摘要

比較五種分頁替換演算法 (page replacement algorithm)在三種不同資料型態中的表現，演算法分別有 Optimal、FIFO、ARB、LIFO、NRA，資料型態分別有均勻隨機、區域隨機、常態隨機。整體的排名為（差）ARB > FIFO > LIFO > NRA > Optimal（優），ARB 和 LIFO 更新記憶體分頁較公平，故在區域隨機資料表現較佳；而 LIFO 和 NRA 更新記憶體分頁較不公平，故在常態隨機資料表現較佳。

壹、簡介

分頁 (paging)技術應用於虛擬記憶體 (virtual memory)上，頁框 (frame)為實體記憶體 (physical memory)切出來的區塊，分頁 (page)為虛擬記憶體切出來的區塊，而頁框與分頁的大小會一樣，最後分頁與頁框的對應表，就稱為分頁表 (page table)。

作業系統要存取分頁時，會先查詢分頁表此分頁是否存在記憶體中？如果不存在，就會發生分頁錯誤 (page fault)。如果記憶體還有餘裕的空間，分頁就會從硬碟中載入記憶體中，並更新分頁表；如果記憶體已經滿了，就會尋找一個犧牲者 (victim)，移除犧牲者，將新分頁從硬碟載入記憶體中，並更新分頁表。尋找恰當的犧牲者，就是分頁替換演算法 (page replacement algorithm)要處理的問題。

本次實驗會實作五種分頁替換演算法，分別為最佳演算法 (optimal algorithm)、先進先出演算法 (first in first out, FIFO)、後進先出演算法 (last in first out, LIFO) 、Additional reference bits algorithm (ARB)、常態隨機累積演算法 (normal random accumulation, NRA) 。NRA 是我設計的分頁替換演算法，融合 LIFO 和常態分佈 (normal distribution)的概念。本次實驗的資料有三種型態，分別為均勻隨機 (uniform random)、區域隨機 (local random)、常態隨機 (normal random)。

貳、分頁替換演算法 (page replacement algorithm)

一、最佳演算法 (optimal algorithm)

最佳演算法的效能是所有演算法的天花板，犧牲者未來將不會再被使用或未來最久才再被使用。因此最佳演算法要能預知未來分頁被使用的情況，但實際上我們無法預知未來，故此演算法很難被實作與應用。

二、先進先出演算法 (first in first out, FIFO)

FIFO 的核心理念是先來者先出去，故犧牲者為先來者，也就是存在記憶體中最久的分頁。FIFO 可以使用佇列 (queue)來實作，是易實作且實用的演算法。

FIFO 的特性使得佇列中的分頁得以經常被更新，而且所有分頁都有機會被替換。

三、後進先出演算法 (last in first out, LIFO)

LIFO 的核心理念剛好和 FIFO 相反，是先來者後出去，故犧牲者為後來者，也就是存在記憶體中最短的分頁。LIFO 可以使用堆疊 (stack)來實作，也是易實作的演算法。LIFO 的特性使得堆疊中的分頁除了頂端 (top)會頻繁更新外，其餘分頁都不會被更新，因此適合頻繁使用相同分頁的資料型態。

四、Additional reference bits algorithm (ARB)

ARB 顧名思義即額外使用參考位元去決定犧牲者，通常會使用 8 位元並間隔一段時間，更新在記憶體內所有分頁的 ARB。更新方式是用位元運算，每次更新位元會往左移 1 位元，而最近有被參考過的分頁，其最高位元會是 1；而最近沒有被參考的分頁，其最高位元會是 0，因此透過 ARB 的值可以粗略判斷分頁最近是否有被使用過？

五、常態隨機累積演算法 (normal random accumulation, NRA)

此方法是我改良 LIFO 而來的，其核心理念和 LIFO 一致，都是預期資料會頻繁使用相同分頁。LIFO 有個缺點就是，犧牲者永遠是堆疊中的頂端，也就是最後進來的分頁會不斷被替換。因此如果一開始載進來的分頁都是不常使用的，LIFO 是無法把這些不常用的先來者替換掉，故我加入常態分佈的抽樣，使得非頂端的部分有機會被替換掉。

同時為了讓越靠近堆疊底部 (bottom)的分頁是使用次數越多的，也導入計數器 (counter)的機制。分頁被使用時，計數器會加 1；分頁成為犧牲者時，計數器會減 1，所以犧牲者一定是計數器為 0 者，且越頂端越容易成為犧牲者。

參、實驗結果

一、參數設定

（一）資料

Uniform		Local		Normal	
Data size	300,000	Data size	300,000	Data size	300,000
Ref. size	1,200	Ref. size	1,200	Ref. size	1,200
Dirty rate	0.5	Dirty rate	0.5	Dirty rate	0.5
Ref. range	1~20	Ref. range	1~100	Set size	30
		Sub rate1	1 / 300	Mean	Ref. size / 2
		Sub rate2	1 / 120	S.D.	Ref. size / Set size

（二）分頁替換演算法 (page replacement algorithm)

Optimal		FIFO		LIFO		ARB		NRA	
Mem. size		30, 60, 90, 120, 150							
						Interval	Set size	Mean	Mem. size
								S.D.	Mem. size / Set size

二、資料型態

資料中每一行代表一筆資料，一行中會有兩欄資料，分別為被使用到的分頁索引值 (1~ Ref. size)與修改值 (0/1)，1 代表有被修改，0 代表沒被修改過。在產生資料前，會自定義修改率 (dirty rate)，因此每產生一個索引，就會隨機產生 0~1 間的一個值，如果此值 ≤ 修改率，則為 1；如果此值 > 修改率，則為 0。

（一）均勻隨機 (uniform random)

均勻隨機使用均勻分佈的隨機抽樣產生被使用的分頁索引頭與區間，再以索引頭為開頭，連續取樣區間內的所有分頁。例如：第一次索引頭為 130、區間為 10，故第一組資料為 130~139；第二次索引頭為 500、區間為 15，故第二組資料為 500~514；以此類推，直到產生到指定的資料大小。

（二）區域隨機 (local random)

區域隨機也是採用均勻分佈的隨機抽樣產生被使用的分頁索引頭與區間，不過不是使用連續取樣，而是在這區間中繼續均勻隨機抽樣。為了製造區域性(locality)，我們會界定每組區域的大小。例如：第一組區域的大小為 1,100、索引頭為 300、區間為 50，故第一組資料會從 300~349 中均勻抽樣 1,100 筆資料；第二組區域的大小為 1,500、索引頭為 700、區間為 40，故第一組資料會從 700~339 中均勻抽樣 1,500 筆資料；直到產生到指定的資料大小為止。

（三）常態隨機 (normal random)

常態隨機是使用是採用常態分佈的隨機抽樣直接產生分頁索引值，常態分佈的平均值為 Ref. size / 2，標準差為 Ref. size / Set size，例如：平均值為 600、標準差為 40，依照 68-95-99.7 原則，有 68%的資料會落在 600±40 內；有 95%的資料會落在 600±80 內；有 99.7%的資料會落在 600±120 內。

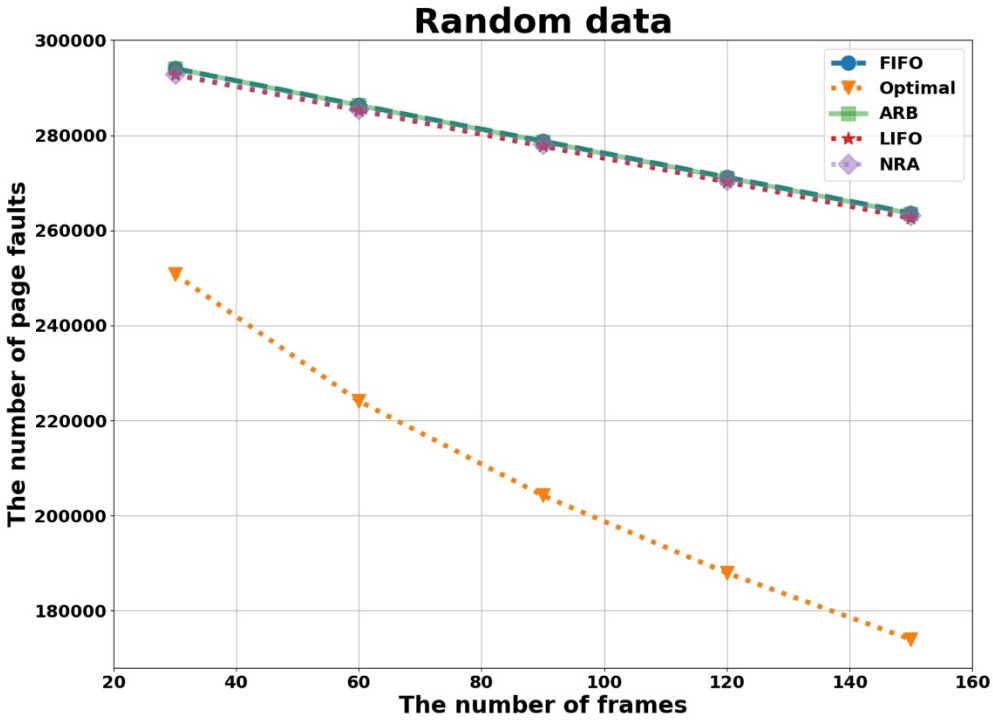
三、實驗結果

本次實驗有五個演算法、三組資料、五種記憶體大小 (30, 60, 90, 120, 150)，且實驗重複 30 次。使用三個指標來代表效能，分別為分頁錯誤 (page fault)、中斷 (interrupt)、寫回 (write back)。

本實驗分頁錯誤定義為只要在記憶體中找不到欲存取的分頁，即會產生一次分頁錯誤。本實驗中斷定義為需寫回或需更新 ARB 時，才會產生一次中斷，故依照此定義，有分頁錯誤不一定會產生中斷，但有寫回則一定產生中斷。本實驗寫回定義為，如果發生分頁錯誤且該分頁被修改過，則會產生一次寫回。因此本實驗預設的寫回方式不是立即寫回，而是當發生分頁錯誤，該分頁要被替換掉且被改過時，才會執行寫回。

（一）均勻隨機

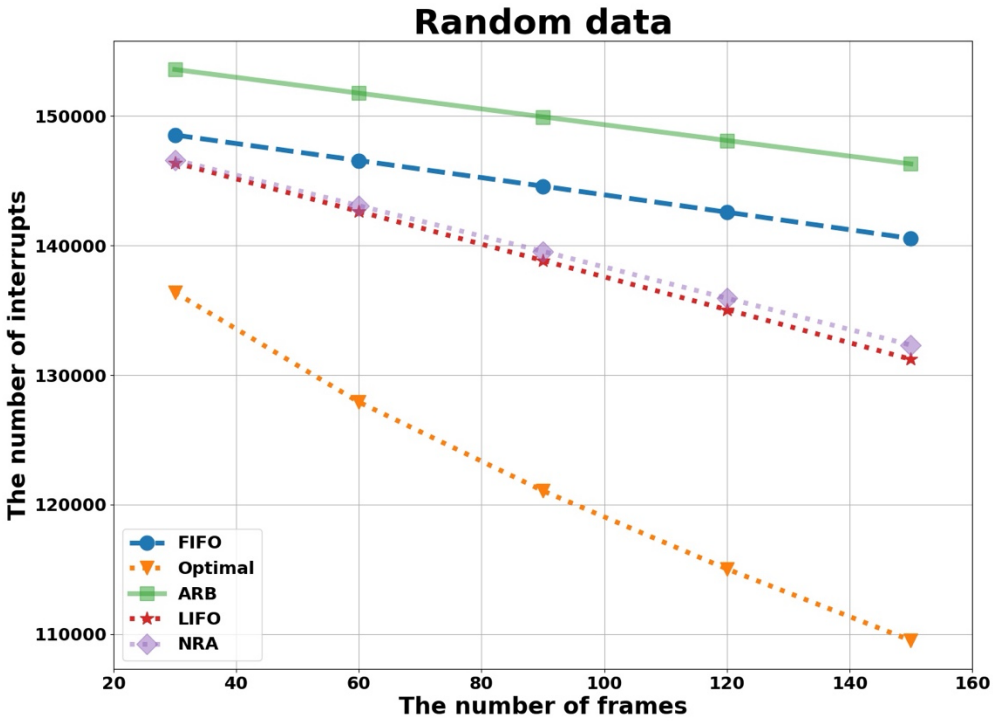
1. 分頁錯誤



（圖一）均勻隨機資料中各演算法分頁錯誤數與頁框大小關係圖

圖一顯示在均勻隨機資料的分頁錯誤數：ARB = FIFO = NRA = LIFO > Optimal，除了 Optimal 外的其餘四個演算法，在均勻隨機資料中的分頁錯誤數並無明顯差異，而且 Optimal 的斜率是比其他四個演算法更斜。

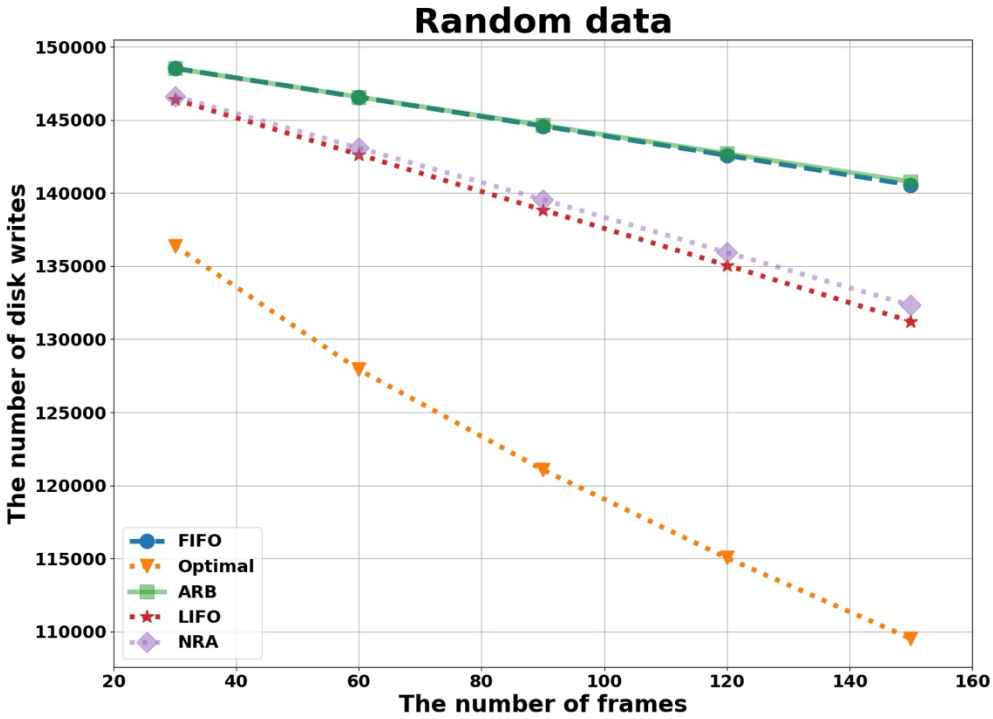
2. 中斷



（圖二）均勻隨機資料中各演算法中斷數與頁框大小關係圖

圖二顯示在均勻隨機資料的中斷數：ARB > FIFO > NRA > LIFO > Optimal，雖然前四個的分頁錯誤數並無明顯差異，但在中斷數就可以看到差異。ARB 最高是因 ARB 的中斷還包含更新 ARB，而 NRA 次低，LIFO 最低，是因為 LIFO 中堆疊中非頂端部分，都不會被替換出去，所以為最低，而 NRA 非頂端部分仍有機會被替換出去，因此次低。

3. 寫回

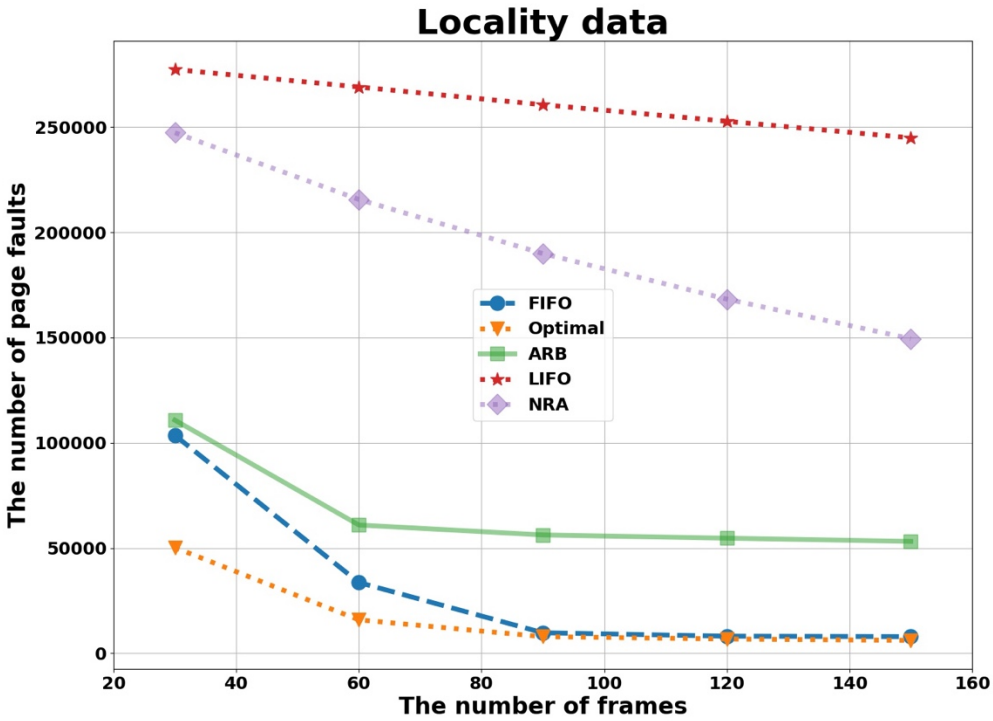


（圖三）均勻隨機資料中各演算法寫回數與頁框大小關係圖

圖三顯示在均勻隨機資料的寫回數：ARB = FIFO > NRA > LIFO > Optimal，圖二和圖三的差異只在於 ARB，其他四條線都相同，因為本實驗中斷的定義為寫回和更新 ARB，所以其他四個演算法的中斷數=寫回數，而 ARB 則會不同。

（二）區域隨機

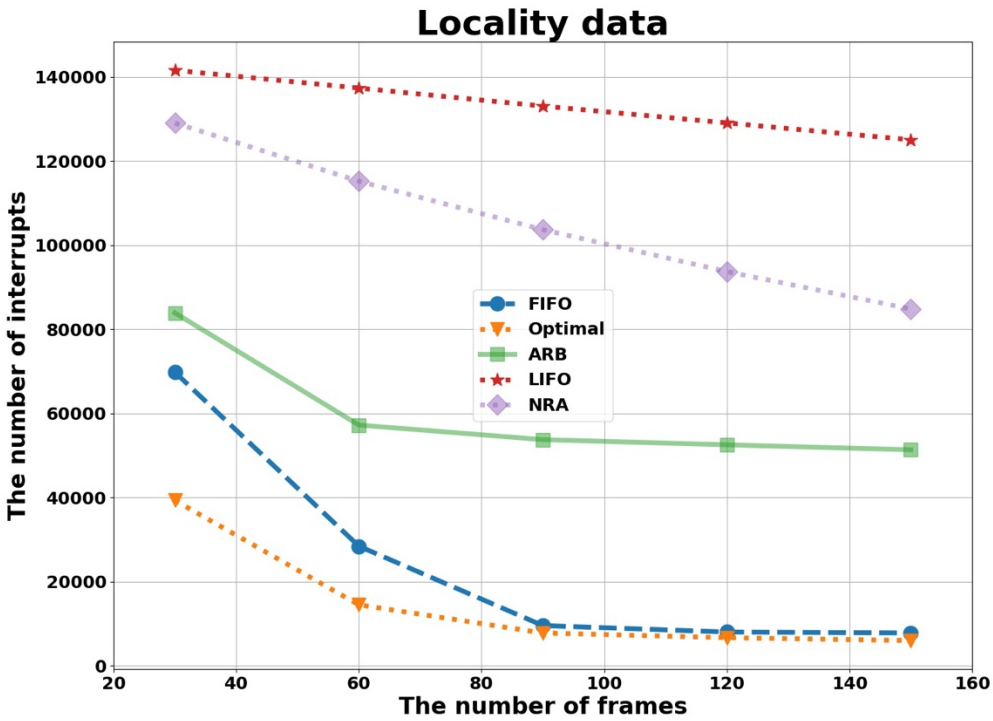
1. 分頁錯誤



（圖四）區域隨機資料中各演算法分頁錯誤數與頁框大小關係圖

圖四顯示在區域隨機資料的分頁錯誤數：LIFO > NRA > ARB > FIFO > Optimal，由於 LIFO 只會更新堆疊的頂端，所以在區域性資料中表現最差，NRA 有機會更新到堆疊的非頂端，所以表現次差。ARB 和 FIFO 在 30 頁框時相近，但當頁框增加時，FIFO 的改善比 ARB 更優，甚至到 90 頁框後快逼近 Optimal，因為 FIFO 替換分頁機率較為公平，而 ARB 替換分頁機率較不公平，所以當區域轉移時，FIFO 能更快適應新區域的分頁。

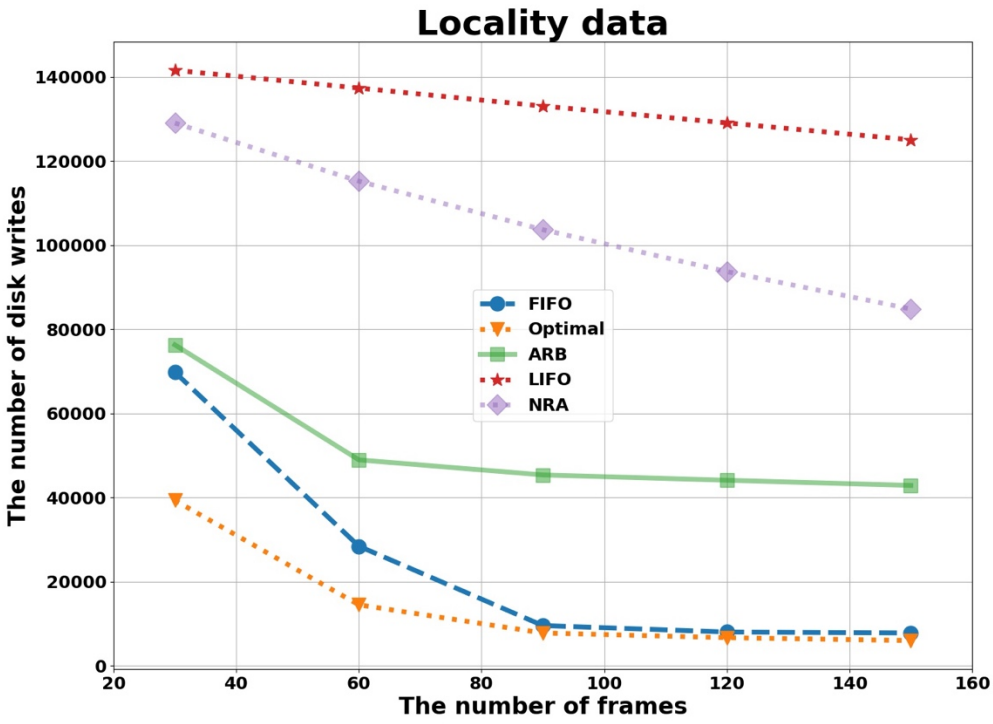
2. 中斷



（圖五）區域隨機資料中各演算法中斷數與頁框大小關係圖

圖五顯示在區域隨機資料的中斷數：LIFO > NRA > ARB > FIFO > Optimal，其原因與圖四的說明相同。記憶體中的分頁，替換機率越不公平，就越難適應新的區域，因此表現就會越差；替換機率越公平，就越易適應新的區域，因此表現就會越好。

3. 寫回

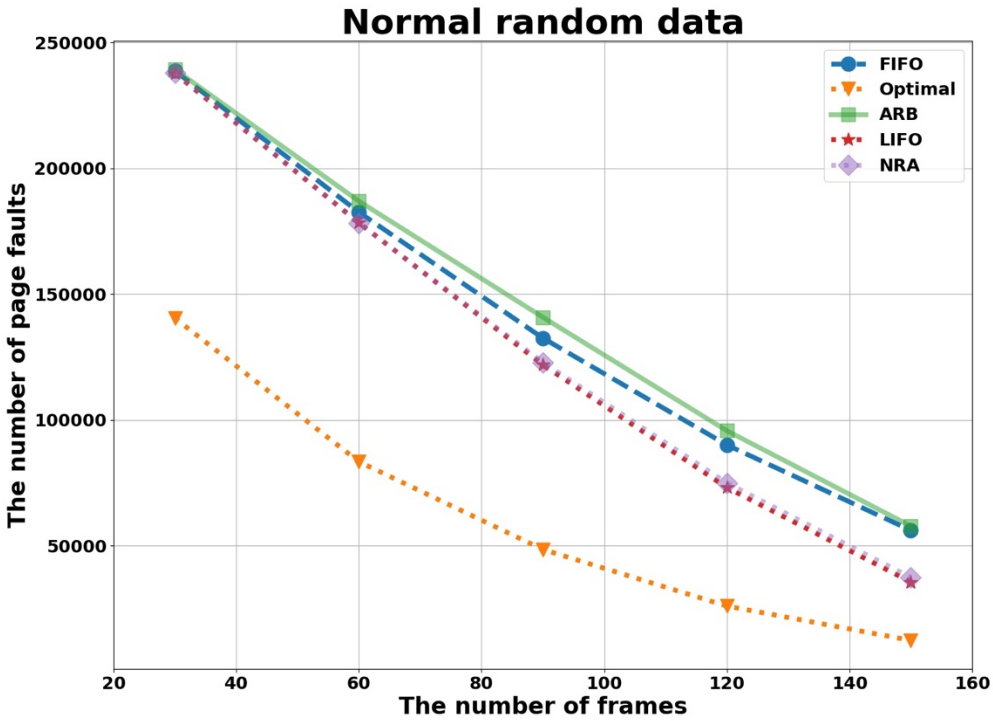


（圖六）區域隨機資料中各演算法寫回數與頁框大小關係圖

圖六顯示在區域隨機資料的寫回數：LIFO > NRA > ARB > FIFO > Optimal，圖五和圖六的差異只在於 ARB，其他四條線都相同。

（三）常態隨機

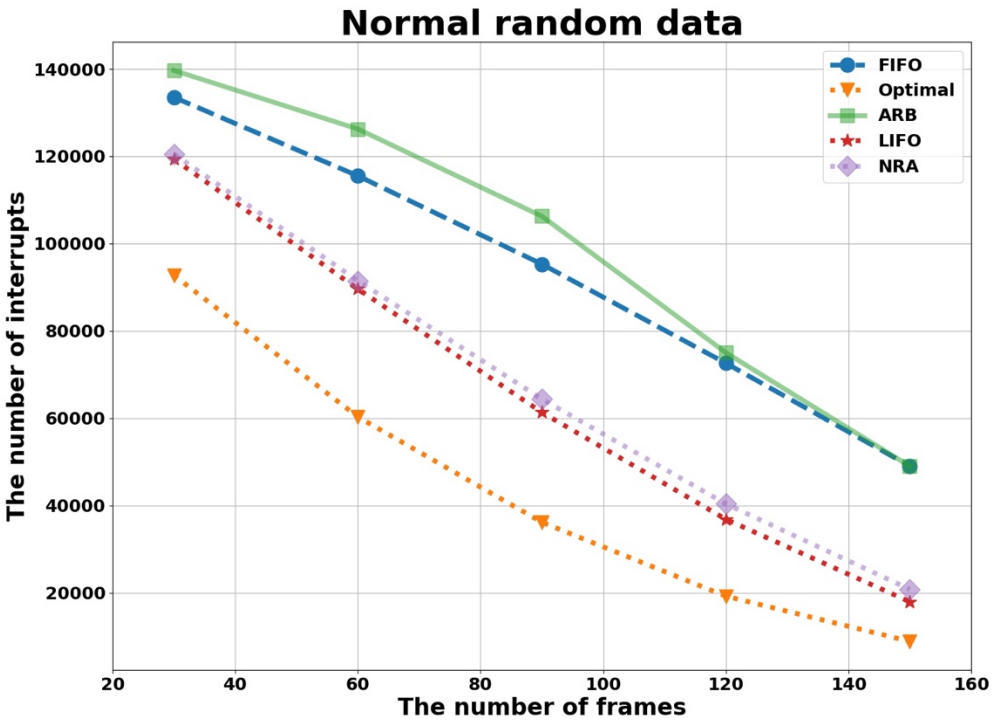
1. 分頁錯誤



（圖七）區域隨機資料中各演算法分頁錯誤數與頁框大小關係圖

圖七顯示在常態隨機資料的分頁錯誤數：FIFO ≥ ARB > NRA = LIFO > Optimal，30 頁框時，前四種表現差不多；中間 90 頁框時，FIFO 有比 ARB 好一點，但後來 FIFO 斜率趨緩；到 150 頁框時，就分成兩群。整體而言，Optimal 的斜率趨緩，LIFO 和 ARB 斜率比 FIFO 和 ARB 陡，因為 LIFO 和 ARB 本身就比較適合經常使用同樣分頁的資料，故表現會比較好。

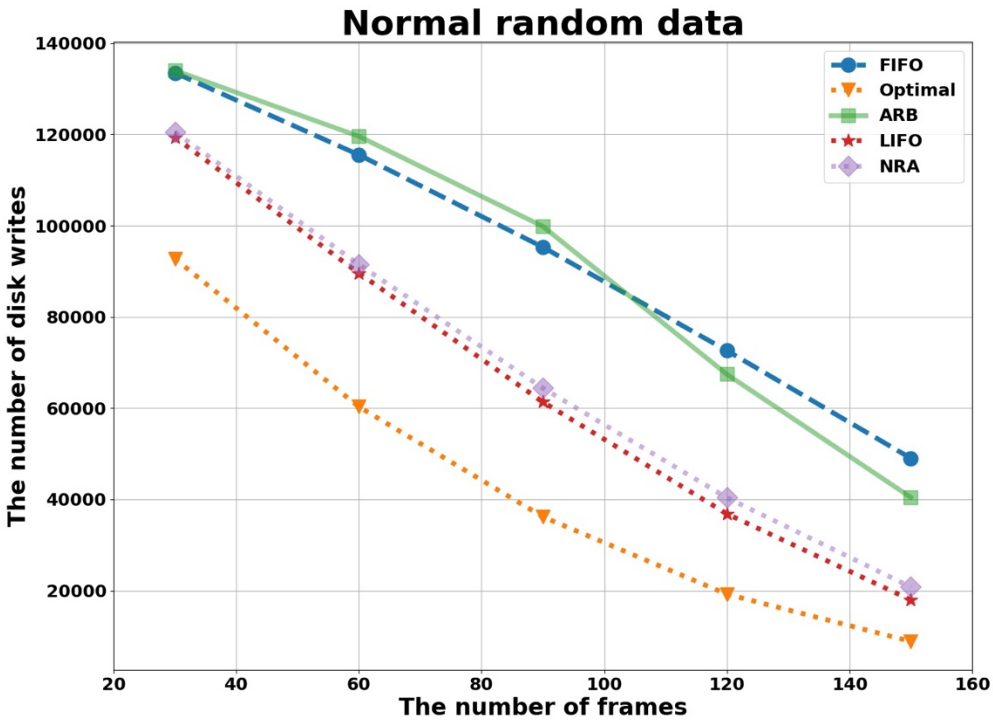
2. 中斷



（圖八）區域隨機資料中各演算法中斷數與頁框大小關係圖

圖八顯示在常態隨機資料的中斷數：ARB ≥ FIFO > NRA ≥ LIFO > Optimal，和圖二說明一樣，由於 ARB 和 FIFO 更新頻繁，所以中斷較多；而 NRA 和 LIFO 更新較不頻繁，所以中斷較少。

3. 寫回



（圖九）區域隨機資料中各演算法寫回數與頁框大小關係圖

圖九顯示在常態隨機資料的寫回數：ARB ≥ FIFO > NRA ≥ LIFO > Optimal，圖八和圖九的差異只在於 ARB，其他四條線都相同。

四、結論

（表一）各演算法在不同資料型態的排名

Performance rank		Optimal	FIFO	ARB	LIFO	NRA
Uniform	Page fault	1	2	2	2	2
	Interrupt	1	4	5	2	2
	Write back	1	4	4	2	2
Local	Page fault	1	2	3	5	4
	Interrupt	1	2	3	5	4
	Write back	1	2	3	5	4
Normal	Page fault	1	4	4	2	2
	Interrupt	1	4	5	2	2
	Write back	1	4	4	2	2
Sum		9	28	33	27	24

表一顯示整體的排名：ARB > FIFO > LIFO > NRA > Optimal，Optimal 為

效能的天花板。

ARB 在整體表現中是最差的，但在區域隨機資料仍有不錯的效能，而 FIFO 在區域隨機的表現最好，甚至逼近 Optimal，因為 ARB 和 FIFO 替換記憶體中的分頁較為公平，尤其 FIFO 是一視同仁，很快就能適應新區域的轉變。反觀 LIFO 在區域隨機中是最差的，因為它替換分頁的機制很不公平，永遠都是先替換新來者，因此它無法去適應新區域。然而 FIFO 這樣的缺點，在 NRA 中得到改善，因此 NRA 在區域隨機中，明顯比 FIFO 有優勢。

在常態隨機資料中，可以看到 LIFO 和 NRA 平分秋色，因此 NRA 雖然改善了 LIFO 的缺點，但同時也沒有失去 LIFO 本身的優點。不過 NRA 受標準差的影響很大，標準差越小，NRA 越像 LIFO；標準差越大，RNA 越像 FIFO。故 NRA 如果再導入自適應參數 (self-adaptive parameter)的概念，讓標準差能隨時間與資料型態的改變，而有所改變，NRA 在區域隨機的資料將會有更顯著的改善。

引用文獻

Magic Len。2015。分頁替換演算法(Page Replacement Algorithm)介紹與模擬。MagicLen [Accessed by Oct. 2021] https://magiclen.org/page-fault/