Project Ver_B

Title: The Buffering Effect on the Page Replacement under MLFQ Scheduler

تحذير هام: علي الطالب عدم كتابة اسمه أو كتابة اي شيء يدل علي شخصيته

Answer Structure

1. Methods

1.1 Modified Clock [Without buffering]

Description:

- search for victim page. It gives the page a second chance by check the used bit and modified bit.
- Start from current position of the pointer scan the frame buffer:
  - **Step 1:** scan for search the first frame (0, 0) not used and not modified. (if step 1 fail to find the frame go to step 2).
  - **Step 2:** scan for search the first frame (0, 1) not used and modified, during scanning change used bit from used to not used. (if step 2 fail to find the frame go to step 1).

  (after step 1 fail and step 2 fail go to step 1 again if step 1 fail again finally go to step 2, in this cycle will definitely find the victim page).
- If found the victim page in any step, stop scan and replace it by allocating new frame for fault, move the pointer to the next page.

Example:

	1^{w}	3^{r}	5^{w}	7^{r}	9^{w}	1^{r}	3^{w}	1^{w}	3^{r}	5^{w}
	$1^{[1,1]}$	$1^{[1,1]}$	$\rightarrow 1^{[1,1]}$	$1^{[0,1]}$	$\rightarrow 1^{[0,1]}$	$\rightarrow 1^{[1,1]}$	$1^{[0,1]}$	$1^{[1,1]}$	$1^{[1,1]}$	$\rightarrow 1^{[1,1]}$
	$\rightarrow$	$3^{[1,0]}$	$3^{[1,0]}$	$7^{[1,0]}$	$7^{[1,0]}$	$7^{[1,0]}$	$3^{[1,1]}$	$3^{[1,1]}$	$3^{[1,1]}$	$3^{[1,1]}$
		$\rightarrow$	$5^{[1,1]}$	$\rightarrow 5^{[0,1]}$	$9^{[1,1]}$	$9^{[1,1]}$	$\rightarrow 9^{[0,1]}$	$\rightarrow 9^{[0,1]}$	$\rightarrow 9^{[0,1]}$	$5^{[1,1]}$
fault	✓	✓	✓	✓	✓		✓			✓

1.2 Page Buffering

Description:

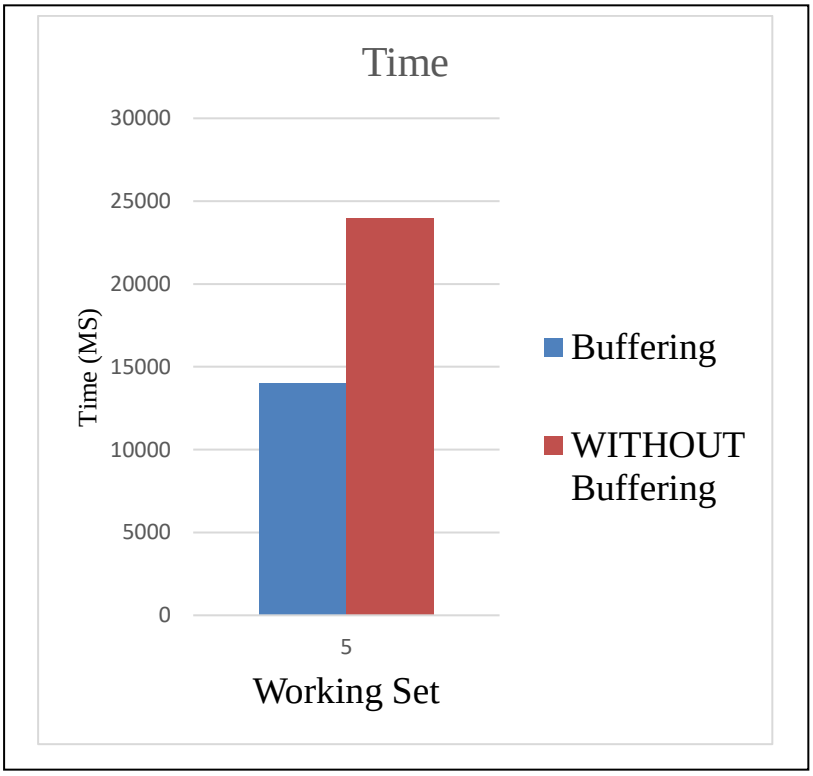
- search for victim page. It gives the page a second chance by check the used bit and modified bit, do process of "step 1" and "step 2" as **Page fault Without buffering.**
- Keep the victim page in **RAM:**
  - If victim page modified insert it in the tail of modified list.
    - If modified list is complete, add it in the tail of free frame list.
  - If victim page not modified insert it in the tail of free frame list.
- If fault **buffered**:
  - If fault modified, remove it from modified list.
  - If fault not modified, remove it from free frame list.
- If fault **not buffered**:
  - Bring the fault from disk by allocating new frame for it, move the pointer to the next page.

Example: [**MUST** be the same example in 1.1 but with enabling buffering]

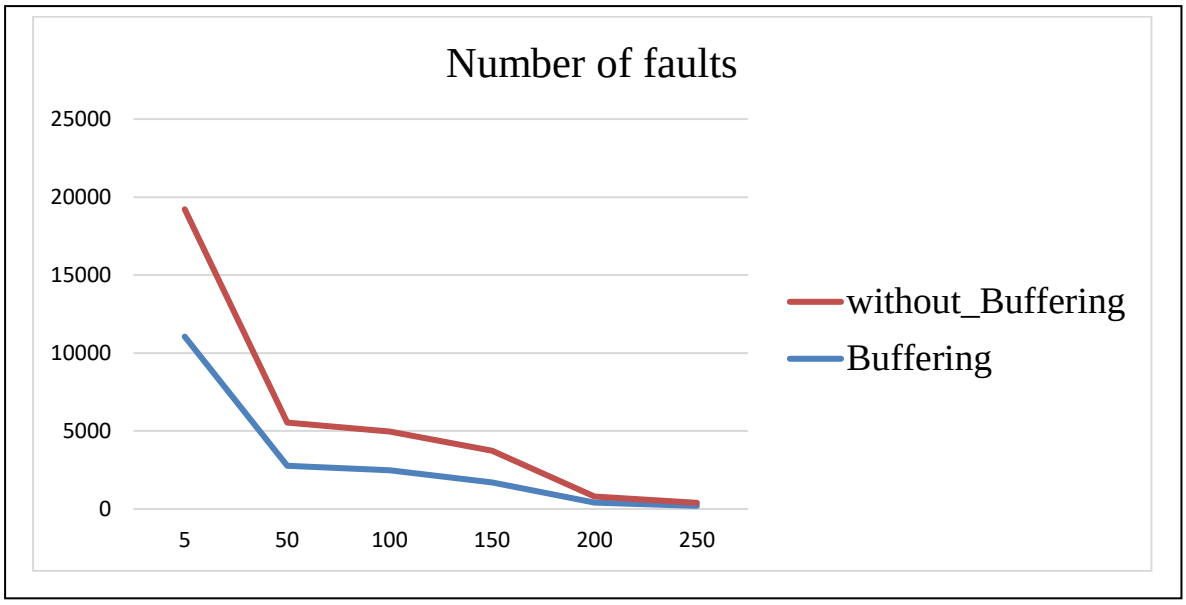
	1^{w}	3^{r}	5^{w}	7^{r}	9^{w}	1^{r}	3^{w}	1^{w}	3^{r}	5^{w}
	$1^{[1,1]}$	$1^{[1,1]}$	$\rightarrow 1^{[1,1]}$	$1^{[0,1]}$	$\rightarrow 1^{[0,1]}$	$\rightarrow 1^{[1,1]}$	$1^{[0,1]}$	$1^{[1,1]}$	$1^{[1,1]}$	$\rightarrow 1^{[1,1]}$
	$\rightarrow$	$3^{[1,0]}$	$3^{[1,0]}$	$7^{[1,0]}$	$7^{[1,0]}$	$7^{[1,0]}$	$3^{[1,1]}$	$3^{[1,1]}$	$3^{[1,1]}$	$3^{[1,1]}$
		$\rightarrow$	$5^{[1,1]}$	$\rightarrow 5^{[0,1]}$	$9^{[1,1]}$	$9^{[1,1]}$	$\rightarrow 9^{[0,1]}$	$\rightarrow 9^{[0,1]}$	$\rightarrow 9^{[0,1]}$	$5^{[1,1]}$
fault	✓	✓	✓	✓	✓		✓			✓
free	F[]	F[]	F[]	$F[3^{(0,0)}]$	$F[3^{(0,0)}]$	$F[3^{(0,0)}]$	$F[7^{(0,0)}]$	$F[7^{(0,0)}]$	$F[7^{(0,0)}]$	$F[7^{(0,0)}]$
modified	M[]	M[]	M[]	M[]	$M[5^{(0,1)}]$	$M[5^{(0,1)}]$	$M[5^{(0,1)}]$	$M[5^{(0,1)}]$	$M[5^{(0,1)}]$	$M[5^{(0,1)}]$

2. Results

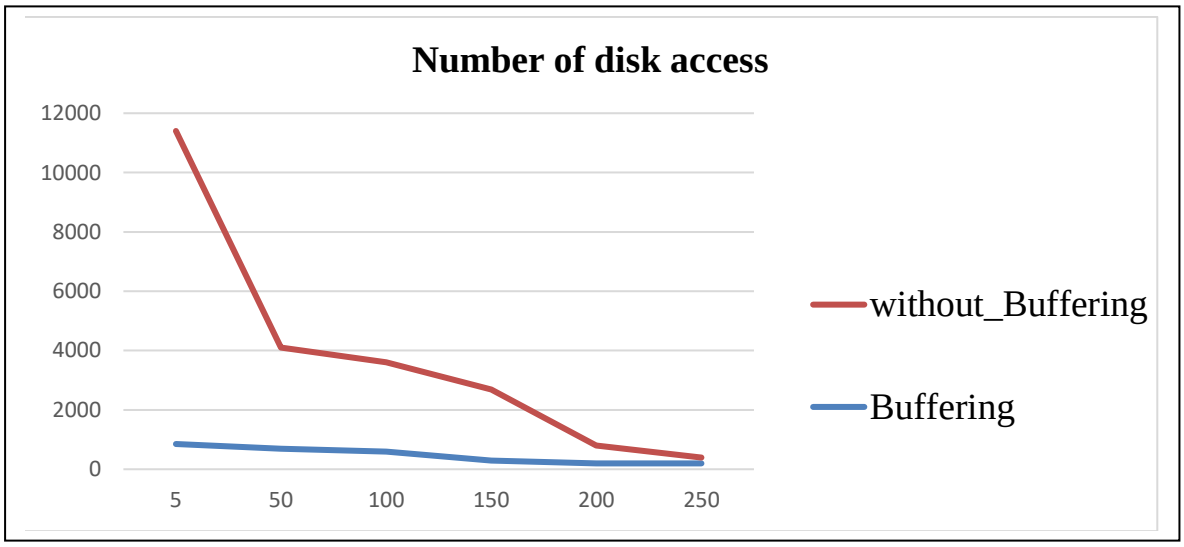
2.1 Graph 1: Time



2.2 Graph 2: Number of faults



2.3 Graph 3: Number of disk accesses



3. Discussion

3.1 Graph 1:

3.1.1 Which is best & why

- **Page Buffer** is best because time taken to take the page faults from desk is less, reduce time to deal with disk.

3.1.2 Which is worst & why

- **Without buffering** is worst because it takes much time.

3.2 Graph 2:

3.2.1 Which is best & why

- **Without Buffer** is best because number of page faults is less.

3.2.2 Which is worst & why

- **Page Buffer** is worst because number of faults is high.

3.3 Graph 3:

3.3.1 Which is best & why

- **Page Buffer** is best because number of disk access is less.

3.3.2 Which is worst & why

- **Without buffering** is worst because it needs many disk accesses.

3.4 The number of disk reads in case of buffering

- changing over the working set size (decreasing)

3.5 Suggestions (if exist):

- Use a new list for not buffered frames, instead of read or write from disk, use this list, after finish all processes write the remaining frames in buffed list on disk. (It will help to reduce time to read from disk, use disk only once).