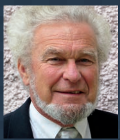
50 & 25 YEARS AGO



EDITOR ERICH NEUHOLD
University of Vienna
erich.neuhold@univie.ac.at

MARCH 1969

[Editor's note: In this issue, two time-sharing systems of 50 years ago are discussed. They are mostly interesting for the explanation they give for the technology that existed then and for reminding us of the tremendous progress we have made since.]

Evolving Digital Computer Systems Architecture (p. 2) "Computing Architectures are distinguished by the explosive growth of their great variety in the last decade. This article will describe that part of computer history of the last two decades, that has led to the future as seen today, plus point out various architecture trends for the future. In the dark abysm of time – to be more precise, the period surrounding the period mid to late 1950's – computer system design became a recognized profession. About the same time, or somewhat before, designers discovered that there were other computer architectures, alternate to the classic van Neumann type. Since then there has been a tremendous proliferation of different types of computer organizations designed and many even built. This trend will continue for the foreseeable future throughout the final sweep of this century." (p. 5) "By trying to be somewhat compatible with old or current systems, systems architectures which are totally different and innovatively new are difficult to sell to management and customers. Still, in the face of these squelching problems, the field, lately, is beginning to see a formidable number of new architectures!" (p. 8) "The most noticeable trend for next generation computer architectures is the building into the hardware many more features – many of which heretofore were only programed. The basic reason for this trend is twofold, namely: programming is costly and logic is low cost." *[Editor's note: The article implies the appearance of many different/novel architectures. However, further reading reveals that what is really meant are things such as separate I/O units, graphic subsystems, intelligent terminals, multiprocessors, parallel architectures, and so on. Despite that somewhat disappointing revelation, the article gives some insights into the thinking about computers 50 years ago. The idea of moving more and more software into hardware—look at IBM—was seen as a great idea then and was tried but has since been mostly discarded in mainline computing.]*

Concepts for Buffer Storage (p. 9) "The paper discusses concepts for buffering information from backing stores in the computer main frames as exemplified in the IBM System/360 Model 85. The four basic types of buffer design examined are: sector, direct-mapping, fully associative, and set associative buffers. The set associative approach appears to yield the best performing buffer." (p. 11) "BUFFER DESIGN VARIATIONS: Four basic types of buffer design have been seriously considered. 1. Sector or Page as in the Model 85; 2. Direct Mapping Blocks; 3. Fully Associative Blocks; 4. Set Associative Blocks." (p. 13) "Given the basic approach of any one of the four described, other decisions must also be made in the buffer design. The two key decisions are those of store through versus swapping and the question of replacement algorithms." *[Editor's note: The article points to set associative blocks as the best approach, and it is widely used today. Many of the other considerations are also still valid despite the tremendous growth in computing power and storage size over the last 50 years.]*

Impact of Terminals on Computer Systems (p. 21) "The Workshop was designed to bring together specialists in the fields of applications, human factors, software design, and system architecture to provide a forum for intellectual interchange and discussion. It was intended to put all aspects of the problem into proper perspective and to approach the system design from an overall point of view. The Workshop started with a review of a number of typical terminal-oriented systems in operation or under development. Then the functional and operational requirements as they relate to applications and human factors was identified and assessed. The impact of terminals on software was assessed. Finally, the requirements relative to applications, human factors, and software implementation were summarized and their

Digital Object Identifier 10.1109/MC.2019.2898718
Date of publication: 2 April 2019

impact on future computer system architecture and design was examined." *[Editor's note: It is interesting to think about the approaches described that have passed the test of time. It may be even more interesting to read the opinion of some of the well-known people who contributed, e.g., Alan Kay, Doug Engelbart, Gio Wiederhold, and others.]*

Are We Responsible? Workshop on the Social Responsibilities of Computer Professionals (p. 35) "Our 'profession,' which could have an impact second to no other technological achievement, has done little to participate in a socially significant manner in the consequences of its works. In our own best interest, as well as the best interests of society, we have much to do. We had best assume leadership, or we will exercise no control over the social impact of information processing technology." (p. 36) "It was concluded that we have a great deal to learn about such matters from the older *[Editor's note: engineering]* professions, and it was observed that even the legal and medical professions are now going through a state of flux concerning codes of ethics and their true value and use. We concluded that we would do well to learn from their experience while considering the merit, structure, and enforceability of such a code for computer specialists. Privacy: It is felt that a real case can be made that information processing technology can improve privacy, and we should take this viewpoint and proceed to make it so, rather than remain on the defensive. We discussed the possibility that privacy is being invaded slowly with the aid of our technology, via decisions being made without the individual's knowledge. The technology may indeed introduce an entity sufficiently new and significant that it calls for appropriate modification of our constitutional rights. … There are many influential people in the computer industry who have a genuine concern about human values. Though there are said to be forces within computer companies that militate against the socially conscious engineer speaking out or being active on this front, it is observed that senior people in the industry are well-enough established to be able to do this without harm. If senior people speak out, it will aid the younger people to have the courage to act on their convictions – to stiffen their backbone." *[Editor's note: This workshop raised many issues that still trouble us today. In a way, forces against answering those ethical questions and addressing concerns about human rights are still winning, and much has to be done far beyond the ethics guidelines that the Association for Computing Machinery and the IEEE Computer Society are developing and updating.]*

MARCH 1994

www.computer.org/csdl/mags/co/1994/03/index.html

Guest Editor's Introduction: The I/O Subsystem A Candidate for Improvement (p. 15) "The hardware consists of processor(s), memory, and 'everything else.' The 'everything else' we generally combine under the umbrella 'I/O,' whose job it is to manage the availability of information to and from the processor(s) and memory. That information comes from storage devices, networks, and nonstorage devices. The I/O subsystem is the collection of all three." (p. 16) "So, while computer architects will surely continue building faster processors and more sophisticated memory systems, many are now focusing their attention on the performance problems of I/O. That attention requires rethinking many of the old paradigms: What do we do at the host, and what do we do at the device? What do we do in hardware, and what do we do in software? What do we relegate to the control of the operating system, and what do we put in the hands of the application programmer? What do we control at the level of the autonomous computing system, and what do we hand off to the network?" *[Editor's note: The articles in this issue cover the areas mentioned here, present a cogent analysis of the state of the art in 1994, and offer some glimpses of the future. However, none of the articles envisioned the tremendous increase in memory, distributed environments, and big data and the requirements resulting from those.]*

An Introduction to Disk Drive Modeling (p. 17) "This article demonstrates and describes a calibrated, high-quality disk drive model in which the overall error factor is 14 times smaller than that of a simple first-order model. We describe the various disk drive performance components separately, then show how their inclusion improves the simulation model. This enables an informed tradeoff between effort and accuracy. In addition, we provide detailed characteristics for two disk drives, as well as a brief description of a simulation environment that uses the disk drive model." (p. 26) "Our full model includes the following details; … The host I/O device driver … The SCSI bus … Disk controller effects … Disk buffer cache … Data layout model … Head movement effects." (p. 27) "We plan to use our refined disk drive simulation model to explore a variety of different I/O designs and policy choices at host and disk drive levels."

Disk Arrays: High-Performance, High-Reliability Storage Subsystems (p. 30) "Disk arrays are an essential tool for satisfying storage performance and reliability requirements. Proper selection of a data organization can tailor an array to a particular environment. … Figure 1 shows the three main methods for performing this translation. The conventional approach is to address each disk independently and map logical block numbers to disk block numbers directly. … Disk striping, also called disk interleaving, folds multiple disk address spaces into a single, unified space seen by the host. This is accomplished by distributing consecutive logical data units (called stripe units) among the disks in a round-robin fashion, much like interleaving in a multibank memory system." *[Editor's note: The article discusses in detail tradeoffs related to performance, reliability, and failure recovery.]*

Caching Strategies to Improve Disk System Performance (p. 38) "Caching can help to alleviate I/O subsystem bottlenecks caused by mechanical latencies. This article describes a caching strategy that offers the performance of caches twice its size. … Finally, we investigate the performance of three cache replacement algorithms: random replacement (RR), least recently used (LRU), and a frequency-based variation of LRU known as segmented LRU (SLRU)." (p. 46) "Although LRU is a well-known policy with low implementation overhead, we find that for small cache sizes SLRU halves the cache size for a given miss rate. SLRU is recommended for cache implementations in the range of 2 to 8 Mbytes, because SLRU performs best on workloads that overwhelm LRU caches. However, caching is dependent on the type of system workload and should be implemented with caution. Write-intensive applications and applications that do not reuse data are not likely to benefit from caching." *[Editor's note: Here I just point to the cache sizes considered in 1994!]*

A Systematic Approach to Host Interface Design for High-Speed Networks (p. 47) "Optical fiber has made it possible to build networks with link speeds of over a gigabit per second; however, these networks are pushing end-systems to their limits. For high-speed networks (100 Mbits per second and up), network throughput is typically limited by software overhead on the sending and receiving hosts. Minimizing this overhead improves application-level latency and throughput and reduces the number of cycles that applications lose to communication overhead." *[Editor's note: This article presents a nice analysis of the state of the art in 1994 for a number of network architectures.]*

Container Shipping; Operating System Support for I/O-Intensive Applications (p. 84) "A rapidly growing class of I/O-intensive applications is multimedia computing, in particular, applications that acquire or present video (or image) and audio streams, possibly transforming them in novel ways under programmer control. These applications are often distributed and interactive, imposing real time constraints for the delivery of large volumes of data transported over potentially long distances. Examples include video teleconferencing with shared work spaces, remote scientific visualization and sonification, and distributed virtual reality." (p.90) "Our design principles led us to use the move model and virtual transfers, a semistructured data organization, and separate transfer and access mechanisms. Much of the inspiration for our implementation of these design decisions came from cargo transportation. The problems encountered in moving cargo efficiently are similar to those we are trying to solve, and many were solved by 'containerization,' an important advance of the 1960s." (p. 91) "Shipping is based on the move transfer model, which is easy to understand and implement because data are never shared. Data always belongs to one, and only one, domain. We decoupled shipping of containers from making their contents accessible, which can be done selectively." *[Editor's note: This article presents an interesting approach involving the packing of data structures into containers and avoiding linearization of such structures before transfer.]*

Product Reviews: Building Expert Systems (p. 97) "Back in the 1980s when everyone seemed to be building some sort of expert system, you wouldn't have expected to build a rule-based system without writing lots of if-then statements. More recently, a number of products let you build (essentially) rule-based systems without ever writing a single rule." (p. 99) "As I mentioned at the beginning of this review, Adept and K-Vision have some superficial similarities. They both simplify the process of building rule-based systems by freeing you from the one-rule-at-a-time mentality and helping you structure the knowledge involved in solving the problem at hand." *[Editor's note: Since 1994, considerable further progress has been made in developing tools to ease the task of building rule-based systems, including approaches for the automatic mining for such rules.]*

Open Channel: US Needs a WAIS in the White House (p. 128) "An automated background investigation tool (ABIT) could derive information about a candidate's tax liability concerning babysitters and nannies. It would empower the search team to learn about an individual's intellectual past and discern his or her opinions on topical issues. … Specifically, the search team would have a wide-area information server (WAIS) in the White House as the ABIT core." *[Editor's note: This article is based on some embarrassing facts that appeared about presidential nominees. Unfortunately, 25 years later, most of these investigative tools have become available and are used legally and illegally by many public and private players, data and privacy protection laws notwithstanding.]*