ABSTRACT

The memory hierarchy is usually the largest identifiable part of a computer system and making effective use of it is critical to the operation and use of the system. We consider the levels of a memory hierarchy and describe the state of the art and likely directions for both research and development. Algorithmic and logical features of the hierarchy are directly associated with specific components and the memory hierarchy both overall and with respect to research and development. We consider the levels of effective use of the memory hierarchy both overall and with respect to research and development. We consider the levels of effective use of the memory hierarchy both overall and with respect to research and development. We consider the levels of effective use of the memory hierarchy both overall and with respect to research and development.

In figure 2, we show what we believe represents the type of large computer system memory hierarchy that will become common in the early 1980's. To figure 1, one will note that we have added in figure 2 two levels: gap filler technology and mass storage. Currently, there are orders of magnitude difference (the access gap) both in cost and performance between random access (mass storage) and main memory. Much time and effort is spent in computer systems in finding efficient ways to move information across the access gap. A computer system using a level of storage whose access gap is propagated significantly in both improved performance and decreased system complexity.

Three technologies are just under development, and/or production fall into the middle of the access gap. Charge coupled devices (CCD)'s (2,4), charge packet semiconductor shift registers (5), and semiconductor shift registers function by shifting magnetic domains. Magnetic bubbles may be no cheaper than CCD's (depending on the mechanism finally chosen, e.g., bubble lattice files [5], will be very cheap), but they are considerably slower. The best possibility is electron beam accessed memory (EBAM) [19]. Small charge patterns are stored on the face of a CRT, using a semiconductor large- and hyperfine beam focusing. Only one EBAM development effort exists at the moment [43] and there is some doubt about its eventual success, since the engineering problems appear to be formidable. All three of these technologies have the potential for inclusion in memory hierarchies over the next few years.

The difference between tape and mass storage is an integrated part of the memory hierarchy. Both gap filler technology and mass storage devices are only just being developed and even in some cases, there are no easy way to easily allow the user to tune the system to improve its performance. By far the most fertile direction for new results (research or development) is in the study and design of memory hierarchies of the future, rather than in the optimization of current systems.

DIRECTIONS FOR MEMORY HIERARCHIES AND THEIR COMPONENTS:

SEARCH AND DEVELOPMENT

University of California, Berkeley

Alan Jay Smith

CMOS 75-0576 and the Department of Energy under contracts W-7405ENG-48 (to LBL) and EY-76-C-03-0515 (to SLAC).

University of California, Berkeley, California 94720. The author is also on the staff of the Lawrence Berkeley Laboratory and a visitor at the Stanford Linear Accelerator Center.

I. INTRODUCTION

Contemporary large computer systems frequently employ a memory hierarchy as shown in figure 1, where we show cache memory, main memory, drums, disks and tape. The efficient use of this memory system is crucial to the operation of the whole computer system. In this paper, we shall examine the memory hierarchy both overall and with respect to its components in an attempt to identify research problems and project future directions for development.

The effects of the design of the memory hierarchy can be considered to fall into two (overlapping) areas: performance and logical view. Performance denotes those aspects of the hierarchy design which affect the measures of performance of the computer system such as throughput, speed, response time, turn around time and cost effectiveness. Logical view refers to the logical view given the user of the memory system: how is the memory addressed, named, where is the information? (virtual vs. real location), how is the information protected, etc. These two views are not independent but are interrelated, since performance is impacted by the logical view, and the cleanliness or uniformity of the logical view can be increased by the design of the system.

II. THE MEMORY HIERARCHY

In figure 2, we show what we believe represents the type of large computer system memory hierarchy that will become common in the early 1980's. To figure 1, one will note that we have added in figure 2 two levels: gap filler technology and mass storage. Currently, there are orders of magnitude difference (the access gap) both in cost and performance between random access (mass storage) and main memory. Much time and effort is expended in computer systems in finding efficient ways to move information across the access gap. A computer system using a level of storage whose access gap is propagated significantly in both improved performance and decreased system complexity.

Three technologies are just under development, and/or production fall into the middle of the access gap. Charge coupled devices (CCD)'s (2,4), charge packet semiconductor shift registers (5), and semiconductor shift registers function by shifting magnetic domains. Magnetic bubbles may be no cheaper than CCD's (depending on the mechanism finally chosen, e.g., bubble lattice files [5], will be very cheap), but they are considerably slower. The best possibility is electron beam accessed memory (EBAM) [19]. Small charge patterns are stored on the face of a CRT, using a semiconductor large- and hyperfine beam focusing. Only one EBAM development effort exists at the moment [43] and there is some doubt about its eventual success, since the engineering problems appear to be formidable. All three of these technologies have the potential for inclusion in memory hierarchies over the next few years.

The difference between tape and mass storage is an integrated part of the memory hierarchy. Both gap filler technology and mass storage devices are only just being developed and even in some cases, there are no easy way to easily allow the user to tune the system to improve its performance. By far the most fertile direction for new results (research or development) is in the study and design of memory hierarchies of the future, rather than in the optimization of current systems.

CPU —> CACHE —> MAIN MEMORY —> DRUM DISK —> TAPE

3 Mips 32K bytes 4 megabytes 5-10 1000 to 50000

Figure 1

40 KB 128-512 KB 32-128

Figure 2

MAGNETIC TAPE

MAIN MEMORY

MAGNETIC DISK

MAGNETIC TAPE

50000 reels 10 KB-10 GB 20-20 10^13 10^6 reels

In this paper, we shall examine the memory hierarchy both overall and with respect to its components in an attempt to identify research problems and project future directions for development.
indicated in figure 2, as well as to those problems that don’t seem to be associated with specific levels in the hierarchy.

11. CACHE MEMORIES

Cache memories, also known as High Speed Buffer memories, are buffer memories managed by the hardware and placed between the CPU and the main memory. Because of the principle of locality, each address extremely often now that after a certain amount of time, the information contained in the cache memory is always searched (in some sense architecturally transparent. It seems clear that as circuit technology is even faster, thus, larger high speed computer systems are memory speed limited. Because of this requirement for high speed, the implementation details of the cache are less.

The performance issues in cache memories concern two goals: maximizing the probability of finding needed information and minimizing the time to access it if it is there. Most of the published research concerns the former. This section attempts to summarize recent work and to evaluate the extent of the search. If the address can map to a large number of locations in the cache, there is a higher probability of finding it, but looking takes longer. This is a well understood problem (see [22] for some data) and set sizes of 2 to 8 are commonly chosen. Selecting a size equal to the information transfer unit (line size) is also a well understood problem; line sizes of the order of 32 bytes have been standard. Prefetching information before it is needed is useful [36] is quite useful for cache memories, although it is not generally implemented in computer architecture.

The access time issue, mentioned above, leads to two possible changes in cache architecture, neither of which has been fully evaluated in the published literature. The cache is generally used for both instructions and data. Instructions are accessed by the fetch and decode (I) unit of the CPU, whereas the data is used by the execution (E) unit. The I and E units are relatively separate, can both access simultaneously active and normally accessed information from each other in the CPU. If each of the I and E units had its own cache, access time could be decreased. Instructions and memory locations accessed by the new process will not be cache resident. This is the problem considered by Eastman and Bennett [12] who consider two goals: maximizing the probability of finding needed information in the cache, but that is not necessarily the case in systems with channels or multiple processors. An important problem, yet to be definitively addressed, is the design of a scheme for maintaining memory consistency in a multiprocessor system where each processor has its own cache. Multiple copies of the same data may exist. There is no generally accepted solution to this problem and further work needs to be done.

A major problem with cache memories is address space swapping during task switching. When the processor switches to another process, all the information used by the old process is dumped (cold start) miss ratios. Very little work has been done on this problem, and much work is needed to find an acceptable solution. One possibility is in real systems or to study how to minimize it. One possibility is two caches – one for user state and one for supervisor state.

In cache memories, therefore, most of the hit
ratio problems - set size, line size, prefetching and cache size, are generally well understood. The remaining problems will be: (1) the design and organization area: split (data/instruction, user/supervisor) cache, virtual vs. real address access and multiple cache consistency.

III. MAIN MEMORY

In early computer systems, and in fact until quite recently, main memory was a scarce and expensive resource. Thus, a very large amount of the system's effort was directed at trying to improve the usage of main memory, with respect to segmentation, paging, [46], compaction, restructuring, etc. All these efforts clearly made memory become relatively inexpensive (although not free(!)). The implication of this is that while intellectual problems in the context of computer system operation.

...阿尔科技有限公司...
V. DISKS AND DRUMS

Even though main memory are a mature technology, and there are few new problems or developments associated with them. A good survey of the disk and drum interface is [2]. File systems and the disk interface have been increasing in density fairly steadily; currently the density seems to be doubling about every five years (this time not being referenced to). What is being done in the industry is producing any but the sort of minor performance (access time) improvements.

One could expect to see perhaps 10% to 30% decreases in access time over the next 3 to 10 years, which is a small enough factor that it will not have a large impact on system performance or operation.

Drums are no longer cost effective with the decreases in cost for main memory and disk and the appearance of gap filler technology. Some of the research activity in disks/drums has been in analytic modeling, and [36] provides a (1976) bibliography of the disk literature. There are no new problems or systems, but which merit further development, publicity, and/or research. The disks used for the Cray I at Los Alamos [35] have little track-to-track transfers can be buffered ahead or behind. In this same system the I/O system has been optimized with "caching" in two aspects: (a) file placement in consecutive sectors is encouraged while minimizing fragmentation and (b) file transfers are kept separate so that reading ahead and writing behind. Further research is needed in the design of such strategy routine algorithms.

The position of the detection photocell is to make disks "intelligent," since the cost of logic is declining rapidly with respect to the cost of the mechanical components of the disk. This is being done with drums, as noted earlier, in the MAP (Relational Associative Processor) system at the University of Toronto [30]. In that case, the storage device is gap programmed to search for the desired record. Similarly, the disk could be allowed to do its own error correction [14], a task which is often slower to occur on or more of the controller, channel, or CPU.

Overall, therefore, we see only three real research areas to do with disks: intelligent disks, disks associated with gap filler technology, and disk (or I/O system) strategy routines. The real issues are of the straightforward development or core extensive implementation and publicity. The directions for development are much the same as those noted earlier. Further research: more logic (intelligent or not) will appear in the disk spindle or controller and this logic will serve to operate, correct errors and buffer the disk.

VI. MASS STORAGE

As noted earlier, mass storage devices (storage on the order of a trillion bits or more) have typically been achieved associated with this new capability are of course new and interesting research problems. Associated with this new capability are of course new and interesting research problems. They include: (a) the problem of moving information from mass store to disk and later from disk back to mass store, the file migration problem; (b) the problem of dealing with files that have to be stored together in the same physical location and attributes of the data. There is nothing new in this idea, yet despite its obvious (to the author) advantages, it has been implemented to only a very limited extent in most systems. An important and pressing development problem is the interworking of the "file system" on the mass store devices.

VII. LOGICAL and USER VIEW PROBLEMS

Despite the comments in the above paragraph, there is some question as to how extensive the address space should actually be. It is reasonable, for example, to make mass storage byte addressable, with the consequent cost of 40 or more bits per address. There is clearly the need for a means of dealing with mountable volumes (tapes, disk). It is probably desirable to allow for dynamic mapping from one set of logical names (directory and file names, and the author) advantages, it has been implemented to only a very limited extent in most systems. An important and pressing development problem is the interworking of the "file system" on the mass store devices.

Another aspect of mass storage problems that has not been fully developed is the interface between operating systems and their file systems. Both command languages and file systems are part of operating systems;
there is considerable need to integrate the two. The management of the physical storage of the data base system should perhaps be subsumed into the overall management of the unified memory hierarchy on which the data base system runs.

VIII. CONCLUSIONS

In this paper we have explored the structure of the memory storage hierarchy with an eye to noting aspects of such hierarchies which are not fully understood, developed or optimized. We found there were many potential roles still to be played on cache organization, but not on the traditional miss ratio problems of cache design. With regard to gap filler technology, almost every aspect of its use deserves both research and implementation problems—should it be used, where?, when?, how? Similarly for mass storage—how should it be operated, now? when should files be migrated? What should be done about reliability and recovery? Those parts of the memory hierarchy of long standing, such as main memory disks and drums pose few new problems. The principal research problem for these levels is now that should these parts of the system be made intelligent with the decreasing cost of logic, intelligent memory or peripherals becomes a viable alternative. We also noted that many questions about the logical structure of the user name space, and how it should map into the memory hierarchy. All of these problems are both interesting and important, and we hope that this paper will stimulate some research in these directions.

BIBLIOGRAPHY

no., 7/9-753.
