Errors in Implementing Virtual Memory and Proposed Solutions

Memory is a vital component in computers, as it stores tasks. However, memory becomes limited when a certain number of tasks are stored, making running more programs impossible. To optimize computers' memory, developers created a mechanism known as virtual memory (VM). This management technique fulfilled its purpose, although errors such as limited frames, a long page table, and thrashing may occur.

In a VM, programs are divided into equal parts referred to as pages, and the main memory is divided into segments that work as "frames" (Yadin, 2016, pp. 205-220). One frame occupies one page (Elahi, 2017, pp. 153-155). When the system loads multiple pages at once, the number of frames become limited. This causes many delays in transfers between VM and main memory (Blanchet et al., 2012, pp. 175-182). In such situations, the system allows new pages to occupy frames that are in use (often inactive frames). To prevent this error, Yadin (2016, pp. 205-220) proposed decreasing the number of transfers happening in parallel or increase the number of frames by increasing memory size.

A key aspect of VM is the page table, this is where page addresses are stored. Having smaller pages are often preferred, but then a long table is required (Blanchet et al., 2012, pp. 175-182). Smaller pages with larger tables could be counterproductive, due to the table consuming more memory. To alleviate this, some manufacturers considered increasing the page/frame size in order to decrease the table's length (Yadin, 2016, pp. 205-220). Whereas others implemented additional table levels, storing one level in the main memory and another in the VM (Blanchet et al., 2012, pp. 175-182).

Disk thrashing is a common flaw that is caused by an increase in paging activity, which is the process of loading a page into the main memory (Yadin, 2016, pp. 205-220). When a part has to be loaded, an interrupt known as Page Fault gets filed (Blanchet et al., 2012). Patel and Sehgal (2018) claim the root of the issue is that VM tends to be slower, which increases Disk input/output (I/O) time. This causes thrashing. They also proposed a logical method that uses compression to increase the VM size and decrease I/O time. During swapping, pages in the virtual memory get compressed and then decompressed in the main memory to be used. This increases the efficiency of the I/O process, ultimately preventing thrashing.

In conclusion, VM is a profound mechanism that optimizes computers’ storage. Although developers face some errors that may hinder its full potential, many solutions are being proposed that show how virtual memory can be utilized with minimal errors.

Bibliography

Blanchet, G., Dupouy, B., and Blanchet, G. (2012). *Computer Architecture*. *E-book Central*. [Online]. Available at: https://ebookcentral.proquest.com/lib/manchester/reader.action?docID=1117269 (Accessed: 11 November 2020).

Elahi, A. (2017). *Computer Systems: Digital Design, Fundamentals of Computer Architecture and Assembly Language*. Springer International Publishing AG. [Online] Available at: https://link.springer.com/chapter/10.1007/978-3-319-66775-1_7 (Accessed: 11 November 2020).

Patel, H. and Sehgal, N. (2018). Using Virtual Memory Compression to Reduce Disk I/O Time and Delay. *2018 IEEE Punecon,* Pune, India. 30th November – 2nd December 2018. IEEE. pp. 1-2. Available at: https://ieeexplore.ieee.org/document/8745396/authors#authors (Accessed: 27 November 2020).

Yadin, A. (2016). *Computer Systems Architecture. E-book Central.* [Online]. Available at: https://ebookcentral.proquest.com/lib/manchester/reader.action?docID=4683317 (Accessed: 29 October 2020).