

高等作業系統 HW1

Page Replacement 演算法和評估

產生 Reference String

Random

首先從 (1, 20) 間產生一個 window size (w)，並在這個區間自範圍 (1, 1200) 中再選擇一個隨機數 start number (n)，依序填入 [n, n+1, n+2 ...] 直到 reference string 長度上限 (300000)。

邊界處理：

- 當填入 reference string 的數字 > reference string 的數字上限 (1200)，重新設定 n 從 1 開始

[5, 6, 7, 8, 9, 10, 24, 25, 26, 98, 99, 1, 2 ...]

Locality

固定一個小 windows size (m)，於作業中設定為 5。接著自範圍 (1000, 2500) 產生一個 window size (w)，並在這個區間先選擇一個隨機數 start number (n)，在一個設定的範圍 (n, n-m) 內再選出要填入的隨機數 picked number (x)，直到 reference string 長度上限 (300000)。

邊界處理：

- 當選中的 n - m 超過 reference string 的下限 (1)，設定 x 以 50/50 的機率在兩種選擇中選取
 - 將 m 範圍向左縮，改自最大可能的 x 中 (1200-m+1, 1200) 選擇
 - 將 m 範圍向右縮，改自最小可能的 x 中 (1, m) 選擇

[... 1, 3, 2, 2, 987, 988, 987, 986, 985, 987 ...]

Sequence and Jump

此為作業要求之自訂 reference string 產生方法，假設在 process 處理 reference string 平常以等差序列 (n, n+1, n+2 ...) 為主，常常會轉到其他的 process 做短期處理。先行定義一般序列 (50, 100) 和跳躍 (5, 10) 時的 window size，在作業中設定有 80% 機率會跳躍，並在個別代表的區間選擇一個隨機數 start number (n)，分別填入 [n, n+1, n+2 ...]。當跳躍時，確保新數字 n 不在新 n 的 ± 100 範圍內。

[... 5, 6, 7, 8, 9, 10, 11, 98, 99, 100, 43, 44, 45, 46, 47 ...]

Page Replacement 演算法

FIFO

使用 First-In-First-Out 的方式替換 frame。

Optimal

透過查看未來的 reference string，若當下選擇的 frame 在未來距離最遠或不再使用則替換該 frame。

Additional Reference Bit

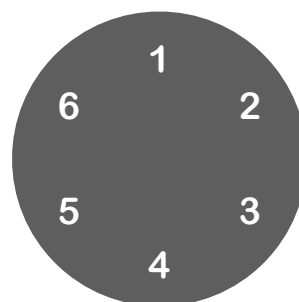
替每個 frame 多紀錄一組 8 bits 的資料（以下稱 info），當 hit 時該 frame info 的 left-most bit 設為 1，並在固定週期（作業中並未使用時間當作 interval，而是設定每 reference 8 次）將 info 向右移一個位元。每當需要替換時，選擇 info 最大者並替換。

Distance

此為作業要求之自訂 page replacement 方式，以檢查距離的方式，並選擇距離較遠者替換。距離的計算方式如下，若要計算 1 到 3 的距離，有兩種方向－順時鐘和逆時鐘，以順時鐘方向計算距離為 2，逆時鐘方向距離為 4，以較短者當作兩者之間的距離。

以 [1, 2, 3, 6, 5, 1, 2, 6] 為例，當 reference 6 時，6 到 1、2、3 的較短距離分別為 1、2、3，因此選擇離最遠的 3 進行替換。

將所有的數字
看作循環的圓形 ►



	1	2	3	6	5	1	2	6
Frame 1	1	1	1	1	1	1	1	1
Frame 2		2	2	2	5	5	2	2
Frame 3			3	6	6	6	6	6
PF	X	X	X	X	X		X	

▲ 以 Distance 進行 page replacement

process 通常會是處理 [n, n+1, n+2 ...] 或是在 n 附近的 instruction，所以這個方式以 references 皆在附近為前提，優先替換掉離最遠的，進而確保接下來的替換都不會距離太遠。

	1	2	3	6	5	1	2	6
Frame 1	1	1	1	6	6	6	2	2
Frame 2		2	2	2	5	5	5	6
Frame 3			3	3	3	1	1	1
PF	X	X	X	X	X	X	X	X

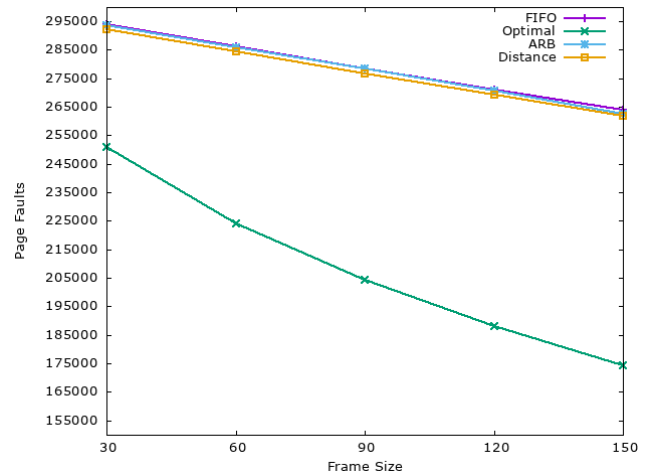
▲ 對比以 FIFO 進行 page replacement

評估

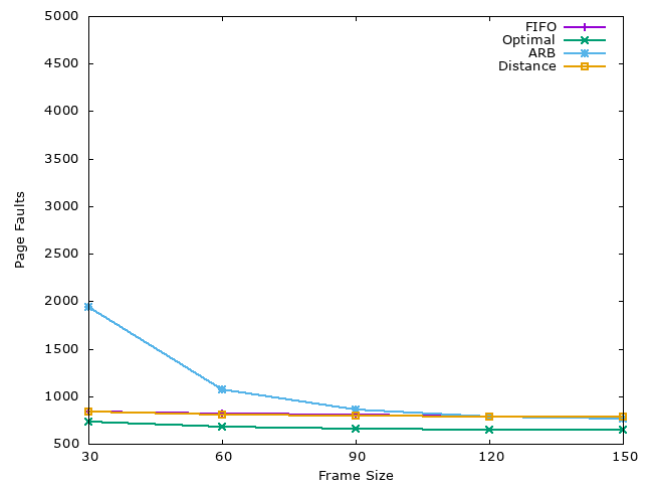
Page Faults

若有 frame 替換的情況則發生 page fault。

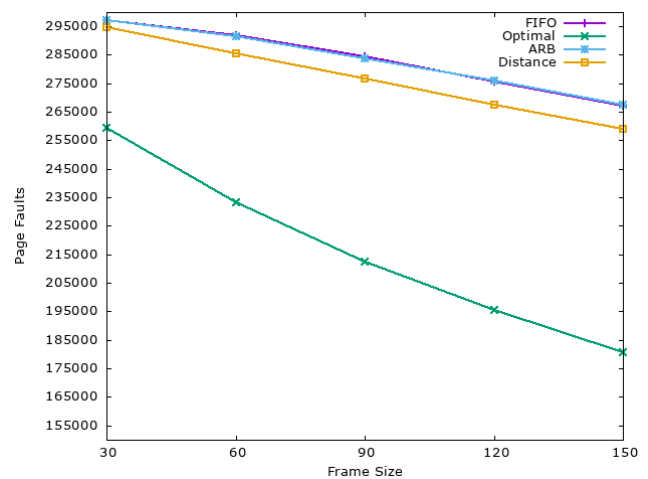
- FIFO
 - 不一定隨著 frame size 增加而大幅減少
 - 可能產生 Belady's anomaly
- Optimal
 - 隨著 frame size 增加而減少
- ARB
 - 隨著 frame size 增加而減少
- Distance
 - 隨著 frame size 增加而減少
 - 結果皆優於 FIFO



▲ Random



▲ Locality

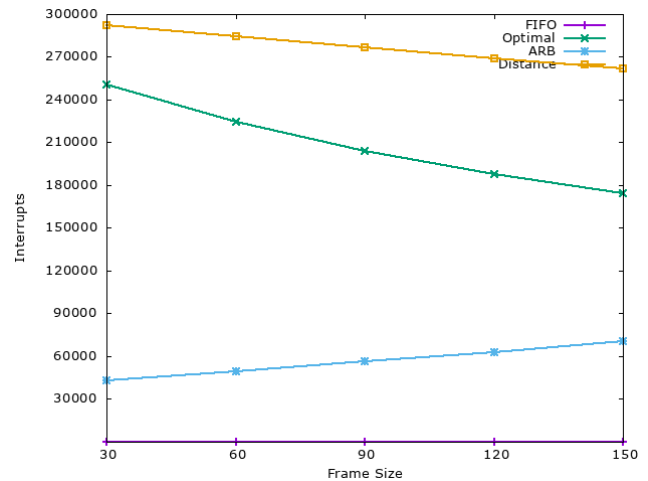


▲ Sequence and Jump

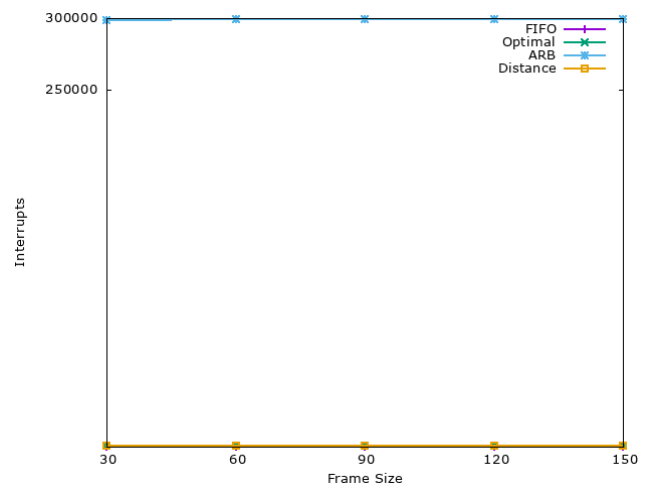
Interrupts

當 OS 需要進行動作時計算，不包含 page faults。

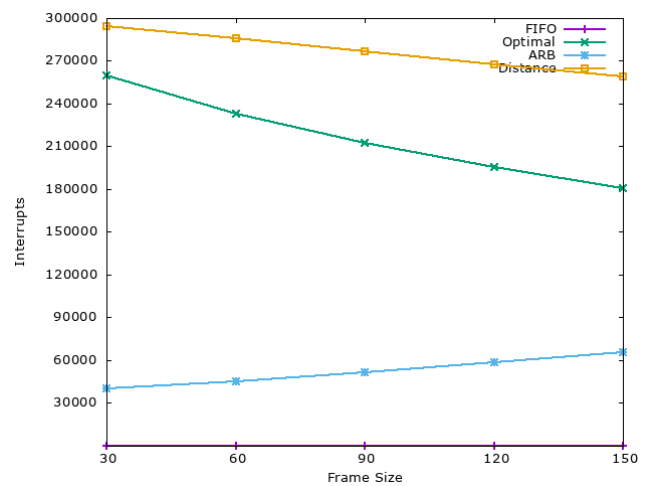
- FIFO
 - 僅單純有置換動作，為 0
- Optimal
 - 需要看未來的 reference string
- ARB
 - 需要執行 shift
- Distance
 - 需要計算與其他 frame 的距離



▲ Random



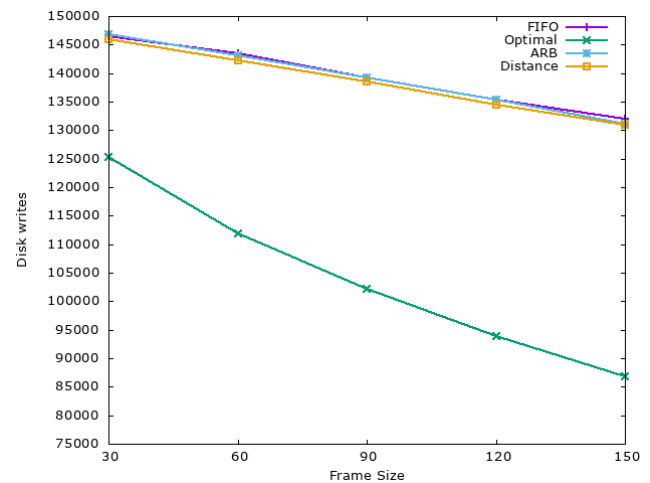
▲ Locality



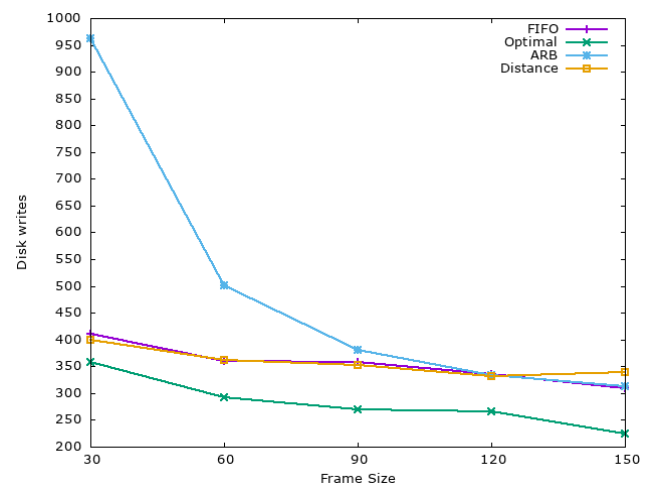
▲ Sequence and Jump

Disk writes

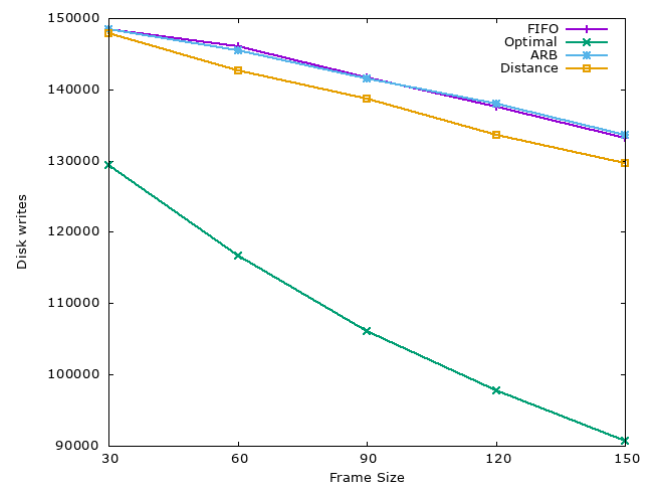
當曾經寫入 frame 的 reference 被寫入發生 disk write，作業並無前後關係，則以機率表示。在作業中以 50% 的機率設定 dirty bit。



▲ Random



▲ Locality



▲ Sequence and Jump