# Describe how concepts such as risc, pipelining, cache memory, and virtual memory ...

Family

A Short Research Paper The computer and information science products are analyzed and evaluated for their performance and efficiency. Judgment of optimality in utilization of the resources like time and storage space play key role in conduction of this analysis. The consideration of time and space for performance analysis is equally important in evaluating both the computer hardware and software. The software being an algorithmic structure can be analyzed through the algorithm analysis techniques. The time complexity is measured as a function of input size n and the growth is taken as a measure to comment on the algorithmic efficiency through asymptotic. In different scenarios the notations for representation of time are different like turnaround time, waiting time and response time. Time efficient algorithms are not space efficient usually due to time/space trade off. These algorithms may prove to be heavier for the hardware in terms of their need for storage space they require to execute. In case of computer hardware the representation of time and space for performance evaluation can be seen in terms of processor speed, bus speed, memory access latency, throughput and the size of storage space accordingly. It has been observed historically that advancement in computer hardware and software always go hand in hand in order to cope up with the requirements of each other. The advent of semi conductor memory introduced the capability of memory addressing, memory segmentation and memory fencing through fence registers. This created the capacity of multiprogramming with memory security against old fashioned batch processing systems. Multiprogramming later evolved into timesharing and facilitated processor sharing. The huge sizes of peripheral devices and offline operations were replaced by the invention of miniaturized

controllers with local buffers. This revolutionized the co-existence of peripheral devices through techniques of spooling and DMA. The memory management schemes evolved from a single partition allocation mechanism to multi partition mechanisms like MVT and MFT. The problems of internal and external fragmentation and compaction resulted in the genesis of paging as a non-contiguous memory allocation mechanism. The overgrowth of program sizes was initially dealt with techniques of overlays. This technique went obsolete due to programming restrictions and enormous sized program code. The concept of virtual memory appeared as a solution to this problem. The fact that has been continuously observed during the evolution of computing was, that with the implementation of new and advance techniques to address old problems the room of improvement in the performance of the computing systems remained persistent. The new solutions in turn produced new problems and this cycle goes on. For example to address the problem of incompatibility between the sizes of physical memory and program code, the concept of virtual memory and the techniques of paging and demand paging were introduced. These in turn resulted in multiple problems like, Access latency due to the involvement of secondary storage. Memory allocation for oversized page tables for which different strategies like multi level paging, inverted page tables and hashed page tables were applied (These have their drawbacks too). The involvement of cache as TLB to reduce access latency. Frame allocation and page replacement strategies and their drawbacks. Thrashing etc. The improvement capacity always coexisted with the upcoming solutions and increasing requirements. The last few decades saw the advent of some

remarkable achievements like the RISC architecture, pipelining, cache, super scalar processing, memory management systems and virtual memory. The computational power increased manifolds as a result of these achievements. Referring to “ Reduced instruction set computing” The type of computer architecture that involved simple instructions that were comprehensive and lesser in number. RISC incorporated registers along with the main memory itself for the allocation and storage of instructions. Superscalar computing involves the parallel execution of instruction by a single CPU. The main transition between CISC and RISC type of computing resulted after the advent of Superscalar computing. Pipelining, on the other hand, refers to a series of data elements. Connected in a series, the output of one process is the input of the next. Pipelined instructions can also be executed in parallel. Overlapping execution of multiple instructions having the same circuitry is greatly assisted by the phenomenon of pipelining. Superscalar computing involves the incorporation of pipelining within it. Thus it can easily be figured out that the combination of pipelining and superscalar processing within a system would definitely result in faster processing speed(1). Main memory is an essential part of every computer system. The main memory holds part of any computer process that is in execution. It is also referred to as RAM. Data can be read from or written to the computer memory. It is essential for the computer memory to have substantial space so as to facilitate the execution of processes. There, however is a limit to the size of RAM that can be incorporated into a computer system. For a number of processes to be executed simultaneously in a system the size of RAM often does not suffice. It is then that the phenomenon of Virtual memory comes in handy. Virtual

memory looks up any pages in the memory that are static and are not being used for long. It swaps these pages onto the hard disk thus freeing space for the allocation of new tasks. This process happens automatically. Virtual memory increases the size of the RAM without introduction of any new physical memory components into the computer system. A system that incorporates virtual memory has its memory divided into physical and logical pages. Cache is another type of memory. Its hardware composition is different from that of the main memory. This enables the access latency of the cache to be much lesser than that of the main memory. The most frequently accessed instructions are placed in cache so that they can be accessed very quickly. There are two main strategies that are used to implement access to and from the cache. Most flexible cache strategy is the " Fully associative cache". The other strategy is the " direct mapped or one way subsystem associative cache". A fully associative cache subsystem involves the mapping of bytes of data onto any one of the lines of the cache memory. In direct mapped cache however fixed blocks of memory are always associated with cache lines whether they may be in use or not. Both the strategies have their specific pros and cons and neither can be weighed over the other (2). The close study reveals that the improvement in the software implementation techniques to utilize the hardware resources optimally is highly necessary. Some of the unsaturated areas for example are software pipelining (3) and multiprogramming scheduling (4). Sources: Silberschatz A., Galvin P. & Gagne G. (2010) Operating System Concepts 8 Ed. John Wiley and Sons. References: (1) Walters Garrison E. (2000). ESSENTIAL GUIDE TO COMPUTING: THE STORY OF INFORMATION

TECHNOLOGY. Prentice Hall. (2) Watts D, Chabrol A, Dundas P, Fredrickson D, Kalmantas M, Marroquin M, Puri R, Rodriguez J, Zheng D.(2000). TUNING IBM SYSTEM X SERVERS FOR PERFORMANCE. IBM Redbooks. (3) Lam, M. (1988). Software Pipelining: An Effective Scheduling Technique for VLIW Machines. UMUC Library. (4) Liu, C. L. & Layland J. W. (1973) Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. UMUC Library.