P3-writeup

Hunsi Jayaprakash

In project 3, I observed how using different page replacement algorithms (LRU and OPT) have different outcomes in performance and can be seen in the different parameters such as number of page faults, memory accesses, and disk writes. In the below graphs, we can observe the performance of LRU and OPT based on different memory splits (x-axis) and the resulting page fault statistics (y-axis). I ran each configuration on the 1.trace file that was provided. In addition to observing memory splits, I also saw how changing the page size and frame number influence the number of page faults for a given page replacement algorithm. I wasn't able to graph my OPT statistics with great certainty but based on the page size, frame, and memory split I estimated how OPT page replacement algorithm might perform.

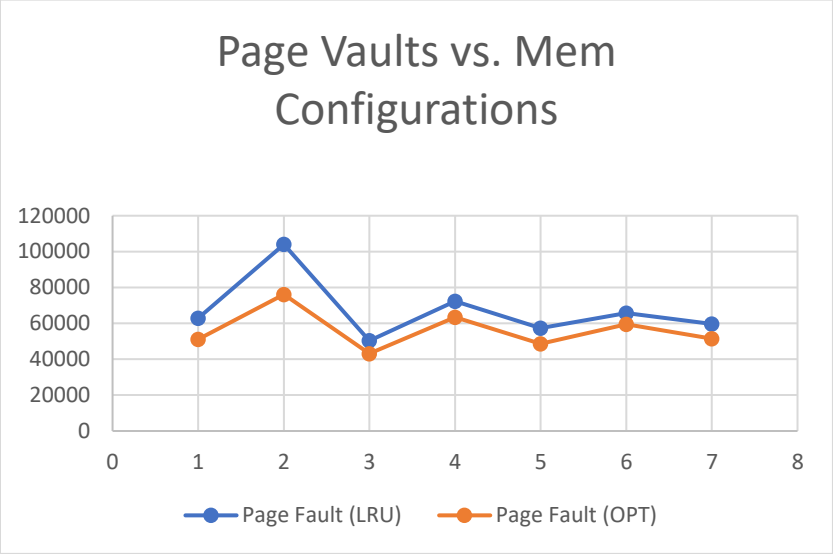

Figure 1. 16 frames and a page size of 4 KB

Memory split: 1:1, 1:3, 3:1, 3:5, 5:3, 7:9, 9:7

Page replacement algorithms: LRU and OPT

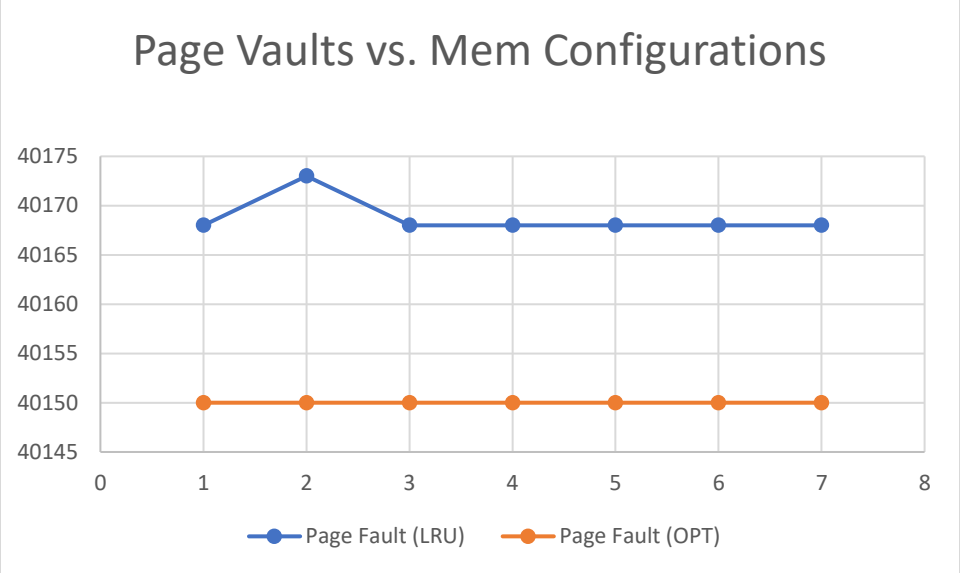

Figure 2. a total of 1024 frames and a page size of 4KB

Memory split: 1:1, 1:3, 3:1, 3:5, 5:3, 7:9, 9:7

Page replacement algorithms: LRU and OPT

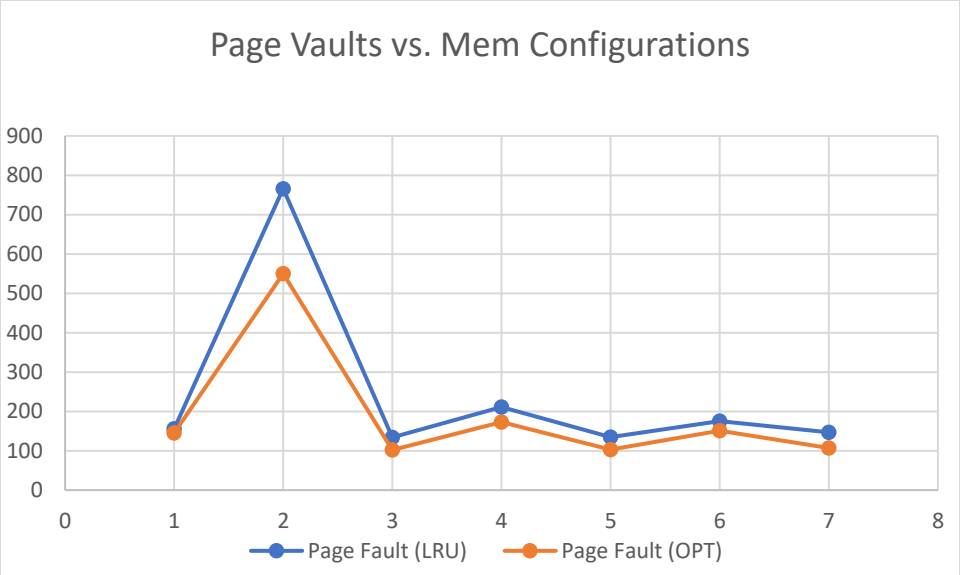

Figure 3. a total of 16 frames and a page size of 4 MB

Memory split: 1:1, 1:3, 3:1, 3:5, 5:3, 7:9, 9:7

Page replacement algorithms: LRU and OPT

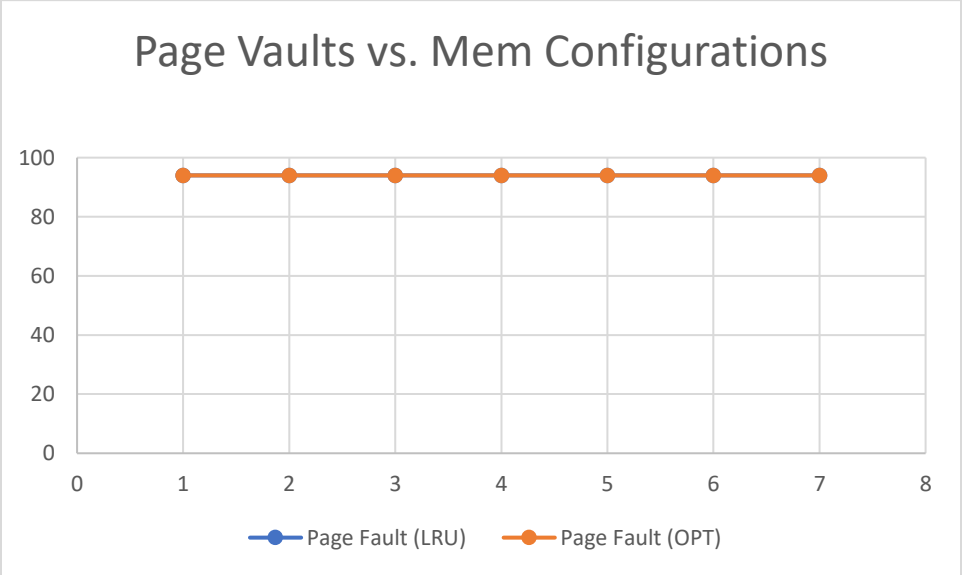

Figure 4. a total of 1024 frames and a page size of 4 MB

Memory split: 1:1, 1:3, 3:1, 3:5, 5:3, 7:9, 9:7

Page replacement algorithms: LRU and OPT

From the above graphs and running a numerous number of simulations I observed that LRU does work well but resulted in more page faults than OPT. However, with a high frame number LRU performs almost as well as OPT with any given memory split. Additionally, increasing the page size decreases the number of page faults by a significant amount. LRU suffers in performance due to the expensive lookup time. OPT utilizes a strategy of selecting the page to be evicted that is going to be used the farthest into the future. Although this is a downside of OPT since it is generally not known what future memory request will occur, in our project we knew the memory requests and were able to take advantage of using OPT which had on average (results from overall figures).