND 快閃記憶體因為體積小, 耐震, 省電, 以及快速的隨機存取能力, 廣泛的應用在嵌入式環境中. 隨著每一個顆粒(cell)可以儲存的位元數(bit)增加, 降低了單位容量的成本, 使得 NAND 快閃記憶體具備取代傳統機械式硬碟的競爭力. 固態硬碟, 以 NAND 快閃記憶體為基礎的儲存系統, 現在已經普遍出現在筆記型與個人電腦, 甚至是大型資料庫的應用場合.

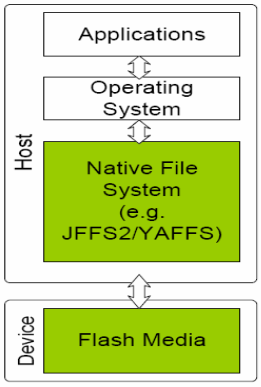
一個快閃記憶體晶片(Flash Chip), 內部被劃分為許多大小相同的區塊(blocks)，而區塊內部又細分為許多大小相同的頁(page), 如圖(1)所示。典型的『頁』與『區塊』之大小, 各自是2KB與128KB, 每一個區塊的壽命有限, 只能承受固定次數的抹除, 之後這個區塊的頁面就無法進行可靠的存取. NAND快閃記憶體具有獨特的物理特性, 這些物理特性與傳統硬碟完全不同, 因此衍生出完全不同的管理議題. NAND快閃記憶體的讀取/寫入/抹除的單位與時間不對稱, 讀取和寫入是以page為單位, 抹除則是以block為單位, 讀取所需的時間最短, 大約是20微秒, 寫入需要200微秒, 抹除極為費時須要1.5毫秒. 一個被寫過的頁是不能被重新寫入的, 除非這個頁被抹除, 因此更新資料的方式是採取out-place的策略, 也就是將最新版本的有效資料找一個乾淨的page存放, 而不是直接覆蓋原本資料存放的頁面, 一份資料的位置可能因為更新而不斷的改變, 所以需要Address Translation機制. 當可寫的空間逐漸變少, 就必須透過抹除區塊回收無效資料所佔據的空間, 這個動作稱為Garbage Collection. 為了避免回收的時候抹除有效的資料, 需要進行一連串的資料搬移與區塊抹除, 但是每一個區塊的有限, 因此我們必須公平的選擇每個區塊來抹除, 這個動作稱之為Wear Leveling. 處理位置轉換, 空間回收以及平均磨損三個議題, 就是快閃記憶體管理的主要任務.



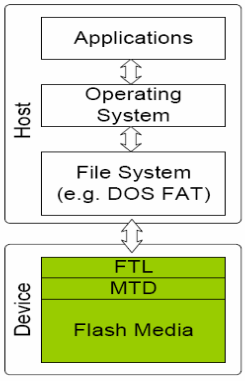
圖(1)快閃記憶體的物理結構

.

快閃記憶體的管理主要有兩種模式,第一種方式如圖(2)-1,是在Host 端採用原生的檔案系統,如 YAFFS[15],JFFS2[16],在快閃記憶體上直接處理各種管理議題,通常適用於嵌入式環境中,例如手機,PDA,Router.在 PC 的環境中,檔案系統的設計都是以以傳統硬碟為設計基礎,這類型的檔案系統並不直接適用於以 NAND 快閃記憶體為基礎的儲存系統,因此在裝置上需要一個抽象的轉換層,將快閃記憶體的物理特性隱藏並且模擬區塊裝置(Block Device)的介面.這一個轉換層稱為 FTL(Flash Translation Layer),負責處理因為 NAND 快閃記憶體衍生的管理議題,這種方式被稱為Disk-Emulation,如圖(2)-2 所示.大容量的儲存裝置如固態硬碟(SSD),CF card, SD card 皆採用此種部署方式.不論採用哪一種部署方式,系統的實際效能皆與快閃記憶體的管理方法有直接的關係.因此要對快閃記憶體裝置進行效能評估(Benchmark),必須針對其使用的管理方法進行測試.



圖(2)-1 原生檔案系統



圖(2)-2 Disk emulation

圖(2)部署快閃記憶體的方式

效能評測的方法主要分成 Trace-Replay, Macro-Benchmark 及 Micro-Benchmark 三種類型.Trace-Replay 和 Macro-Benchmark 的目標都是測試一個裝置在一般使用情形下的整體效能.Trace-Replay 是將一個實際的存取行為紀錄並且在待測物上重現,而 Macro-Benchmark 則是透過人工合成的存取樣式代替實際的 Workload 對待測物進行測試.Micro-Benchmark 的目標是要測試某一個特定操作的成本,或是極端情形下的效能表現,其存取樣式並不一定存在實際的應用環境中.現階段的硬碟評測工具,絕大多數都是以傳統機械式磁碟為待測物,測試的結果 Seek 和 Rotate 的成本以及資料傳輸的成本.使用傳統為磁碟打造的 Micro-Benchmark 並不足以完全反映固態硬碟管理方法所付出的抹除和資料搬移的成本,採用這些工具所得到的測試結果可能會讓使用者做出錯誤的選擇.

現階段的測試主要有兩大問題,首先是一般的黑箱測試使用的效能

指標並沒有辦法指出造成效能低落的核心原因,其次是測試使用的存取樣式並不能夠涵蓋各種真實的存取樣式.因此沒有辦法完整的反應管理方法實際運作的效率.我們提出了一套基於實際 Workload 的全新效能評測與分析方法,可以有效解決這兩個問題.我們收集了八種常見於一般個人電腦的 Workload 的寫入 trace,以 trace-Replay 的方式對待測物進行效能評測,並且提出了新的效能指標以及對 Workload 進行特徵分析,以釐清效能不彰的原因.

我們提出了全新的效能指標 Per-KB-Response,傳統的指標如吞吐量獲回應時間,都包含了傳輸的成本,而我們的指標將回應時間除以傳輸資料量因此消除了傳輸成本的影響凸顯有異常高回應時間的小的寫入動作,可以辨別固態硬碟的管理活動.要診斷固態硬碟效能低落的原因,可以透過觀察一個 Workload 中異常的 Per-KB-Response 在空間和時間上的分布情況,因此我們歸納出了當寫入緩衝,空間回收,以及位置轉換機制效率不佳時的典型症狀,可以利用這些症狀診斷固態硬碟的哪個環節出了問題.但是對於使用者來說,判讀這些症狀並不是一件簡單的事,畢竟這需要對固態硬碟管理有一定程度的了解,因此我們必須將我們的方法轉化為讓一般使用者可以重複利用的形式.

為了要讓使用者可以更直接的利用我們的研究成果,我們的做法是將收集到的 Workload 進行分類.我們分類的方式是先對 Workload 進行特徵分析,我們的分析著重在與固態硬碟管理有關的部分,除了一般常見的循序與隨機比例,寫入長度分佈外,我們加入了冷熱資料混合的情形以及散亂的程度,我們分析各種特徵會對固態硬碟管理及效能造成的影響,並且實際測試驗證後,將有類似特徵的 Workload 集合起來成為一個套件,我們稱之為 Benchmark Suite.我們一共有五個套件,可以針對傳輸速度,位置轉換,空間回收,寫入緩衝,以及整體效能進行評估,使用者可以依照需求直接取用一組 Benchmark Suite 針對特定議題進行測試,而測試的數據就直接代表固態硬碟處理該議題的處理成效,不用進行判讀測試結果的動作,因使簡化了測試的過程讓一般使用者方便同時又可以得到明確的效能數據以及其效能優劣的核心原因.

本論文的組織架 構如下:第二章是一些常用的硬碟效能評測工具的測試方法以及相關的文獻探討.第三章介紹基礎的固態硬碟軟硬體架構及管理方法作為特徵分析的依據.第四章描述我們提出的效能評測方法,第五章是 Workload 的分析與分類.第六章是實驗的環境設定,方法,以及結果的討論.第七章是本篇論文的結論

2. Related Work

在這一章我們探討目前廣泛被使用的效能評測工具以及方法,以及討論與開發快閃記憶體效能評測工具與技術有關的文獻.效能評測工具方面,我們選擇容易取得且免費的工具作為探討的對象,我們在 Windows 環境下使用它們並且紀錄了它們的存取行為模式.我們同時也觀察一些開發快閃記憶體管理演算法的文獻中所採用的驗證效能的方法.最後,我們介紹動機與我們類似的研究成果.

在 Micro-Benchmark 方面,既有的測試工具大多數都是以磁碟為測試的對象,如 HDTUNE, HDTACH, ATTO, 以及 H2bench, 以快閃記憶體為測試對象的工具比較少,目前比較容易取得的是 FDBench.表(1)是我們紀錄到的測試樣式以及項目.我們發現有些工具只進行讀取的測試,如 HDTUNE 與 HDTACH, 對於固態硬碟來說這種測試可以很輕鬆的取得優異的效能讀數,因為讀取的行為不會引起抹除以及資料搬移.在測試樣式方面,典型的樣式為大 Request 組成的循序存取以及小 Request 組成的隨機存取,前者的目標是測試資料傳輸的速率,後者則是測試磁碟 Seek 和 Rotate 的時間,但是對於固態硬碟來說,可以沿用前者進行傳輸速率的測試,後者雖然也可以用來測試管理演算法,但是意義不大.以快閃記憶體為測試對象的 FDBench, 將 Request 的大小提升至 64KB 以及 1MB, 這種做法減輕了管理上資料搬移的負擔反而隱藏了管理方法的成本.對於固態硬碟的管理方法,目前仍然沒有一個公認的可靠 Micro-Benchmark.

Macro-Benchmark 方面,比較知名的代表如 PCMARK[17]和 SYSMARK[18], 這類型的工具如它們的名稱所暗示的,測試的對象是一個系統平台,例如個人電腦,因此它們會合成各種存取樣式去模擬出現在目標系統上的 Workload 對儲存系統進行測試,最後依據儲存系統在各個 Workload 中的效能讀數計算出一個分數作為,分數越高的裝置代表越適合作為目標平台的儲存系統.採用這類型的裝置對固態硬碟做測試可以從分數上分出高下,不過這種指標並不明確,且這些工具的存取樣式缺乏比較詳細的描述,因此對於管理方法的開發並沒有太大的幫助.

工具名稱	測試項目	測試樣式
HDTUNE	Transfer Rate Access Time Burst Rate	32KB 循序讀取 0.5KB 隨機讀取
HDTACH	Burst Rate Random Access Read Throughput	0.5KB,128KB,256KB,512KB 循序讀取 0.5KB 隨機讀取
ATTO	Read Throughput Write Throughput	可調大小 循序讀取與寫入 隨機讀取與寫入
H2Bench	Seek Time Transfer Rate	0.5KB 64KB 循序讀取與寫入 隨機讀取與寫入
FDBench	Read Throughput Write Throughput	64KB,1MB 循序讀取與寫入 隨機讀取與寫入

表(1)常見的 Micro-Benchmark 整理

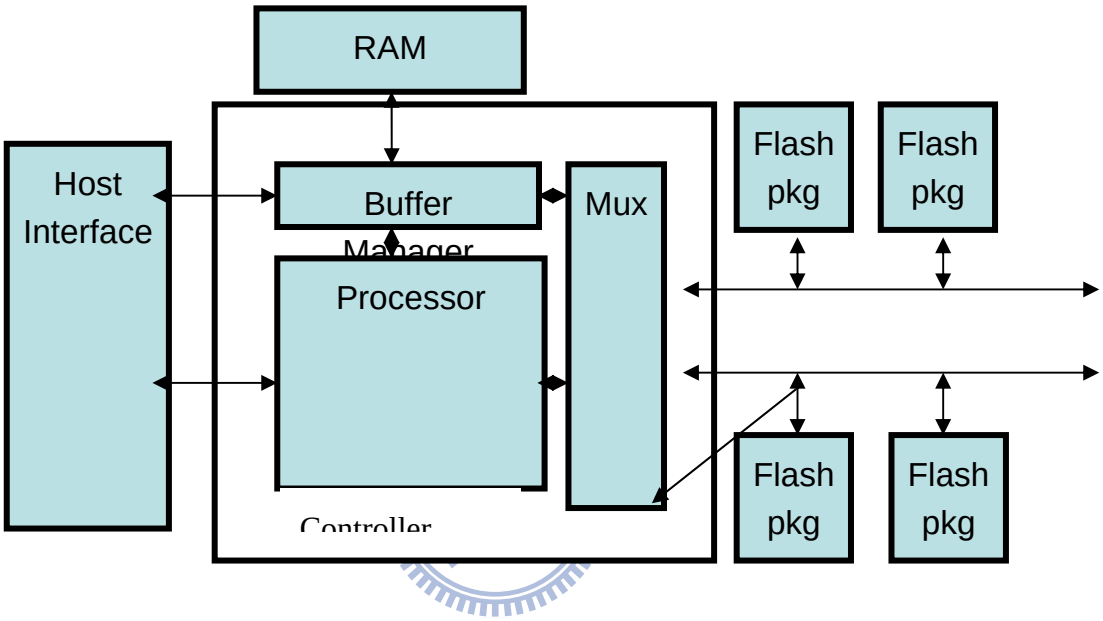
由以上的討論得知,典型的硬碟測試樣式包含了循序存取以及隨機存取兩種類型,循序存取是測試硬碟資料傳輸量,因此通常搭配較大的資料傳輸量,採用 MB/sec 作為 Metric.隨機存取是測試磁碟的 Seek 以及 Rotate 的成本,因此通常搭配較小的資料傳輸量,採用的 Metric 是 IOPS.從 MB/sec 以及 IOPS 並無法辨識管理方法的成本及效率,而循序與隨機兩種模式並不能夠囊括所有的實際存取行為,因此使用這些工具得到的測試結果,自然無法代表固態硬碟實際的效能.

.目前為止,只有少數論文在探討如何固態硬碟的效能以及測試方法,[1]提出了各種不同的設計方法對固態硬碟效能的影響,並且以軟體模擬的方式驗證,但是並沒有針對實際的產品測試.[2]提出了具體測試固態硬碟管理的存取樣式.但是沒有提供實際測試的結果,有待進一步的實驗證明其有效性.[3]定義了一套非常完整的固態硬碟效能評測方法,包括測試前的 Setup 以及測試後的 Cleanup,還有一系列測試用的存取樣式,著重在效能評測的方法論,沒有具體說明要如何測試出管理方法的差異.因此我們希望可以開發出一套測試實際固態硬碟產品的效能評測與分析的方法.

3. SSD Management

3.1 SSD Overview

圖(3)是一個典型的固態硬碟硬體架構區塊圖.固態硬碟的傳輸介面是和一般磁碟相同,都是採用標準的傳輸介面,如 SATA,IDE 等規格,因此固態硬碟是採用 Disk-Emulation 的方式部署快閃記憶體,而所有管理的方法都已韌體的形式存放在控制器中.最近由於市場上對於效能的要求越來越高,因此很多固態硬碟又加入了額外的 RAM 作為寫入緩衝,因此在控制器中必須加入額外的控制電路管理緩衝區.快閃記憶體的管理以及寫入緩衝的管理合稱為固態硬碟的管理.



圖(3)固態硬碟的硬體架構

固態硬碟的效能極限是由硬體的架構以及規格所決定,例如平行化增加頻寬的方式,快閃記憶體晶片的規格,甚至是處理器的規格等都會影響固態硬碟的效能.但是實際上使用者感受到的效能,是來自於固態硬碟管理運作以後的效能,因此我們必須要針對固態硬碟的管理設計我們的效能評測方法.

3.2 Flash Memory Management

快閃記憶體的管理的三大議題是位置轉換,空間回收,以及平均磨損.這三個議題彼此互相牽制與影響.以外部黑箱測試的角度來說,我們只能夠測得因固態硬碟管理成本所延誤的回應時間,內部的具體管理行為則無法得知,因此我們沒有辦法評估平均磨損的效果.我們在這一章主要介紹位置轉換,空間回收,以及寫入緩衝的管理機制,以及它們對效能的影響.

3.2.1 Address Mapping

由於快閃記憶體採用 out-place 的更新策略,因此一份有效的資料在快閃記憶體的實體位置會不斷的改變,因此需要一套有效率的方法將邏輯位置轉換成為實體的位置,並且充分的利用快閃記憶體上的空間.固態硬碟的讀寫是以頁面為單位,而抹除是以區塊為單位.因此在位置轉換使用的單位上,主要可以分成 Page-Level, Block-Level 以及融合兩者的 Hybrid 方法.Page-Level 的轉換方式可以非常有效的利用快閃記憶體上的空間,並且可以任意的把資料集中到某個區塊或是從區塊中分離特定的資料出來,這種對應方式可以很細膩的操縱資料擺放的方式而且可以充分的利用快閃記憶體上所有的空間,因此又稱為 fine-grain 的轉換方式,但是這種方式需要大量的 RAM 支援,因此並不適合應用在固態硬碟上.Block-Level 的對應方式對 RAM 的需求比 Page-Level 少,但是沒有辦法像 Page-Level 般靈活的運用快閃記憶體上的空間,當碰到小的寫入與更新時,會產生嚴重的碎烈情況而造成空間利用度不佳的情形,除此之外,也無法任意的將資料集中或是分離.

目前固態硬碟上的位置轉換方法,大部分都是採 Hybrid 的方式,這種方式將所有的區塊分成 Data Block 以及 Log Block 兩類,Data Block 內是存放有效的資料,而 Log Block 主要是吸收 Data Block 內的資料更新.資料寫入會先用區塊為單位對應到 Data Block 中,如果 Data Block 中已經有這份資料的話就轉移到 Log Block 之中,而 Data Block 與 Log Block 之間是採取 Page-Level 的對應.因此可以比較有效率的處理小的資料更新,但是當這些小的更新散佈在不同的 Data Block 時,仍然會出現空間利用率低落的情形,不過有很多不同的 Data Block 以及 Log Block 的對應方式可以減輕此種現象[4][5][6].

空間利用率低的情況,會造成一種空間不足的假象,因而造成頻繁的空間回收動作影響外部的效能.當一個 Workload 並沒有很強烈的更新行為,如大範圍的凌亂寫入,卻引發固態硬碟上很多的空間回收動作的時候,就代表這個固態硬碟的位置轉換方式沒有辦法有效的利用快閃記憶體上的空間,效能因頻繁的空間回收動作而低落.一個設計良好的位置轉換方法,應該要能夠充分的利用實體空間而有效率的處理各種長度的寫入行為.

3.2.2 Garbage Collection

空間回收的目標,是透過抹除回收被無效資料所佔據的空間,但是抹除的單位是一個區塊,在抹除之前必須先搬移一個區塊中的有效資料,因此空間回收的動作會引發一連串的讀寫以及抹除,這些讀寫以及抹除

就是快閃記憶體管理的主要成本,也是固態硬碟的效能瓶頸所在.

為了要使用最少的管理成本回收最多的空間,空間回收演算法必須仔細的考量回收啟動的時機,要抹除的區塊,以及資料搬移到哪個區塊存放[7],我們稱這些考量為空間回收的策略.一般來說,空間回收啟動的時機應該要盡量的往後延遲,因為這樣可以減少資料搬移的成本,而且可以提高每次抹除所回收的空間,增進空間回收的效率.而在選擇要抹除的區塊時,通常都會選擇包含最多無效頁面的區塊作為抹除的對象,這是一種以效率為考量的Greedy 策略,但是由於快閃記憶體的壽命限制,因此還必須加入平均磨損的考量.

在實際的Workload之中,資料會有冷熱之分,當一個資料頻繁的被更新時,我們稱其為熱資料,反之則為冷資料.當冷熱資料混合再一個區塊中時,會對空間回收的效率造成影響[8][9],因為熱資料常常會引發空間回收的動作,但是此時夾雜在一起的冷資料會需要資料的搬移的成本,因此在搬移的過程中,應該要將冷熱資料分離到不同的區塊中.

冷熱資料的分佈狀況依Workload會有很大的差異,因此空間回收的策略可能會暴露在非常不利其運作的環境中.一般最常見的情況就是冷熱資料的混合,若此時沒有有效的冷熱資料分離方法,效能會因過多的資料搬移而不佳.另外一種情形就是冷熱資料分離但是冷熱程度差異極大的情形,在這種情形下如果沒有辦法回收冷資料佔據的空間給熱資料使用,那麼熱資料的更新就會造成一種競爭少數空間的現象引發頻繁的空間回收,亦會造成效能低落.空間回收的效率很容易受冷熱資料影響,因此固態硬碟的效能可能會因為Workload不同而導致效能出現差異,甚至可能造成即時系統的排程出現失誤[10].

3.3 Buffer Management

為了追求更極致的效能,越來越多的固態硬碟都增加了額外的RAM作為寫入的緩衝或是讀取的快取,由於快閃記憶體的寫入動作比較慢,且會觸發管理的動作,作為寫入的緩衝比較能夠體現出差異,所以接下來我們將討論重點放在寫入緩衝上

固態硬碟對於寫入緩衝的管理可以分成兩類,第一種類形沿用傳統的管理機制,例如FIFO,LRU等策略,而第二種方式是針對快閃記憶體的特性設計新的Buffer管理機制,例如FAB[11],BPLRU[12]等策略.不管是哪種策略,只要加裝了Buffer對固態硬碟的效能提升都有幫助,但是提升的程度會有顯著的不同.原因是第一種策略的目的只是單純利用RAM的優勢

縮短存取的時間,並無意減輕固態硬碟的資料搬移以及抹除的成本.第二種類型的管理策略與固態硬碟的管理緊密的配合,希望利用 Buffer 的機制,減輕快閃記憶體的管理成本而提升固態硬碟的效能,最直接的方式就是以一個 erase unit 作為管理的單位,在 Buffer 中盡量將同一個 erase unit 中的更新收集完後寫回快閃記憶體上,如此可以節省資料搬移的成本並且提升空間的利用率,透過降低管理成本解決效能的問題.

當固態硬碟的寫入緩衝沒有考慮快閃記憶體管理的成本時,因為並不能夠減輕管理的負擔,再緩衝區還可以吸收資料的時後,效能會非常的好,但是當緩衝區的資料寫回時,必要的資料搬移與抹除仍然不可避免,此時效因此效能會有很大的落差產生,對效能產生不良的影響.寫入緩衝設計不良的時候會導致固態硬碟效能出現極大的落差.

3.4 SLC vs MLC

快閃記憶體的管理成本來自於有效資料的搬移以及抹除,反應在外部的效能就是回應時間的高低.為了提高容量,越來越多的固態硬碟選擇 MLC 的快閃記憶體晶片做為儲存媒體.表(2)是 SLC[13]與 MLC[14]的比較,MLC 的晶片在讀寫的時間上比 SLC 高,此時固態硬碟的管理優劣的差異,就更容易顯現,因此管理方法與策略對於效能的影響對採用 MLC 晶片的固態硬碟來說更為顯著.另外在壽命部分,MLC 比 SLC 的壽命少了一個數量級,因此對於平均磨損的挑戰很大.MLC 的出現讓固態硬碟管理的地位更加的重要.

Spec\Cell	SLC	MLC
Read Latency	0.08 ms	0.16 ms
Write Latency	0.2 ms	0.9 ms
Erase Latency	1.5 ms	1.5 ms
Erase Cycle	1000K	100K ms

表(2)SLC 與 MLC 的規格比較

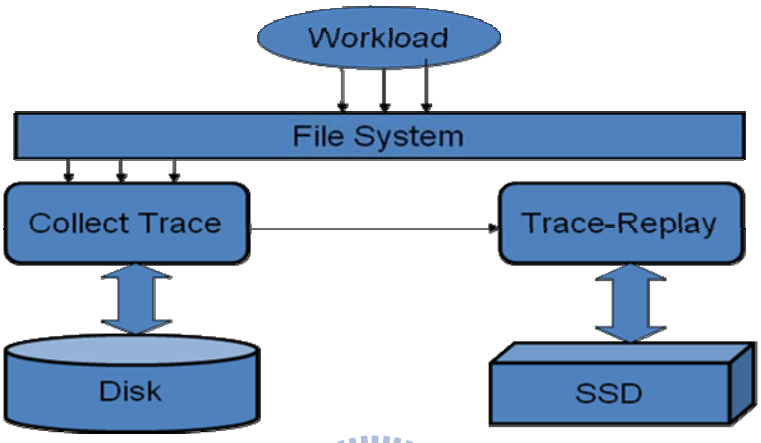
4. Benchmarking SSD

我們對於固態硬碟的效能評測,是以黑箱測試的形態測試外部的效能.黑箱測試的優點是測試所需的環境設定以及參數比較簡單,不需要特殊的設備或式平台就可以展開,劣勢則是很難從測試的結果判讀出造成效能低落的核心問題,如果不改善這個問題,那麼我們的測試結果失去了意義.為了改善這個問題,我們設計出了一套全新的效能指標,可以用來診斷固態硬碟效能異常的原因,彌補了黑箱測試的不足.

4.1 Benchmark Methodologies

我們提出的固態硬碟效能評測方法主要有兩個步驟,第一個步驟是收集實際 Workload 的存取記錄,這些存取的紀錄稱為 Trace,我們稱為 trace-gathering . 第二個步驟是在待測的固態硬碟上完整的重現存取行為,稱為 Trace-Replay.

現階段的效能測試工具多以循序與隨機的測試樣式為主無法包含所有實際的存取行為,因此也無法反應固態硬碟的真實效能,因此我們採用真實的 Workload 作為我們的測試樣式,圖(4)是我們的系統結構.



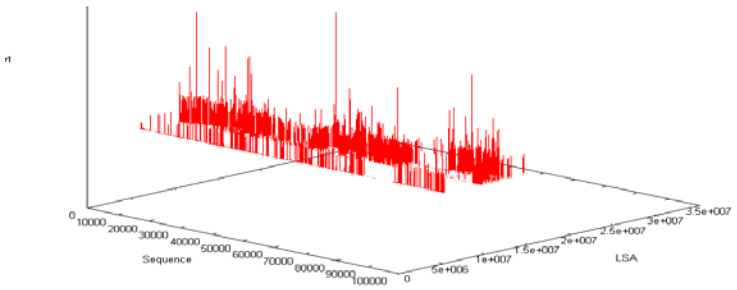
圖(4)系統架構

在 trace-collecting 階段,我們首先建立一個專屬的硬碟給我們的目標 Workload 使用,在這個硬碟上並沒有存放其他的使用者資料,當一個 Workload 收集完成後便執行格式化消除之前 Workload 所殘留的資料. 我們這麼做的目的是希望可以讓一個 Workload 的特徵可以被我們很完整的捕捉到而不被其他程序所稀釋掉. 鮮明的特徵有利於我們辨認一個 Workload 對特定管理議題的衝擊並且分類. 由於固態硬碟的管理都是被寫入所引發,因此我們只收集一個 Workload 的寫入行為. 在收集 trace 的時候,必須要在檔案系統層以下的驅動程式層進行,如此才能收集到實際會發生在硬碟上的存取行為.

在 trace-replay 階段,我們同樣在驅動程式層在待測的固態硬碟上重現一個 Workload 的寫入行為. 待測固態硬碟的初始狀態為無任何資料,以免影響管理方法的決策. 為了避免固態硬碟上的背景管理機制被啟動,寫入與寫入之間不能有延遲時間. 背景的管理機制通常對提升效能是有幫助的,但是這並不代表管理的方法比較優異而成本較少,因為管理的成本只跟寫入的位置以及先後順序有關係,所以在測試的時候不能讓背景活動隱藏了管理方法的缺陷.

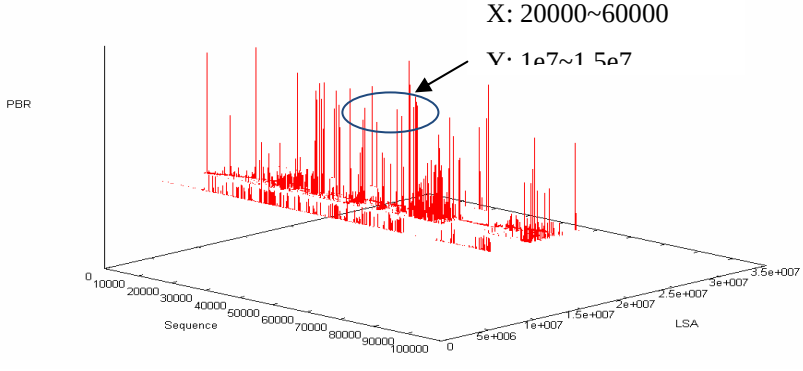
4.2 Performance Metrics

回應時間是一個常用於固態硬碟測試的指標,因為當一個寫入動作啟動了管理機制,則回應時間會因為資料搬移與抹除而延遲.但是在實際的Workload 中,每一筆寫入動作的資料長度並不相同,資料長度較長的寫入動作本身就需要比較長的傳輸時間,因此單純從回應時間辨認Workload 引發的管理活動會容易造成混淆,如圖(5)所示.管理成本與傳輸成本混雜在一起,因此必須要提出一個新的指標反映固態硬碟的管理活動.



圖(5)使用回應時間判讀管理活動

我們將每一筆寫入動作的回應時間除以傳出的資料長度,得到了新的指標:Per-Byte-Response,簡稱為 PBR.我們的目標是要突顯傳輸長度短但是回應時間卻很長的寫入動作,因為這種現象很明顯是管理的成本所造成的,我們稱類型的寫入動作為異常.為了要能夠辨認效能低落是哪個管理環節所造成的,我們設計了一種三維的視覺化方式,我們以每一個寫入動作的編號為 X 軸,以寫入動作的起始位置為 Y 軸,而 Z 軸則為每一個寫入動作的 PBR 值. 呈現異常的寫入在空間與時間上的分佈情形,再進一步從分布情形判讀是哪一個管理環節出了問題導致效能不佳,如圖(6)所示,和圖(5)對比之下,可以明顯辨認出效能的瓶頸.

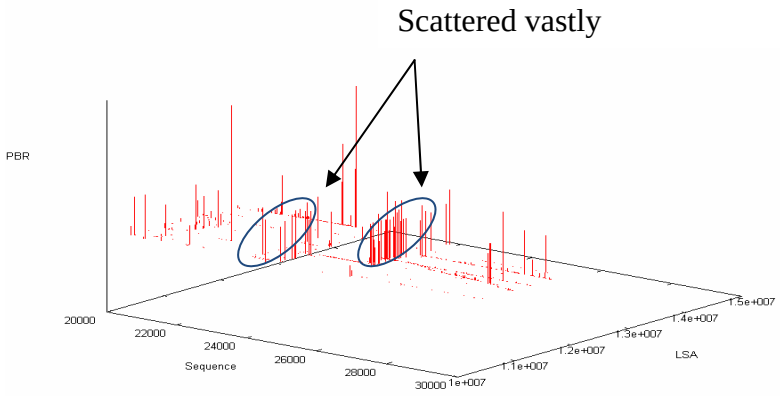


圖(6)使用 PBR 判讀管理活動

4.3 Symptoms of SSD Performance Issues

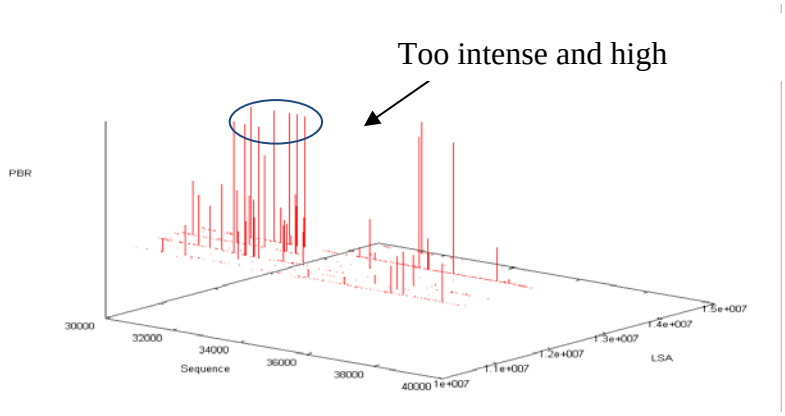
對於一般使用者來說,直接判讀某一個 Workload 造成的異常寫入分佈並不是一件簡單的事,畢竟這需要對固態硬碟的管理有一定程度的了解.因此我們從 PBR 的異常分佈整理出一些可以輕易判讀的典型症狀,並且告訴使用者這些症狀對應了固態硬碟的管理上的何種缺失,這樣我們的效能指標才可以被一般使用者重複利用.正常的固態硬碟管理活動,應該是一種很被動但是很有效率的行為,因此在分佈圖上的情形,應該是異常的寫入數量極少,且這些異常的讀數也不高.當管理的某個環節失效時,在分布圖上的表現是有大量的寫入異常出現,我們將這些異常的分佈歸納成三種狀況,分別對應了位置轉換,空間回收,以及寫入緩衝管理失效的情形. 可以在圖(6)中標示 X 座標介於 20000 至 60000,Y 座標介於 1e7 到 1.5e7 的圖形觀察到這三個現象.

第一種異常的分布情形是,異常的寫入動作散亂的分佈在大範圍的邏輯位置上,如圖(7)所標示的情形.這種情形表示這些異常與空間有密切的關係.在更新的行為並不強烈的情況下,卻引發了頻繁的管理動作,典型的原因固態硬碟上的空間並沒有妥善的利用,因此我們可以把這種異常歸類到位置轉換機制無法完整的利用實體空間所導致的結果,



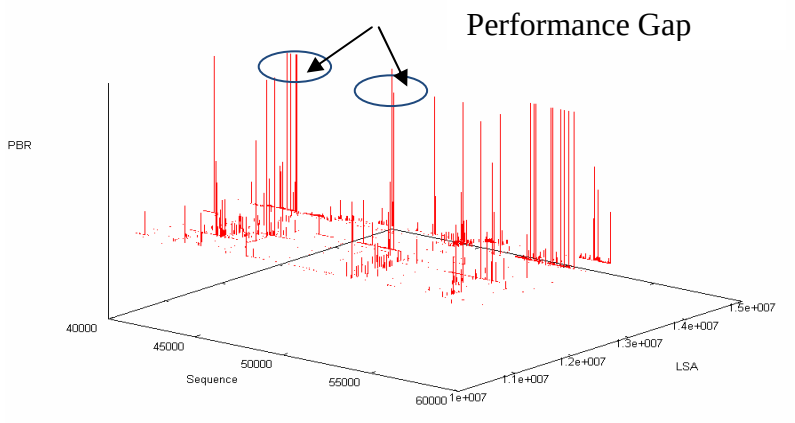
圖(7)Mapping 機制不良

第二種異常的分布情形是在小範圍且短時間內密集的出現而且 PBR 讀數極高,如圖(8)所示,這種情況表示某份資料正被頻繁的更新,因此引發了空間回收的動作.可以從兩個方向來解讀這種異常,首先,異常高的 PBR 讀數表示空間回收時付出很多資料搬移的成本,典型的原因為冷熱資料沒有分離.而密集的出現表示空間回收的機制在策略上出現問題,例如過早啟動或是挑選不當的回收目標,因此我們將短時間小範圍的密集寫入異常歸類為空間回收機制不良的症狀.



圖(8)回收機制不良

第三種異常的情形是出現效能落差極大的時候,如圖(9)所示,這種落差的情況有可能與寫入緩衝有關.當緩衝區還有空間的時候,寫入被吸收到 RAM 中,因此效能極佳,但是當緩衝區開始將資料寫回到固態硬碟上時,可能會引發固態硬碟的管理動作而造成效能下滑,如果緩衝區的管理機制沒有搭配快閃記憶體的管理,那麼下滑的情況可能會更嚴重,因為管理的成本並沒有減少.我們可以將這種效能落差很大的情形,歸咎到寫入緩衝管理的方法沒有辦法減少管理的成本.



圖(9)寫入緩衝不良

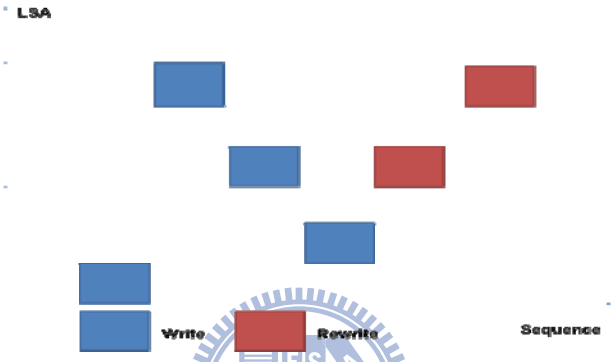
5. Workload Characterization and Benchmark Suites

我們的測試使用真實的 Workload 對固態硬碟進行效能測試,而固態硬碟的效能取決於 Workload 的寫入行為,我們將 Workload 的寫入行為歸納為特徵,從這些特徵的強度我們可以分析一個 Workload 對固態硬碟管理造成的衝擊,並且將 Workload 分類到 Benchmark Suite 中,讓使用者可以直接取用其中的 Workload 對固態硬碟特定管理議題進行測試.除此之外,詳細的特徵描述有助於使用者避免類似的場合中使用效能不佳的固態硬碟.

5.1 Macroscopic Characterization.

我們的 Trace 格式紀錄了一筆寫入動作的順序, 起始位置, 以及資料傳輸的長度, 從這三個參數中取出與固態硬碟管理成本相關的特徵, 作為我們將 Workload 分類到 Benchmark Suite 中的依據, 我們的分析並不牽涉到固態硬碟上的硬體架構, 因為這些資訊在黑箱測試階段是無法得知的.

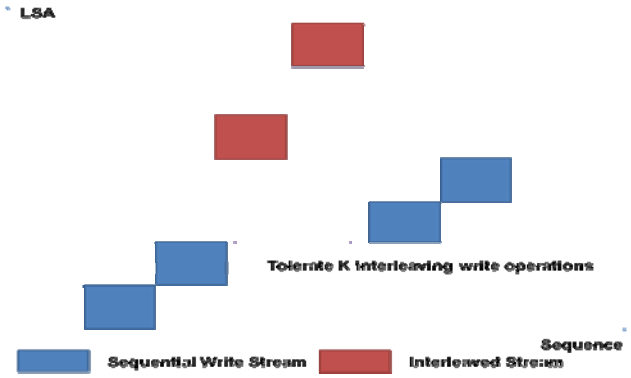
我們將一個 Workload 中的寫入動作, 分類成為寫入與更新兩類. 若是一筆寫入動作的範圍之前並沒有被寫入過任何資料, 此寫入動作稱為寫入, 若寫入的位置含概已經被寫入過的區域, 則稱此寫入動作為更新, 如圖(10)所示.



圖(10)寫入與更新

我們分別統計寫入以及更新的資料量, 兩者資料量的比可以做為一種描述整體時間區域性的指標. 當一個 Workload 的更新資料量越大的時候, 表示固態硬碟的空間回收方法負擔越大, 若此時寫入的資料量大於寫入緩衝區的容量, 那麼這些更新就沒有辦法完全的被吸收, 這時也會考驗寫入緩衝的管理機制. 因此寫入與更新的資料量可以看出一個 Workload 對於寫入緩衝的需求以及空間回收造成的負擔.

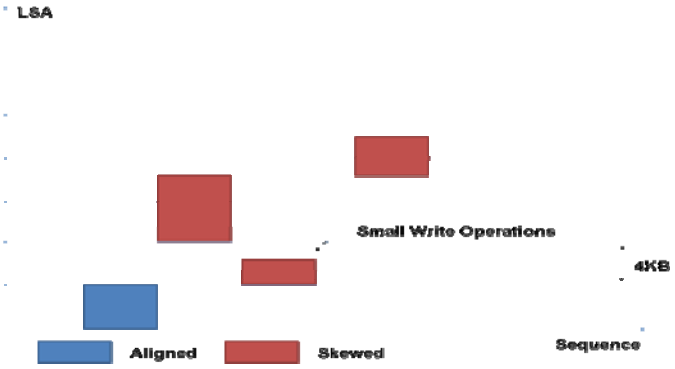
我們進一步將寫入與更新依照模式分類為循序與隨機兩種類型. 我們對於循序的定義比較寬鬆, 因為現在的作業系統多半支援多工的環境, 所以循序的寫入動作間可能會被其他寫入行為中斷, 因此我們在判斷循序時必須設定一個容忍的 K 值, 也就是容許兩個循序的寫入動作間安插 K 個寫入動作, 若是不屬於循序的寫入動作, 我們把它歸類為隨機, 如圖(11). 一個 Workload 中循序的寫入動作所佔的比例可以表示空間區域性的強度, 我們可以用這個指標分析一個 Workload 需要的管理成本. 循序的寫入動作通常需要的資料搬移比隨機的少, 而循序的寫入動作占多數時也表示寫入的動作並不會很散亂, 比較不會有空間利用度不足的問題. 因此我們用循序的比例來判斷一個 Workload 所需要的管理成本.



圖(11)循序與隨機

最後我們針對一個 Workload 的資料傳輸長度進行統計,我們以 2 的冪次方為間距,統計每一個間距內的次數.我們以 4KB 作為判斷寫入動作大小的依據,我們將長度 4KB 以下的寫入動作歸類為小型.我們接下來判斷每一個寫入動作的起始位置以及終止位址是否有對齊 4KB 的邊界,並且計算對齊的寫入動作占整體的比例.統計一個 Workload 中大小寫入動作的分佈以及對齊的比例可以評估一個Workload 對於位置轉換單位的精細度需求,除此之外也可以判斷對於硬體傳輸速度的需求.

我們將一個Workload 中更新,循序,對齊的比例以及傳輸長度分佈的統計稱為巨觀的分析(Macroscopic),因為這些指標呈現的是一個 Workload 整體的特徵.對於規模較小且特徵明顯的Workload,這四項指標就足以作為分類的依據.但是對於一些規模較大的 Workload,對固態硬碟管理有挑戰的特徵可能隱藏在某個特定的時間和空間中,而這些特徵不一定會顯現在這四項指標之中,因此必須要搭配微觀(Microscopic)的分析方法.



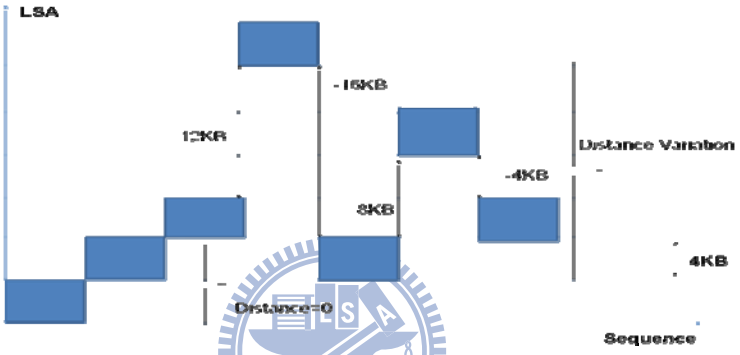
圖(12)對齊與偏移

5.2 Microscopic Characterization

微觀的分析主要是針對隨機寫入動作為主的 Workload,更進一步的了解寫入動作在空間以及時間上的分佈,並且了解其特徵形成的原因.這樣的分析有助於找出管理方法具體的缺陷.

5.2.1 Seek Distance

所謂的 Seek Distance 指的就是一筆寫入動作的終止位置與下一筆寫入的起始位置的距離,對於傳統的磁碟來說,這項指標代表效能成本.但是對於固態硬碟來說,不會有讀寫頭的移動成本,因此 seek distance 的值大小並不直接代表效能的成本.但是這個概念可以沿用來表示寫入動作在空間上的分佈.對於固態硬碟的管理來說,寫入動作的散亂程度會影響位置轉換機制的效率.要表示資料的散亂程度,可以計算每一個寫入動作間的 Seek Distance,如果分佈的情形是集中的話,Seek Distance 的值會收斂在固定的小範圍中,而且彼此差異不大.當寫入大範圍的散亂,則 Seek Distance 之值會呈現極大的差異,無法收斂於小範圍之中.因此我們可以觀察 Seek Distance 變化的情形,判斷一個 Workload 寫入動作的凌亂程度.



圖(13)Seek Distance 示意圖

要觀察一個 Workload 中寫入動作的散亂情形,可以觀察單位時間內 Seek Distance 的變異情形,如圖(13)所示,當循序的寫入時,Seek Distance 會等於 0,可是寫入動作轉為比較隨機時 Seek Distance 就開始產生變動,而這變動的幅度會隨寫入的範圍而增加.短且散亂的寫入動作情形對於設計位置轉換方法來說是很重要的一件事,因為這會影響空間的利用率.

5.2.2 Life Cycle

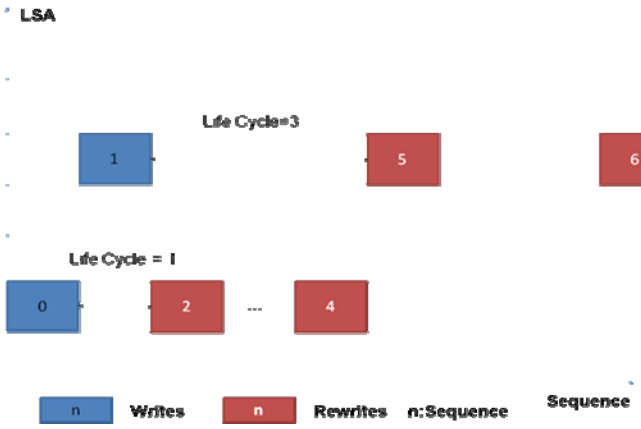
因為時間區域性的關係,資料會有冷熱之分,所謂的冷熱程度表示資料被更新的頻繁程度,因此冷熱程度就是某一個位置的寫入動作在時間上的分佈.一般常見的做法是直接統計一個位置的寫入動作次數,代表其冷熱程度.但是這種做法對於大規模的 Workload 並不適用,因為這些寫入動作可能是在短時間內密集的出現,也有可能是分散在長時間中,如圖(14)所示.因此在判斷資料冷熱程度的時候,必須納入時間的考量.我們使用了 Life Span 和 Life Cycle 量化冷熱程度,定義如下:

定義:Life Span

令某個 LSA 的位置為 X, FIRST_ACCESS(X)表示 X 第一次寫入的 Request

編號, LAST_ACCESS(X)最後一次寫入的 request 編號.
Life_Span(X) = LAST_ACCESS(X)-FIRST_ACCESS(X).

定義:Life Cycle:
令 Write_Count(X)表示一個位置 X 被寫入的次數,則
Life_Cycle(X) = Life_Span(x) / Write_Count(x). 若 Life_Cycle(X)=0, 表示這個位置並沒有發生更新.



圖(14)Life Cycle 示意圖

我們可以透過觀察單位空間中,資料冷熱程度的差異情形,了解冷熱資料的分佈以及混合程度,掌握冷熱資料的分佈以及差異程度可以對於空間回收機制是一件很重要的事,當冷熱資料可以正確的識別且分離的時候,可以節省很多資料搬移的成本.

5.3 Benchmark Suites

我們依據特徵分析的結果將 Workload 分類到不同的 Benchmark Suite 中,共有五種類別,分別是 Transfer, Buffer, Mapping, GC 以及 Overall.
Transfer Suite 主要是用來評估硬體傳輸的成本.Buffer, Mapping,以及 GC 用來評估管理的成本,最後 Overall Suite 用來評估整體的效能成本.

Transfer Suite 中的 Workload 通常是以循序的寫入為主,資料傳輸長度以長(例如 64KB)的占大多數.循序的寫入可以測試固態硬碟的平行化架構,長的資料傳輸可以測試存取的頻寬.除此之外, 循序的寫入動作以及長的資料傳輸比較不會引起高成本的管理動作,傳輸的成本可以獨立出來.這部分的 Workload 其實與典型的循序存取樣式類似.

在測試固態硬碟管理方面,由於平均磨損的成效無法從外部測量, 所

以我們只有 Buffer, Mapping , 以及 GC 三個 Suites. 在測試管理方法時, 需要的是以短的且隨機的寫入動作為主的 Workload, 這樣可以減少傳輸成本的介入. 在 Buffer Suite 中, Workload 的寫入以及更新的資料量必須要超過寫入緩衝的容量, 如此才能觸發寫回的機制, 因此要注意寫入與更新的比例和資料量這兩項指標. 在 Mapping Suite 中, 我們需要的是有散亂的更新行為的 Workload 還有短的或偏移的寫入行為, 可以測試空間利用的情形, 因此要注意散亂的程度以及偏移的比例這兩項指標. 最後在 GC Suite 中, 我們需要的是很強烈的密集更新行為, 搭配不同冷熱資料的混合情形, 此時要注意冷熱混合的情形, 以及更新的比例這兩項指標.

Workload 的規模可能相當的龐大, 其中包含了各個 Benchmark Suite 的特徵, 此時它的效能不是由單一特徵所決定, 從指標上看不出明顯的特徵. 因此我們將這類型的 Workload 分到 Overall Suite 中, 用來測試傳輸成本和管理成本合併以後的效能成本. 我們另外會依時間分段檢視其存取行為, 以免忽略了可用的存取行為.

6. Experiment

在實驗結果的部分, 我們首先呈現每一個 Workload 的特徵分析結果, 並且

6.1 Environment Setup

6.1.1 Test Bed and Testees

測試平台的部分, 使用一般的個人電腦即可, 但是在主機板的部分必須要支援 SATAII 的傳輸規格, 測試平台的硬體與軟體的規格如表(3)所示:

CPU	Intel Core 2 Dual 1.87GHz
Motherboard	ASUS P5B
RAM	DDR2 2GB
OS	Windows XP + SP2

表(3)測試平台的規格

在待測物的部分, 我們選定了四款固態硬碟. 以價格分類的話, MTRON 和 SAMSUNG 屬於比較高階的產品, 它們皆有額外的 RAM 作為寫入緩衝. OCZ 屬於中階的產品, 而創見屬於低階. 詳細的規格如表(4)所示

廠牌	介面	顆粒	容量	控制器
MTRON	SATAII	SLC	32GB	MTRON
SAMSUNG	SATAII	SLC	32GB	SAMSUNG
TRANSCEND	SATAII	SLC	16GB	SMI
TRANSCEND	SATAII	MLC	32GB	SMI
OCZ	SATAII	MLC	64GB	JMICRON

表(4)待測物的規格

6.1.2 Workloads

我們主要收集常見於一般個人電腦上常見的 Workload, 因為目前固態硬碟已經普遍出現在小筆電中而越來越普及. 我們的 Workload 主要分成一般應用, 網路應用, 安裝作業系統, 以及 P2P 的應用四大主題, 每個主題包含兩個 Workload, 每個 Workload 收集的情境如表(5)所述

名稱	使用者情境
Copy MP3	將 200 個 MP3 檔案從一目錄複製到另一目錄中
PCMARK05	執行 PCMARK05 的 HDD test
Browser	使用 Internet Explorer 5.0 瀏覽網頁 3 小時
EMAIL	使用 Outlook 2003 收取 GMAIL 信箱中 200 封郵件
Install Windows	安裝 Windows Server 2003, 所有設定皆為預設值, 檔案系統為 NTFS
Install Linux	安裝 Fedora Linux Server 4, 所有設定皆為預設值, 檔案系統為 EXT3
eMule	使用 eMule 0.48b 下載 3 個檔案, 為時 3 小時
BT	使用 u-torrent 下載 10 個檔案, 為時 3 小時

表(5)Workload 的使用者情境

在一般應用的主題中, 我們特別收集了 PCMARK05 這個常用的測試工具, 觀察其測試樣式是否可以測出固態硬碟管理的缺失. 在網路應用中, 我們收集了網頁瀏覽以及收信兩種行為, 是非常見於小筆電的 Workload. 使用者拿到固態硬碟通常會想要利用其作為系統碟, 因此我們收集了安裝作業系統的 Workload, 安裝作業系統的快慢會決定使用者對某固態硬碟的第一印象, 且安裝的過程中基本上已經包含一個作業系統中常見的存取行為. 最後 P2P 的部分被視為非常不利於固態硬碟的 Workload, 原因是小且隨機的寫入動作, 因此我們收集了 eMule 和 BT 這兩個常見的 P2P 應用程式的存取行為, 以此做為對固態硬碟最嚴苛的考驗.

6.1.3 Trace-Collection and Trace-Replay

Trace-Collection 部分,我們在 Windows XP 的環境下,使用一獨立的硬碟並且於其上建立了一個 16GB 的 NTFS 磁碟分割作為收集 Workload 的專屬空間,使用 Diskmon[19]作為收集 trace 的工具.當收集完畢以後,必須執行格式化的動作將使用者資料刪除.一般應用的 trace 都是以此法收集但安裝作業系統類的 Workload 的收集方式比較不同,因為在安裝的階段沒有辦法直接啟動任何的 trace 機制.我們在 Windows 中使用 VMWARE 建立一台以實體硬碟作為儲存系統的 Virtual Machine,在上面安裝作業系統,由於 Host OS 端為 Windows,因此我們仍然可以使用 Diskmon 收集 trace.

Trace-Replay 部分,我們使用 Windows API 中的 CreateFile()以及 WriteFile()建構我們的 Replay 工具,首先以 CreateFile()將固態硬碟以實體裝置的模式開啟,接下來使用 WriteFile()執行 Synchronous 的寫入動作.在測量效能的部分,我們在每一筆寫入動作的前後,放置 RDTSC()函式,這是一個可以讀取 CPU 的 Clock Cycle 的組合語言函式,所有的換算動作都是在 Replay 結束後進行,因此並不會造成太多的延長時間而影響測試的準確性.

6.2 Macroscopic Workload Characterization

我們首先對我們收集到的八個 Workload 進行巨觀的分析,一些特徵很明顯的 Workload 可以直接以巨觀分析的結果進行分類.我們判定循序的方式是容忍兩個連續的寫入動作間安插 10 個非連續的寫入動作.每一個指標的強度以其佔整體傳輸資料量的比例表示.表(6)為分析結果:

Workload	Data Transferred	Sequential Ratio	Rewrites Ratio	Alignment Ratio	Frequent Data Length
Copy MP3	816MB	96%	0.1%	0%	64KB
PCMARK05	2130MB	82%	81%	0%	64KB,4KB
Browser	477MB	4%	80%	31%	4KB
EMAIL	40MB	35%	56%	40%	4KB
Install Windows	1888MB	56%	21%	0%	64KB,4KB,0.5KB
Install Linux	2387MB	25%	18%	0%	4KB,128KB
eMule	9437MB	55%	5%	0%	512KB,4KB,12KB
BT	12587MB	73%	5%	0%	512KB,64KB,4KB

表(6)巨觀的 Workload 分析結果

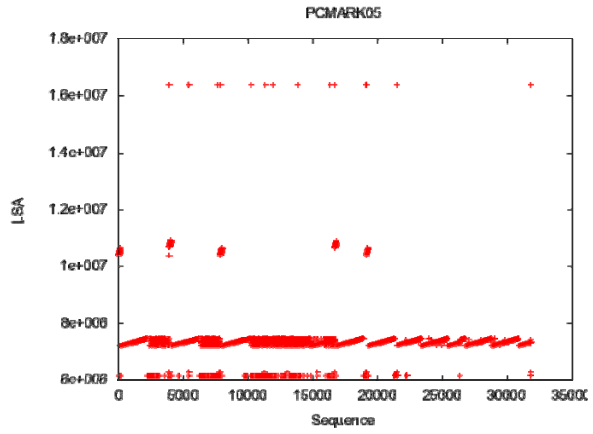
在巨觀分析的結果中,4KB 是最常出現的傳輸長度,因為這是 NTFS 和 EXT3 檔案系統預設的邏輯分頁大小,但是實際運作時,卻沒有對齊 4KB 的邊界,原因是檔案系統通常會保留數個開頭的 sector,導致寫入的動作產生偏移,這種情形會使固態硬碟效能退化.

Copy MP3, PCMARK05,Browser,以及 EMAIL 的特徵非常強烈,因此可以直接進行這四個 Workload 分類.Copy MP3 絕大多數的寫入動作都是 64KB 的循序寫入,更新的強度極弱,因此不會消耗管理的成本,可以歸類到測試傳輸速率的 Transfer Suite 中.PCMARK05,Browser 以及 EMAIL 三者都有比例極高的更新,空間回收的效率可以明顯反映在外部效能上,所以可以直接分類到測試空間回收的 GC Suite 中,不過 Browser 和 EMAIL 的冷熱資料混合差別仍然需要微觀的分析.

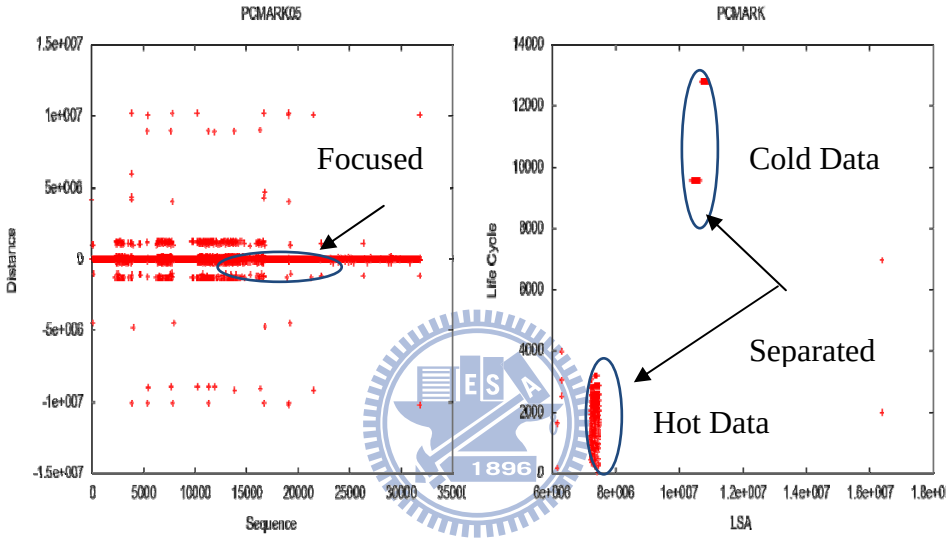
安裝作業系統和 P2P 下載的四個 Workload 規模比較大,沒辦法直接以巨觀分析的指標進行分類,因為其中的某一部分寫入行為可能對固態硬碟效能產生威脅,但無法直接反映在整體效能上,且巨觀指標的結果與預期有落差,例如安裝 Linux 的過程中只有 25%的寫入動作屬於循序,而 P2P 的下載是循序的寫入動作占大多數,所以必須等微觀的分析完成以及了解其特殊行為後再進行分類.

6.3 Microscopic Workload Characterization

由於 Copy MP3 中絕大多數的寫入動作都是循序的寫入,因此我們不對這個 Workload 進行微觀的分析.PCMARK 是一個很受歡迎的測試工具,和傳統的磁碟工具不同,它是以模擬真實的 Workload 行為的方式進行測試.我們的巨觀分析中也顯示了很強的更新行為,因此我們利用微觀的分析觀察 PCMARK 對於固態硬碟管理方法有甚麼挑戰.圖(16)是 PCMARK 的寫入動作,以及 Seek Distance 和 Life Cycle 的分佈圖.從 Seek Distance 分佈圖來看,我們可以發現 PCMARK 的寫入動作很集中,一般普遍認為 PCMARK 中是以隨機的存取為主.但事實上由於範圍過於集中的關係,寫入變得很容易連接在一起,又由於我們定義的循序是可以被中斷的,因此統計出來的循序比率很高.而從冷熱資料的分佈來看,可以很明顯的發現冷熱資料是幾乎是完全分離的,而且冷熱資料的差異程度很大,減輕了辨認的難度.



(a)寫入動作分佈圖



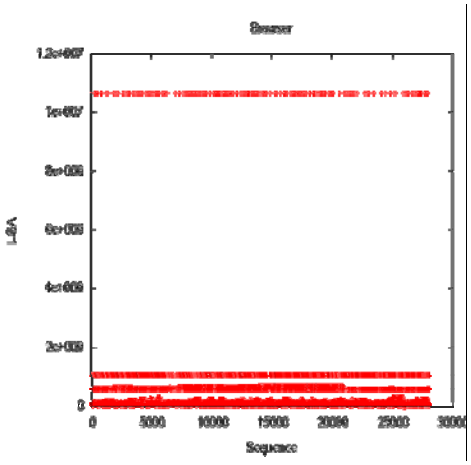
(b)Seek Distance 分佈圖　　(c)Life Cycle 分佈圖

圖(16) PCMARK05 的微觀分析

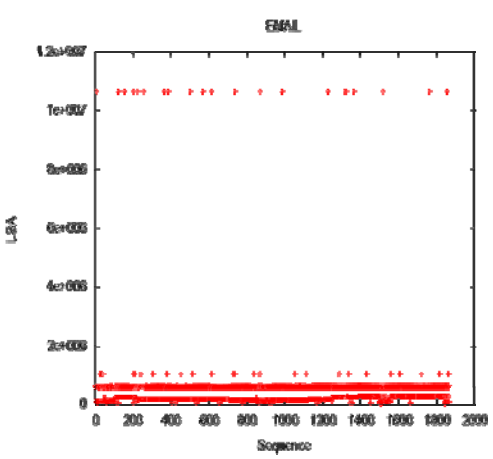
從微觀的分析結果發現,雖然PCMARK有很強烈的更新行為,但是這些更新的行為並不會對固態硬碟的管理構成威脅,不過這種測試樣式仍然比典型的磁碟測試樣式得到的結果有效,不過PCMARK實際的評分也會納入讀取的效能表現,所以仍然有管理成本沒辦法彰顯的問題.我們將PCMARK的寫入動作分類到GC Suite中,它可以測量固態硬碟在冷熱資料分離的情況下,空間回收運作的效能.

圖(17)-(a)(b)為Browser以及EMAIL的寫入動作分佈圖,兩者有極為類似的存取行為,皆為小範圍的密集分布情形.Browser會將網頁的圖片,影片或音效等多媒體元件暫存在硬碟中,下次開啟同一個網頁以後就不用再重新載入,因此可以加快網頁開啟的速度,而這些暫存檔案由Index.dat進行管理,這個檔案需要常常被更新,屬於熱資料.而EMAIL的管理與Browser類似,也有熱資料的存在.因此Browser和EMAIL都是代表小範圍的密集更新行為,但是有冷熱資料的存在所以循序的寫入動作

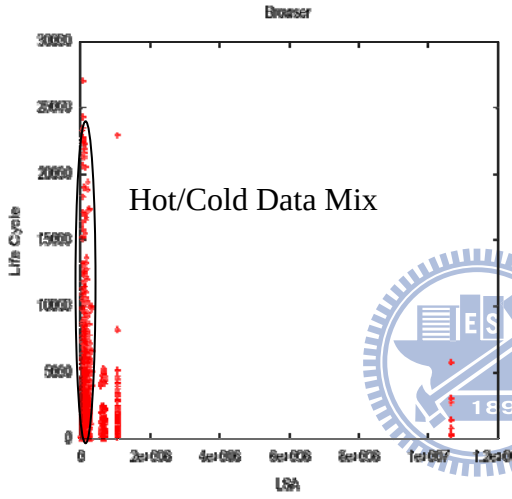
比率較少.



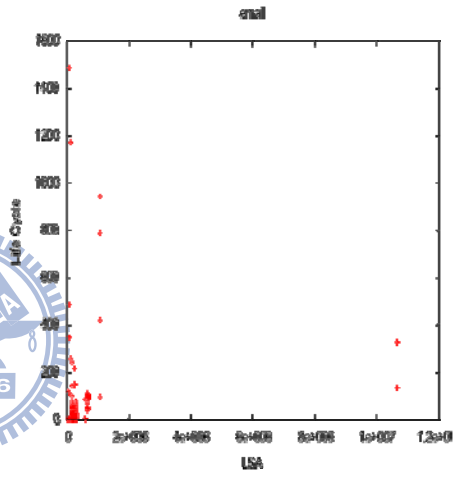
(a)Browser 的寫入動作分佈



(b)EMAIL 的寫入動作分佈



(c) Browser 的 Life Cycle 分佈



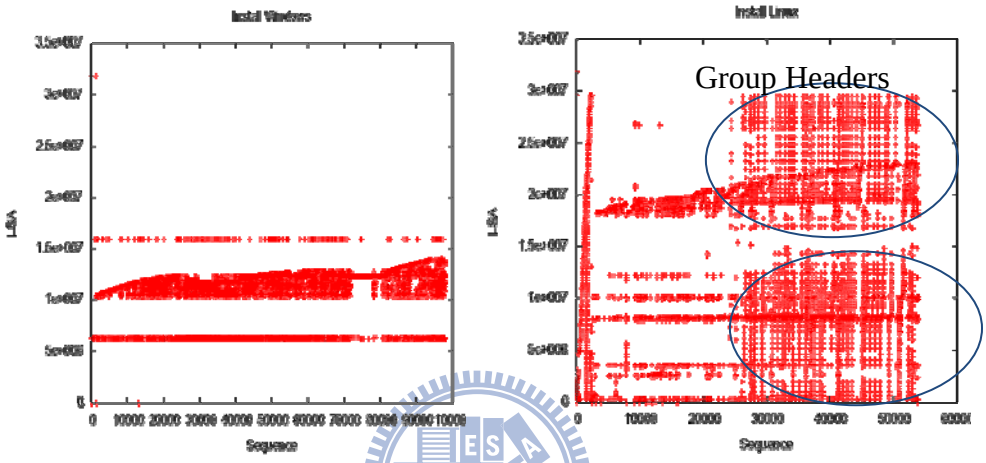
(d)EMAIL 的 Life Cycle 分佈

圖(17)Browser 和 EMAIL 的寫入分佈

由於這兩個 Workload 的寫入分佈圖上可以清楚辨認出集中的情形，因此我們略過 Seek Distance 的分析. 這兩個 Workload 因為更新行為強烈，所以直接分到 GC Suite 中. 我們進行冷熱資料混合程度的分析，了解這兩者間的差異，冷熱資料在空間中的分佈如圖(17)-(C)(D)所示. EMAIL 由於 Workload 的整體規模比較小，因此資料冷熱程度的差異也比較小，冷熱混合的情況也並不嚴重，所以適合做為一個入門的測試，觀察固態硬碟空間回收啟動的. Browser 的冷熱混合的情況就相當嚴重，而且冷熱的差異並不明顯，在辨識上也有一定的難度，因此 Browser 適合做為考驗空間回收方法極限的 Workload. 因此在我們的 GC Suite 中，有三種不同程度的情境可以測試固態硬碟空間回收的效率.

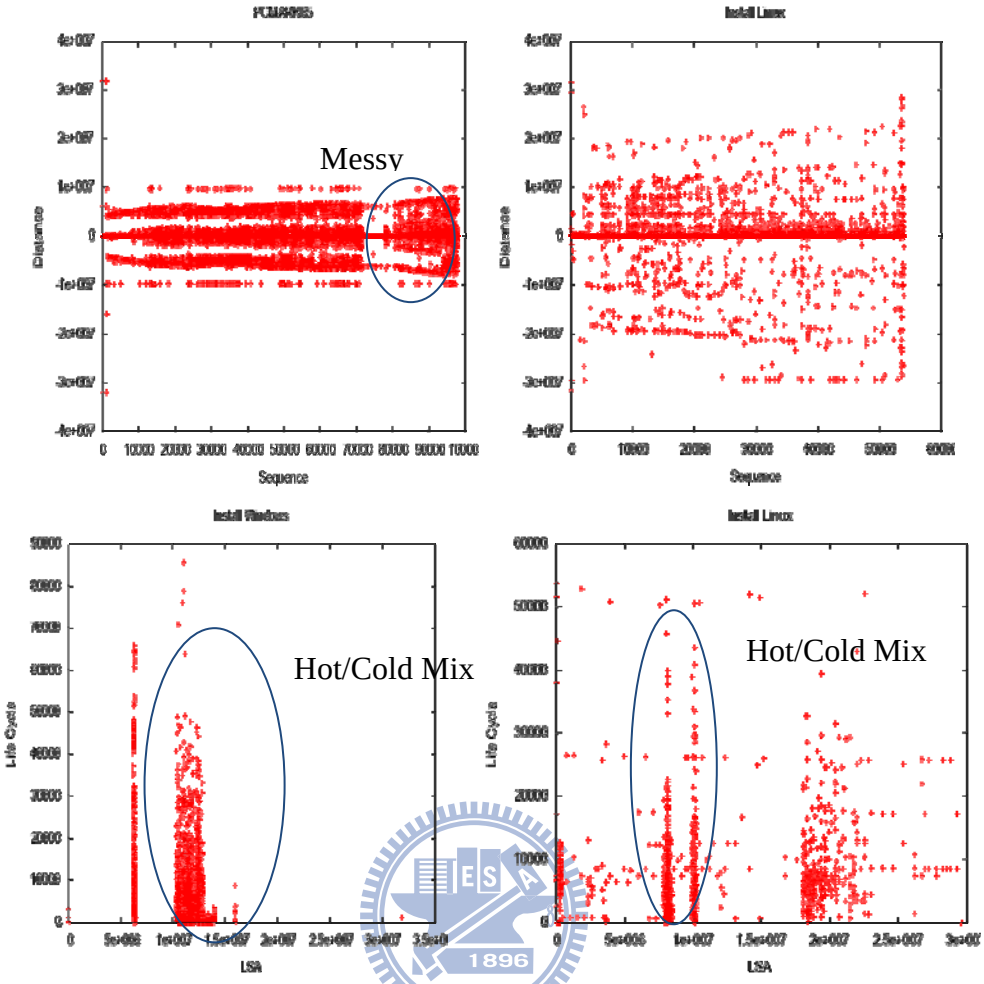
圖(18)為安裝 Windows 以及 Linux 的寫入動作分佈圖. 巨觀的指標中 Linux 的循序比例只有 25%，從其寫入分佈圖來看，出現了大範圍的散亂寫

入分佈,如圖(18)中的標示,我們進一步的分析這些散佈的寫入,發現這些寫入動作發生的位置都是在 Group Headers 的範圍中,這是 EXT2/EXT3 存放 metadata 的空間[].造成這種現象的原因是因為在 EXT2/3 檔案系統預設的環境中,即使讀取也會觸發一個寫入去更新 inode 中的 atime,因此造成了隨機的寫入動作超過循序的情形.在 Windows 的部分,從巨觀的分析中看不出很強烈的特徵,基本上循序的寫入為主,因為安裝的動作主要就是複製檔案,伴隨一些檔案系統以及一些設定組態檔的隨機寫入動作.



圖(18)安裝 Windows 和 Linux 的寫入分佈

圖(19)為兩者的 Seek Distance 以及 Life Cycle 的分佈圖.從 Seek Distance 的分佈圖上可以反應出 Linux 的大範圍散亂的寫入動作分佈,其實 EXT2/EXT3 的設計目標是要盡量的減少 Seek Distance,減低傳統磁碟的負擔,但是當我們單獨考慮寫入部分的時候,Seek Distance 就會因為寫入四散在 group header 中而呈現散亂的情形.安裝 Windows 的部分,大致上屬於比較密集的分佈,不過也是有部分散亂的寫入動作存在,但這不是檔案系統造成的,而是安裝過程中一些組態的設定檔.在冷熱資料的分佈上面,兩者皆有冷熱混合的情形,不過以安裝 Windows 的情況較為嚴重,

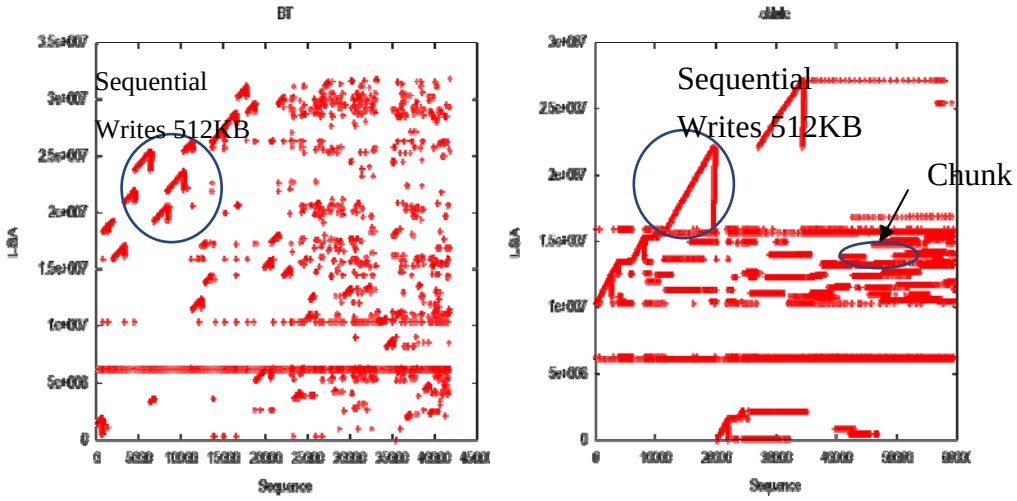


圖(19)Windows 以及 Linux 的 Life Cycle 和 Seek Distance 分佈圖

從以上分析的結果,Linux 的散亂 group header 更新行為非常用來測試固態硬碟的位置轉換機制,雖然也有冷熱資料混合的情形但是特徵沒有散亂的寫入分佈強烈,所以我們把 Install Linux 分配到 Mapping Suite 中.而安裝 Windows 中寫入的資料長度有大有小,循序與隨機的比重接近,有冷熱資料混合的情形,也有部分散亂的寫入動作,參雜了所有的特徵,因此我們將它分類到 Overall Suite 中,做為固態硬碟整體效能評估的測試情境.

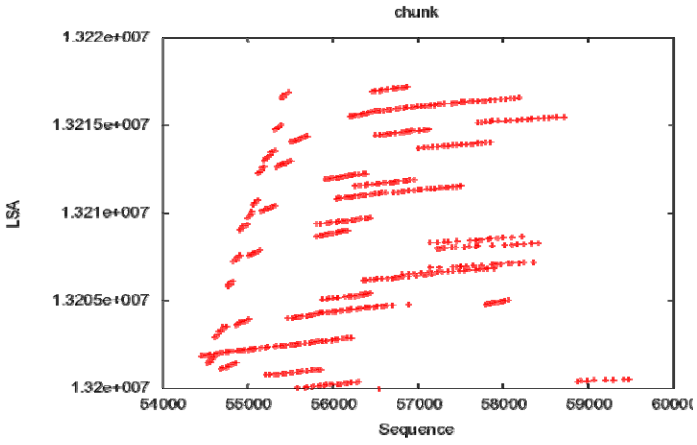
BT 和 eMule 都是很受歡迎的 P2P 下載軟體,它們的原理非常類似,概念上都是將檔案切割成數個 chunk,並且以此作為下載的單位,而同一時間可以有好幾個 chunk 在下載.兩者的主要差別在於 chunk 的大小,在 BT 中這個大小為 1MB,但是在 eMule 中為 9.28MB.兩者在巨觀的分析中有蠻高比例的循序寫入動作以極大的資料傳輸長度,我們發現這種行為來自於檔案開始下載之前預先保留空間的動作,如圖(20)中標示.另外巨觀的分析中顯示,BT 的傳輸長度多半比較長,我們發現這是因為我們使用的軟體 μ-torrent 預設有 32MB 緩衝區的關係,所以 chunk 會在緩衝區中完成

下載以後再一次寫回固態硬碟中,因此 BT 的行為是長的散亂寫入動作為主,而非短的散亂寫入,前者的重點在於傳輸速度,後者的影響在於位置轉換的空間利用率.

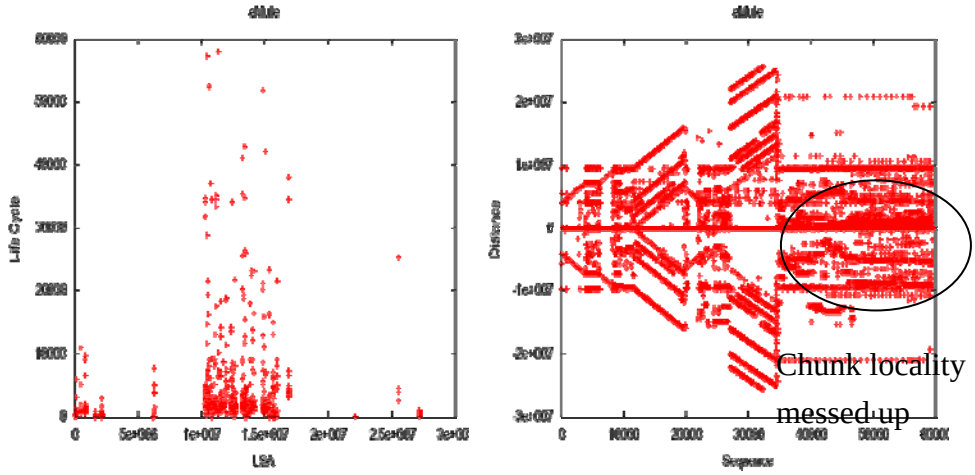


圖(20)BT 和 eMule 的寫入分佈

eMule 的緩衝只有 128KB,因此作用有限,下載時的資料傳輸單位介於 4KB 到 12KB,當 Chunk 開始下載的時候,並不是很散亂的存取,而是會集中在一個 chunk 內.因此一個 Chunk 有很強的空間和時間區域性.但是因為同一時間可能有好幾個 chunk 一起下載,所以導致一種很凌亂的假象,把觀察的單位時間延長就可以很清楚的觀察到一個 chunk 的空間與時間區域性,如圖(19)所示.



圖(21)eMule 的 chunk 下載行為



圖(22)eMule 的 Life Cycle 和 Seek Distance

我們收集到的 BT 中大部分寫入的資料都是較長的 512KB 以及 64KB, 而更新只佔了其中的百分之五, 因此決定效能的關鍵在於傳輸的速度, 因此我們把它歸類到 Transfer Suite 中. eMule 的部分前半段與 BT 一樣都是長的循序寫入, 但是後半段開始 chunk 下載的時候, 有非常特殊的空間與時間區域性的特徵. 這種特徵非常適合拿來測試空間回收或是寫入緩衝, 因為能否掌握一個 Chunk 的時間與空間區域性, 效能會有極大的差異. 不過我們將 eMule 的 chunk 下載行為分類到測試寫入緩衝管理的 Buffer Suite 中, 原因是好幾個 chunk 同時下載的時候營造的大範圍散亂假象可能會讓緩衝的空間一下就消耗完畢而頻繁的寫回, 因此這時緩衝區的管理策略是否能夠減輕管理成本就可以被體現出來.

6.4 Benchmark Suites

由以上巨觀與微觀的分析結果, 我們整理出了我們的 Benchmark Suites. Copy MP3 以及 BT 因為循序寫入且資料傳輸長度大, 在這種情況下傳輸的成本比管理的成本明顯, 因此歸類到 Transfer Suite 中. 在 GC Suite 方面, 依據巨觀分析的結果, 我們選擇更新強度高的 PCMARK05, EMAIL, 以及 Browser 三者, PCMARK 代表的是比較容易處理的循序更新, 而微觀的分析結果顯示 EMAIL 代表的是沒有冷熱資料混合的隨機更新, 而 Browser 代表的是最不好處理的高強度密集隨機更新搭配冷熱資料混合的情況, 因此這三個 Workload 恰好可以完整的測試固態硬碟的空間回收機制在不同強度的更新之下的表現. 在測試寫入緩衝的 Buffer Suite 部分, 可以利用 eMule 以 chunk 為單位的下載檔案行為, 因為每一個 chunk 都有很明顯的空間與時間區域性的情形, 可以體現寫入緩衝的管理單位大小差異. 位置轉換的部分, 可以利用 EXT2/EXT3 的 Group Headers 的更新進行測試, 我們收集到的 Trace 恰好包含了大範圍的 Group Header 更新行為. 最後想要評估固態硬碟包含傳輸以及管理成本的整體效能, 可

以使用 Install Windows 進行測試,因為其規模夠大,且其中涵蓋了所有可以測試固態硬碟效能成本的特徵.我們將每一個 Workload 的特徵以及最後的分類結果整理如表(7).

Workload	Summary	Suite
Copy MP3	96% Sequential 0.1% Rewrites 64KB	Transfer
BT	73% Sequential 5% Rewrites 512KB,64KB	Transfer
PCMARK05	82% Sequential 81% Rewrites 64KB,8KB	GC
EMAIL	35% Sequential 56% Rewrites 4KB	GC
Browser	4% Sequential 80% Rewrites Hot/Cold Mixture 4KB	GC
eMule	55% Sequential 5% Rewrites Chunk-Level Locality 4KB,12KB	Buffer
Install Linux	25% Sequential 18% Rewrites EXT2/EXT3 Group Headers 4KB,128KB	Mapping
Install Windows	56% Sequential 21% Rewrites Hot/Cold Mixture 4KB,64KB,0.5KB	Overall

表(7)Benchmark Suite 分類結果

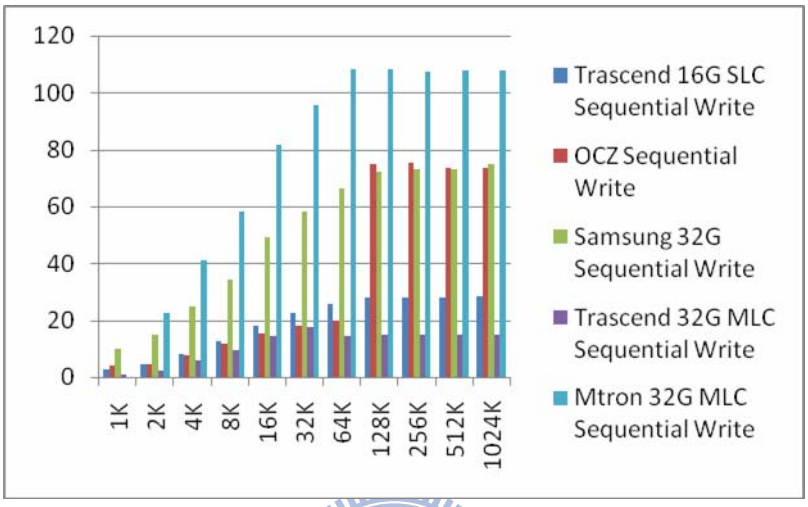
6.5 Benchmark Results

我們的評測分為兩部分,第一部分是使用典型的循序以及隨機的測試結果,第二部分使用 Benchmark Suite 中的 Workload 對固態硬碟管理的評測結果,將兩者比對,證明固態硬碟的管理對效能有決定性的影響.

6.5.1 Results of Current Storage Benchmarks

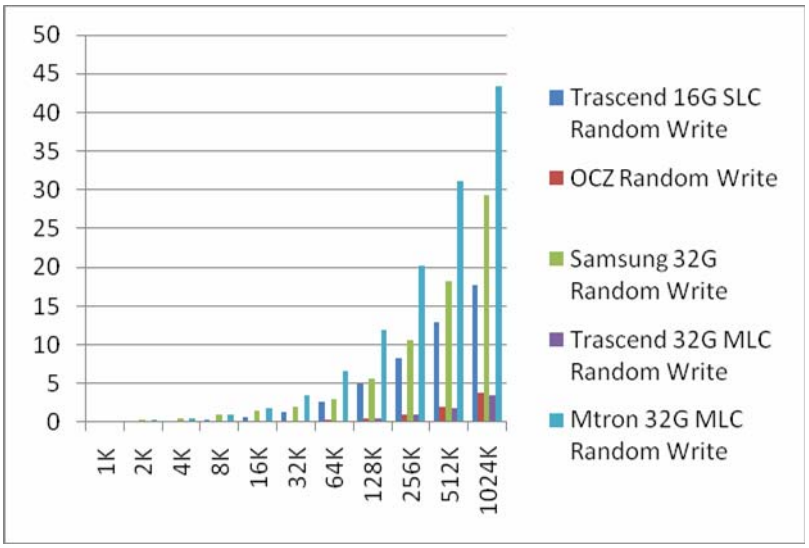
我們使用了有名的 IOMeter[20]調配典型的循序以及隨機寫入樣式對我們的待測物進行初步的測試,並且將測試結果所得到的效能讀數以及待測物間的效能優劣關係做為我們的 Benchmark Suite 測試的比較基礎.

循序寫入的測試可以看出一個固態硬碟的效能極限,圖(21)的測試結果顯示,MTRON 的固態硬碟擁有極為優異的傳輸速度,大幅勝出其他的代測物.OCZ 的固態硬碟在傳輸長度較長的循序寫入測試與 Samsung 不分軒輊,但是不擅長處理資料長度短的寫入.而 Transcend 系列的固態硬碟效能明顯遜於其他三者.在循序寫入的測試,得到的結果是 MTRON 大勝其他四者,OCZ 和 Samsung 不相上下,Transcend 系列的的故態硬碟遠遜於其他三者.



圖(23)IOMeter 循序寫入測試:X 軸為傳輸長度,Y 軸為 MB/sec

在均勻隨機寫入測試的部分,所有的代測物都呈現效能不佳的情形,不過採用 SLC 晶片的固態硬碟均大幅勝過採用 MLC 的固態硬碟,這個測試忠實的反映了 SLC 和 MLC 在先天體質上的差異,如圖(22)所示.在 SLC 的固態硬碟中,MTRON 和 Samsung 可能因為配備有寫入緩衝而在效能上都勝過 Transcend 的 SLC 一籌,而 MTRON 又略優於 Samsung 表示其寫入緩衝可以比較有效的吸收短的隨機寫入動作.均勻隨機寫入測試的結果顯示了 SLC 的優越性,以及加入寫入緩衝可以提升整體效能.



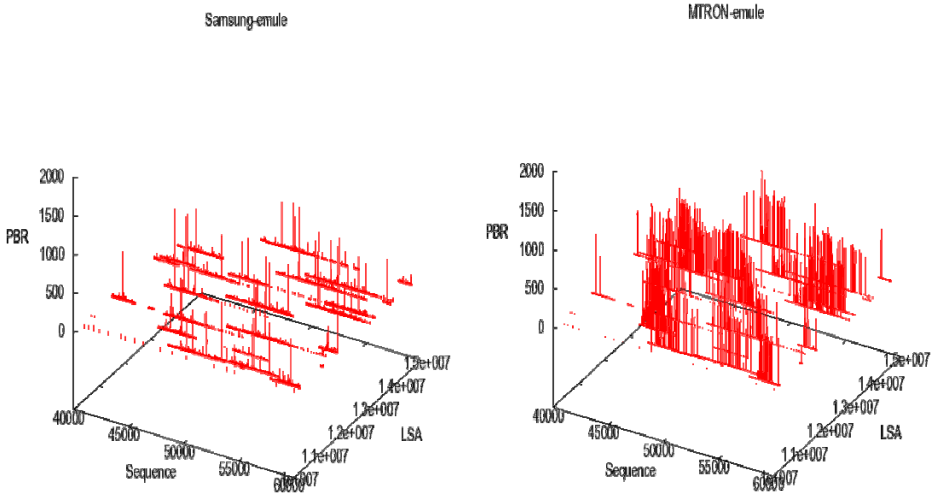
圖(24)IOMeter 的隨機寫入測試:X 軸為傳輸長度,Y 軸為 MB/sec

IOMeter 的測試結果大致上與待測物的產品定位一致,其中 MTRON 的效能最為優異,而 Transcend MLC 則敬陪末座.這樣的測試看似非常完整有代表性,但實際上固態硬碟的效能不一定會這樣的測試結果一致,因為這兩個典型樣式不足以代表真實的存取行為,我們將使用我們的 Benchmark Suite 的測試證明這個現象.

6.5.2 Results of Benchmark Suites

為了證明管理方法對實際效能有決定性的影響,我們選擇 Buffer, GC 以及 Mapping 三個 Suite 對 MTRON , Samsung , OCZ, 以及 Transcend SLC 的管理方法進行測試, Transcend MLC 由於控制器與 Transcend SLC 相同,因此兩者的差別只在於 SLC 與 MLC 體質上造成的效能差異,因此我們以 Transcend SLC 做為代表..我們首先使用 Buffer Suite 釐清 MTRON 與 Samsung 對於寫入緩衝區管理的差異,接下來使用 Mapping Suite 測試每個待測物的位置轉換機制的差異,最後用 GC Suite 中最有挑戰性的 Browser 測試空間回收策略的差異.

當待測物中包含了配備有寫入緩衝的固態硬碟時,所有的寫入動作可能都會先進入緩衝區中,因此我們必須先測試出寫入緩衝的能力,如此才能比較客觀的解讀後續的測試結果.圖(25)是我們以 Buffer Suite 中的 eMule 對 Samsung 以及 MTRON 這兩個固態硬碟的測試結果.

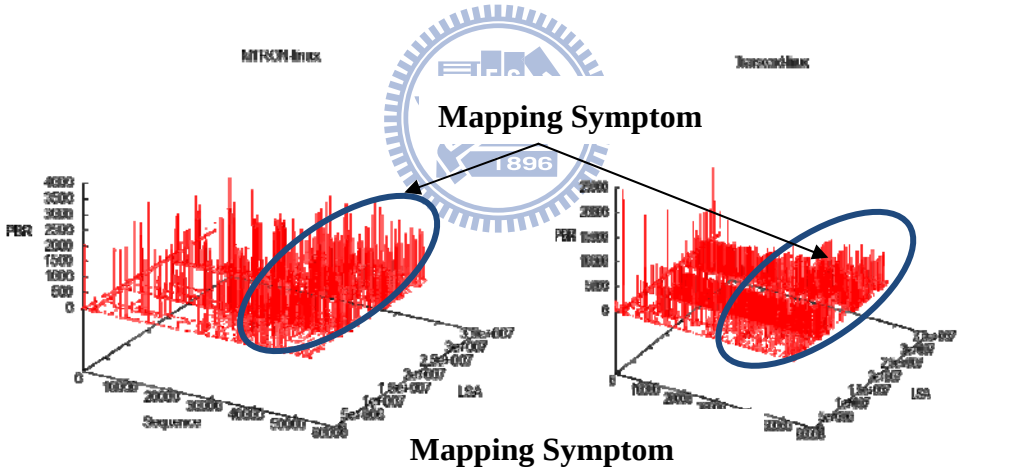


(a) Samsung 的結果,吞吐量為 18MB/s　　(b) MTRON 的測試結果,吞吐量 8MB/sec

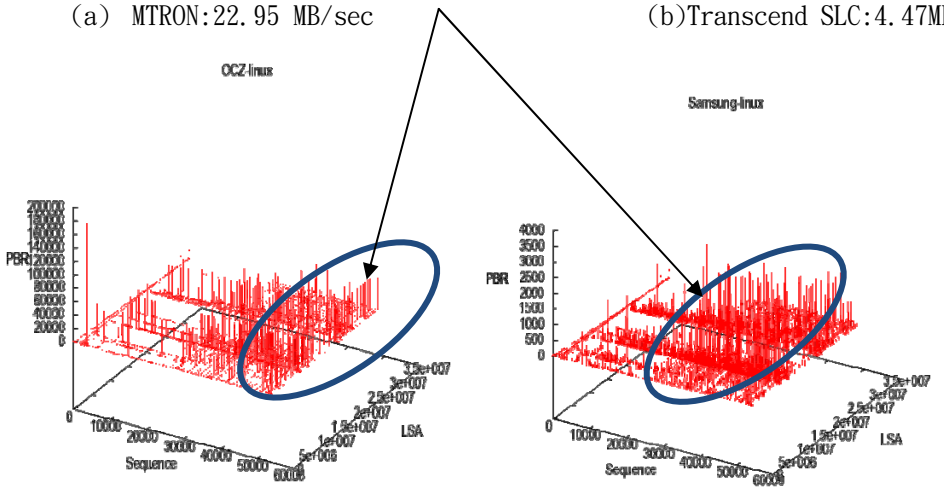
圖(25)Buffer Suite 的測試結果

在 IOMeter 的測試中,MTRON 的效能極為優異,在短的隨機寫入測試中表現也都優於 Samsung.但是當它的寫入緩衝管理的缺點被暴露出來的時候,效能就嚴重的退化,在效能表現上輸給了 Samsung .MTRON 的 PBR 跳動情形是典型的沒有配合底層 FTL 設計的寫入緩衝,由於並沒有減少管理的成本,因為 chunk 引起頻繁寫回的時候容易觸發管理動作而造成效能低落,而 Samsung 的寫入緩衝管理的優勢在此時得以彰顯,從 PBR 的跳動情形看起來可以有效的減低管理成本,可以推論它的管理方法結合了 FTL 的設計.這種差異在 IOMeter 的隨機測試中,並沒有出現空間與時間的區域性,因此無法測試出緩衝區管理方法不同造成的效能差異.我們的測試顯示 Samsung 的寫入緩衝可以減低管理成本,而 MTRON 的實際效能可能會因為寫入緩衝失效而嚴重下滑.

位置轉換的方法以及所使用的單位,可能會影響空間回收所需要的成本,且邏輯位置轉實體位置是資料要寫到快閃記憶體上的第一個動作,因此在測試空間回收之前,必須先了解待測物的位置轉換機制.圖(26)是我們以 Mapping Suite 中的 Install Linux 對四個待測物進行測試的結果.



(a) MTRON:22.95 MB/sec (b)Transcend SLC:4.47MB/sec

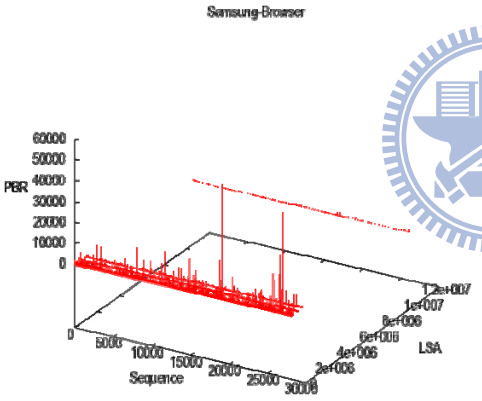


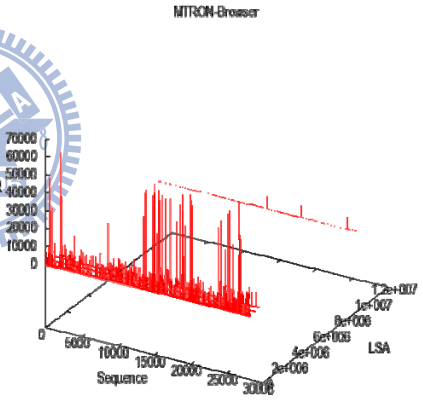
(c) OCZ: 5.95 MB/sec　　(d) Samsung: 34.36 MB/sec

圖(26)Mapping Suite 的測試結果

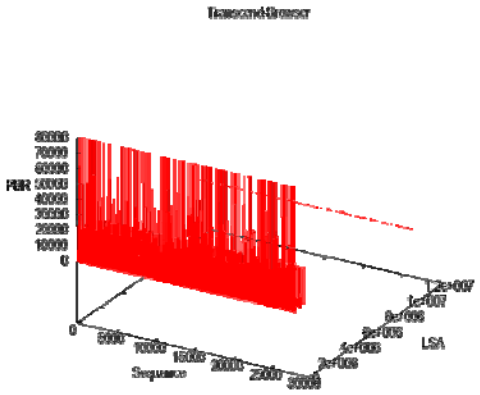
四個待測物都顯現了空間利用度不足的典型症狀,如圖(26)中的標示,這表示它們的位置轉換方法與單位類似,無法有效處理散亂的寫入動作,不過在這一段的整體效能還是因為寫入緩衝的介入而有顯著的差異,Samsung 和 MTRON 明顯勝過 Transcend SLC 與 OCZ,而 Samsung 又優於 MTRON 與 Buffer Suite 的結果吻合.透過這個測試我們可以得知,現階段的固態硬碟位置轉換的機制皆使用粗糙的轉換單位,而這種做法可能不適合 Linux 的使用環境,因為 EXT2/EXT3 的 Headers 可能會引起大範圍的散亂寫入動作造成空間利用度不佳的問題.

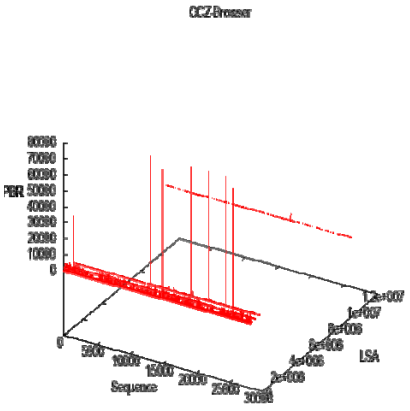
在了解待測物的緩衝區管理以及位置轉換機制的差異後,最後進行的是空間回收的測試.我們直接選擇 GC Suite 中最具有挑戰性的 Browser 測試我們的固態硬碟,測試的結果如圖(27)所示.





(a) Samsung: 19.6 MB/sec　　(b)MTRON: 16.7 MB/sec



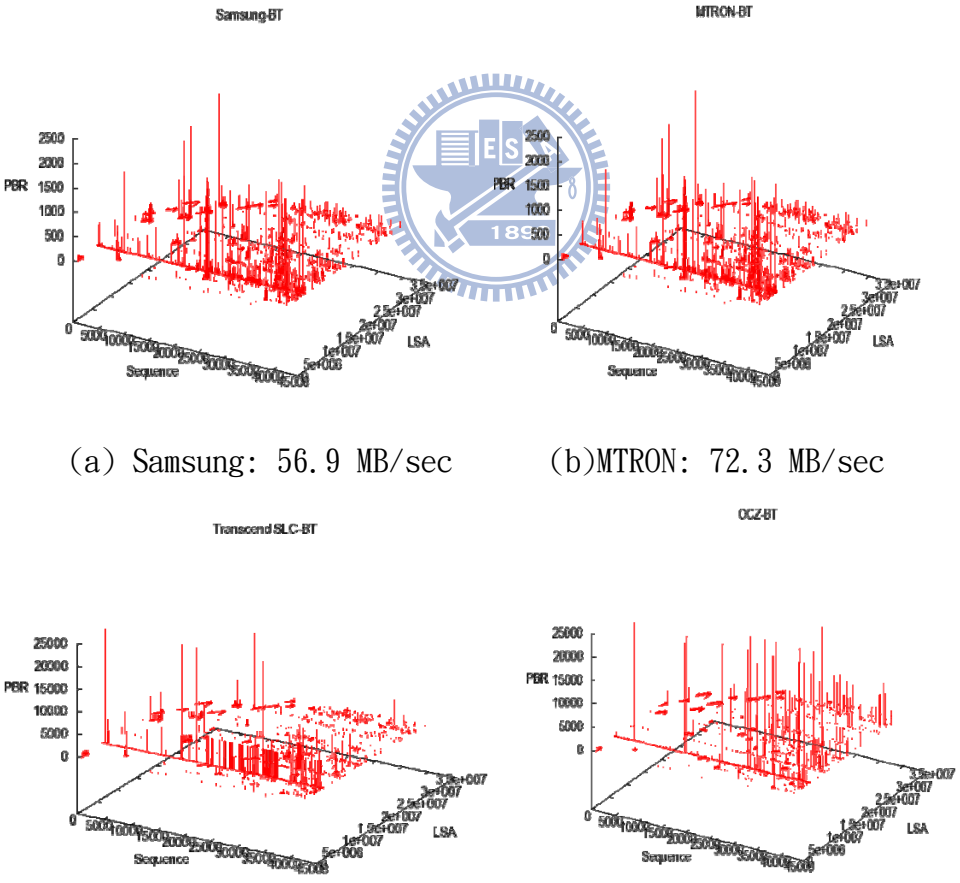


(c)Transcend SLC:0.98 MB/sec　　(d)OCZ: 9.5 MB/sec

圖(27)GC Suite 的測試結果

在 IOMeter 中隨機測式的測試結果顯示 Transcend SLC 的效能優於 OCZ,但是在我們的 GC Suite 測試中卻呈現相反的情形.Transcend SLC 的空間回收機制頻繁的啟動,表示 Browser 的更新強度超過了它的空間回收機制的能力極限.OCZ 的固態硬碟可以有效的減少空間回收啟動的頻率,但是每次啟動的成本很高,表示其對冷熱資料分離的能力,仍然有待加強.MTRON 的空間回收機制策略可能與 Transcend SLC 相去不遠,只是它多了寫入緩衝的幫忙所以數據比較優異.Samsung 仍然是待測物中最優秀的,可以應付高強度的更新以及冷熱資料混合在一起的艱困環境.這個測試中體現了典型的存取樣式完全無法得到的結果,這些存取樣式並不喜歡小範圍的密集更新,因為對於硬碟來說這樣的移動成本及小,但是這卻可能是固態硬碟的效能瓶頸所在.

我們選用 Transfer Suite 中的 BT 進行傳輸速度的評測,圖(28)是我們的測試結果.



(a) Samsung: 56.9 MB/sec　　(b)MTRON: 72.3 MB/sec

(c) Transcend SLC:12.8MB/sec　　(d)OCZ: 23.6 MB/sec

圖(28)BT 的測試結果

雖然圖中呈現有異常的PBR分佈,這是因為檔案系統的寫入動作而不

是BT本身.整體的效能還是決定於循序且資料長度較長的寫入動作.測試的結果與IOMeter的結果吻合,不論是效能的優劣關係以及讀數.我們可以拿BT和eMule比較,BT因為有寫入緩衝的幫忙而使效能在固態硬碟上優於 eMule,而其使用的策略是將chunk收集後再一次寫回,這種做法有利於快閃記憶體的管理因為減少空間回收的資料搬移成本,類似為減輕管理成本而以erase unite為管理單位的固態硬碟寫入緩衝管理方法.這也是我們將eMule分類到Buffer Suite中的原因之一.整體而言BT的行為就是循序與均勻隨機的寫入動作混合,但是都以資料傳輸長度較長,因此對於管理造成的負擔不大,所以關鍵在於傳輸的速度,檔案系統的存取並無決定性的影響.在只測試傳輸速度的情況下,效能的讀數及關係又貼近IOMeter的結果.

6.5.3 Discussion

IOMeter的測試結果顯示,在循序寫入方面效能依照優劣排序為MTRON, Samsung ,OCZ ,Transcend SLC,隨機的部分則是MTRON ,Samsung ,Transcend SLC,OCZ.但是我們使用Buffer Suite, Mapping Suite,和GC Suite的結果顯示,這四個待測物的優劣關係可能因為Workload不同而呈現差異,主要的原因是我們的Workload中包含了small hot data的存取行為.

在隨機寫入的測試中,Transcend SLC大幅度的勝過OCZ,但是當我們使用Mapping Suite以及GC Suite的測試時,結果卻是相反的.最主要的原因是我們的Workload包含了Small Hot data.在GC Suite中,我們用密集的small hot data混合cold data對固態硬碟的空間回收進行測試,要處理好這個Workload必須要能夠把回收啟動的時機往後延遲,累積足夠的無效資料可以減少資料搬移的次數並且提升每次抹除回收的空間,除此之外,如果具備有辨別Hot和Cold data的能力並且將它們分離,又可以更進一步的減少資料的搬移而提升效能.OCZ的固態硬碟雖然是MLC,但是因為有良好的策略,所以在效能上大幅度的勝過了Transcend SLC.在Mapping Suite的測試部分,大範圍且散亂的small hot data可以考驗位置轉換機制的空間利用的精細程度,精細的轉換因為可以任意搬移資料而提升空間回收的效率,若是單位較粗糙的話可以透過調整Data Block與Log Block對應的比例提升區塊的利用率.在這項測試中OCZ的效能仍然勝過Transcend SLC,表示其轉換機制可以比較有效的利用空間.

前面的討論可以知道,固態硬碟的效能關鍵在於處理Small hot data的能力,配備RAM為緩衝的固態硬碟應該要發揮RAM的優勢幫助FTL處理

Small Hot Data.在 Buffer Suite 的測試中,Samsung 的效能明顯優於 MTRON,與 IOMeter 的結果完全不同.在這項測試中,體現出了緩衝管理策略的重要性MTRON 的緩衝區雖然容量比較大,但是並沒有搭配可以幫助快閃記憶體管理的寫回策略,只能利用 RAM 較短的存取時間降低每次寫入所需的時間,但是並沒有辦法有效的減少快閃記憶體管理所需要的資料搬移以及抹除的成本,因此當寫入的強度超過緩衝吸收的範圍而頻繁寫回時,效能會與緩衝區還有空間實呈現很大的落差.因此,固態硬碟的寫入緩衝寫回策略應要配合 FTL,減輕管理成本才是提升效能的根本之道,也就是要能夠吸收 Small Hot data 並且減少散亂的程度,容量的大小其實並不是最重要的因素.

7. Conclusion and Future Work

目前的效能評測方法與工具,因為將待測物設定為機械式的磁碟,其典型的循序與隨機存取的測試樣式,並沒有辦法囊括真實的所有存取行為,除此之外,其使用的效能指標並沒有辦法呈現固態硬碟的管理成本,因此現階段的測試工具所得之結果並不可靠.

為了改善此問題,我們收集了常見的 Workload 搭配 Trace-Replay 的方式營造貼近真實的存取行為,並且提出新的 Per-Byte-Response 效能指標可以具體的辨認出典型的固態硬碟管理不良的情形.為了讓使用者可以直接利用我們的研究成果,我們將收集到的 Workload 分門別類形成不同的 Benchmark Suites.使用者想要測試固態硬碟管理的某個環節時,可以直接取用對應的 Benchmark Suite.我們以其中三個與固態硬碟管理直接相關的 Suite 對四顆固態硬碟進行效能評測,得到了與傳統工具不同的結果,這顯示了固態硬碟的管理方法對實際效能有關鍵性的影響.

我們提出的效能指標 Per-Byte-Response 可以有效的找出管理方法的典型缺失,但是要發展為成熟可用且廣被接受的工具,必須要提出更明確清楚的評定管理方法優劣機制以及擴增 Benchmark Suite 的規模.未來我們希望以 Per-Byte-Response 為基礎,打造一個評分的機制,這個機制除了評定外部效能的優劣外也要能夠呈現管理成本的高低,讓使用者可以得到更完整且易懂的測試結果.在擴展 Benchmark Suite 方面,我們希望未來可以納入 Linux 環境下的 Workload,建構出非常完整且真實的效能評測工具.

Reference

[1]Nitin, A., P. Vijayan, et al. (2008). Design tradeoffs for SSD performance.

USENIX 2008 Annual Technical Conference on Annual Technical Conference. Boston, Massachusetts, USENIX Association.

[2]Po-Chun, H., C. Yuan-Hao, et al. (2008). The Behavior Analysis of Flash-Memory Storage Systems. Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing, IEEE Computer Society.

[3] Luc Bouganim., et al., uFLIP: Understanding Flash IO Patterns, 4th *Biennial Conference on Innovative Data Systems Research (CIDR)* January 4-7, 2009, Asilomar, California, USA.

[4]Sang-Won, L., P. Dong-Joo, et al. (2007). "A log buffer-based flash translation layer using fully-associative sector translation." ACM Trans. Embed. Comput. Syst. **6**(3): 18.

[5]Jeong-Uk, K., J. Heeseung, et al. (2006). A superblock-based flash translation layer for NAND flash memory. Proceedings of the 6th ACM \& IEEE International conference on Embedded software. Seoul, Korea, ACM.

[6]Chanik, P., C. Wonmoon, et al. (2008). "A reconfigurable FTL (flash translation layer) architecture for NAND flash-based applications." ACM Trans. Embed. Comput. Syst. **7**(4): 1-23.

[7]Chiang, M. L. and R. C. Chang (1999). "Cleaning policies in mobile computers using flash memory." J. Syst. Softw. **48**(3): 213-231.

[8]Jen-Wei, H., K. Tei-Wei, et al. (2006). "Efficient identification of hot data for flash memory storage systems." Trans. Storage **2**(1): 22-40.

[9]Seungjae, B., A. Seongjun, et al. (2007). Uniformity improving page allocation for flash memory file systems. Proceedings of the 7th ACM \& IEEE international conference on Embedded software. Salzburg, Austria, ACM.

[10]Li-Pin, C., K. Tei-Wei, et al. (2004). "Real-time garbage collection for flash-memory storage systems of real-time embedded systems." ACM Trans.

Embed. Comput. Syst. **3**(4): 837-863.

[11]Heeseung, J., et al., FAB: flash-aware buffer management policy for portable
media players. Consumer Electronics, IEEE Transactions on, 2006. 52(2): p. 485-493.

[12]Hyojun, K. and A. Seongjun (2008). BPLRU: a buffer management scheme for improving random writes in flash storage. Proceedings of the 6th USENIX Conference on File and Storage Technologies. San Jose, California, USENIX Association.

[13] Samsung Elec. 2Gx8 Bit NAND Flash Memory (K9WAG08U1A). 2006.

[14] Samsung Elec. 2Gx8 Bit NAND Flash Memory(K9GAG08U0M-P).2006.

[15] Aleph One Company, “Yet Another Flash Filing System (YAFFS) "'

[16] Linux MTD Project, “Journaling Flash File System (JFFS), Journaling Flash File System 2 (JFFS2), and Journaling Flash File System 2 (JFFS3).”

[17]FUTUREMARK coporatiom ,PCMARK05,
http://www.futuremark.com/products/pcmark05/

[18]BAPCO,SYSMARK,
http://www.bapco.com/products/sysmark2007preview/index.php

[19] Diskmon.http://technet.microsoft.com/en-us/sysinternals/bb896646.aspx

[20] Iometer Project. “Iometer” ,OPEN SOURCE DEVELOPMENT LAB. 2004. www.iometer.org/.