**Assignment 3**

**Part 1: Understanding Memory Hierarchy**

In high-performing computer architecture, the designing of the memory hierarchy plays an important role in modern-day computer systems. The major purpose of this approach is to access the data with high speed efficiently but the cost needs to be within range. This approach helps manage the data and its fast flow within the system, and it uses the principles of data locality to bring the information near the processor for efficient and repeated access.

**Memory Technologies**

In memory hierarchy designing concepts memory technologies like Static Random-Access Memory (SRAM) and dynamic Random-Access Memory (DRAM) have an impact on the performance of a system. SRAM is known for its speed which can retain the information as long as the power of the system exists and thus it is used in CPU cache. Ideally, it is found that SRAM is used for the L1, L2, and sometimes in L3 caches within the processors. As the cost is high for SRAM it is not used in the main memory of the system. It helps to keep the frequently used information close to the CPU so that the accessing time is less and this helps in minimizing the latency and increases the overall performance of the system. On the other hand, the data accessing speed of DRAM is slower than SRAM and there is a requirement of refreshing it in certain time intervals which is not required for SRAM at all. The cost is lesser than SRAM thus it is used in main memory i.e. Random Access Memory (RAM). DRAM holds the information as well as the instructions that may be required by the CPU soon and it is working like an immediate storage before the day is being written to the secondary storage. Apart from that, discussing the memory technologies, Emerging Memory Technologies like the usage of Non-Volatile Memory (NVM) provide an extra advantage in memory hierarchy as an added new layer into it. It is placed within the DRAM and secondary storage which is used as a high-speed storage option and reduces the storage access time in modern-day computer systems (Wu, 2021).

**Advanced Cache Optimization**

Another important factor of memory hierarchy design is Advanced Cache Optimization. This technique consists of approaches like prefetching, using of victim cache, cache partitioning etc. All of these techniques are very useful while designing the hierarchy of the memory. Prefetching is a technique where the required information is loaded into the cache memory before the CPU requests it. Prefetching is of two types i.e. hardware-based and software-based. In hardware-based prefetching, logically the information is predicted and loaded but in the case of software-based prefetching, developers optimized the code for accessing the information and its patterns. All of them are together helpful to reduce cache missed which actually improves throughput. The other used approach in Advanced Cache Optimization is the usage of victim caches, where a small amount of cache memory is used to store the information that was recently removed from the main cache. It is helpful to reduce the conflicts cache misses, and reusability of the information and decreases the rates of cache miss penalties which improves the overall system performance with improvement in cache hit rates. Another approach is cache partitioning which divides the cache into multiple parts so the interference is reduced and the improvement of data locality of the processes can be observed. In high-performing computerized systems, this helps in cache utilization through multiple cores and ensures the cache resources are adequate for each system core (Zhao, 2020).

**Virtual Memory and Virtual Machines**

Virtual memory fundamentally uses page tables and address translations which allows programs to use more memory that is not available physically and makes the processes isolated. It also used the various Page Replacement Algorithms like Least Recently Used (LRU), First-In, First-Out (FIFO) etc to balance the memory usage and performance of the overall systems. The virtual memories are well capable of controlling the concurrency and allow the system to multitask. It uses a memory isolation technique to isolate memory spaces for different processes and prevent the application from accessing other's information if it is not explicitly shared. In this way, virtual memory improves performance and efficiency and also improves the response time and allows systems to execute more numbers of concurrent processes.

On the other hand, Virtual Machines (VMs) use virtual memory to operate multiple operating systems within one physical machine and this is created by adding an extra layer of abstraction in its own virtual memory system (Kloda, 2019).

**Cross-Cutting Issues**

While focusing on the improvement of the performances of the system through various approaches incorporated with Memory Hierarchy Design, the cross-cutting issues also play a big role where the cost and performance need to be balanced at the very first stage. The consumption of power also needs to be efficiently handled while designing. In modern days, the utilization of memory patterns is quite complex so the designing needs to be proper and diverse so the different workloads (AI, database transactions etc.) can work efficiently. Nowadays, to resolve these cross-cutting issues, various emerging trends and technologies such as 3D Stacked Memory, In-Memory Computing etc. implemented to improve the performance of the overall system by preventing the issues.

Moreover, it is very crucial work to design memory hierarchy in high-performance computerized systems nowadays as it is a complex task that consists of the incorporation of different memory types, optimization strategies and the latest technologies and also significantly manages the power usage and adaptability.

**References:**

Kloda, T., Solieri, M., Mancuso, R., Capodieci, N., Valente, P., & Bertogna, M. (2019, April). Deterministic memory hierarchy and virtualization for modern multi-core embedded systems. In *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)* (pp. 1-14). IEEE.

Wu, K., Guo, Z., Hu, G., Tu, K., Alagappan, R., Sen, R., ... & Arpaci-Dusseau, R. H. (2021). The storage hierarchy is not a hierarchy: Optimizing caching on modern storage devices with orthus. In *19th USENIX Conference on File and Storage Technologies (FAST 21)* (pp. 307-323).

Zhao, J., & Di, P. (2020, October). Optimizing the memory hierarchy by compositing automatic transformations on computations and data. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (pp. 427-441). IEEE.