ENGINEERING WORKSTATIONS

An engineering workstation is a complete computer system used for engineering applications. A typical engineering computation requires a moderate amount of computing power (especially in the form of floating point operations), relatively high-quality graphics capabilites, and a large main memory.

Workstations may be classified at an intermediate level between powerful minicomputers and affordable personal computers in terms of cost and performance. (However, personal computers are catching up with workstations in terms of processor power and memory size.) In addition, workstations are characterized by the following properties:

- *High Performance.* A modern workstation is equipped with a single or multiple RISC-based or CISC-based processors of 100 MHz or more, and a main memory of 64 megabytes or larger. High-speed and high-resolution graphics may be supported by a special processor separately.
- *Single User.* Usually, one workstation is dedicated to one user and not shared (in most cases) with other users. But some powerful workstations are used as departmental servers as well.
- *Multitasking.* Complex applications require the system to support multitasking. The user may use the workstation to do several tasks concurrently, and make full use of the power of the workstation.
- *Networking.* Workstations are networked for easy access to remote computing resources. Workstations can be connected to local area networks or even wide area networks such as the Internet. The commonly used communication protocol is transmission control protocol/Internet protocol (TCP/IP) (1).

In terms of computing power and physical size, workstations can be classified into three types: notebook, desk-top, and server. Notebook workstations are rare because of their low performance/cost rates in comparison with notebook personal computers (PC). From a hardware configuration standpoint, a workstation may be constructed using a single processor, or using multiple processors ranging from two to up to a few dozen. A bus-based symmetric architecture is adopted for a small number of processors, but when the number of processors increases, more complex structures such as hierarchical buses may be used to enhance the data access bandwidth.

The most common operating system for workstations is UNIX. It is a time-shared multitasking system supporting standard languages such as C. A graphical user interface (GUI) is provided to facilitate the use of these workstations. Usually such an interface is based on the X Window system (2), and is a WYSIWYG (what you see is what you get) interface. Two popular GUIs on workstations are Open Look and Motif.

Workstations are used by scientists and engineers to do computationally intensive or graphics intensive applications such as research, software development, scientific computation, and CAD/CAM. They may also be used for desk-top publishing, scientific and medical visualization, image analysis, and other applications.

The idea of a workstation originated at Xerox during the 1960s, but it was not until the development of microprocessor technology that workstations became widely available. Modern workstations began to appear in the market in the 1980s. Apollo Computer, Inc. released its first Apollo/DOMAIN workstations (DN100s) in 1981 with the operating system Aegis SR1 installed on them. Andreas Bechtolsheim designed a computer called Sun for the Stanford University Network in 1981. Afterwards he formed a company named Sun Microsystems, Inc. in 1982, and shipped Sun-1 workstations in the same year. Since then, workstations have become affordable computers for engineers and scientists.

Currently, the leading workstation manufacturers are Sun Microsystems, Hewlett-Packard Company, Silicon Graphics Incorporated (SGI), and Digital Electronics Corporation. (Apollo Computer was purchased by Hewlett-Packard in 1989.) They provide a large variety of workstations in the market ranging from single processor desk-top workstations to multiprocessor (up to 64 processors) departmental or even enterprise servers.

ARCHITECTURE

Similar to most other types of computers, a workstation mainly consists of one or more processors, main memory, mass storage devices (disks), and other I/O devices such as mice and keyboards. Workstations are also equipped with network adaptors and such multimedia devices as CD drives, microphones, and speakers. Initially, most workstations used 32-bit traditional CISC (complex instruction-set computer) (3) processors such as the Motorola 680x0s. However, at present, 64-bit RISC (reduced instruction-set computer) (3) based processors running at more than 100 MHz are widely used in modern workstations. For example, the RISC processors UltraSPARC 1, MIPS R10000, and PowerPC are used in Sun UltraSPARC stations, SGI O_2 machines, and IBM RS/6000 workstations, respectively.

Two levels of cache (Level 1 and 2) usually help bridge the gap between processors and memory. Level 1 cache, which is on the processor chip, and is often called on-chip cache, is about 32 k bytes. Level 2 cache, sometimes called external cache, is 256 kbytes to 1 Mbytes in size. The size of main memory varies from 32 to 128 Mbytes.

All components of a workstation are connected via several buses as shown in Fig. 1. The processor and memory are connected by a system bus, and all I/O devices share a common I/O bus. Simplicity and low cost are the major merits of such a structure, which has been adopted by most workstation manufacturers.

However, with the increase of processor computing power and high bandwidth requirement of multimedia devices, buses are becoming the bottleneck, and their performance improvement always lags behind the need. Some new architectures have been proposed. For example, Sun developed the ultra port architecture (UPA) (Fig. 2), a crossbar-switched interconnect that enables multiple simultaneous data transfers. In its O_2 products, SGI uses the unified memory architecture (Fig. 3). The high-speed multiported main memory is at the heart of the system. All data types required by various subsystems such as the monitor and video I/O are stored in memory. Each subsystem accesses the memory through high-speed ports designed to ensure that no data starvation occurs on any port. This architecture allows more efficient use of the entire system memory and increases performance by reducing the buffer copies among those components.

Most multiprocessor workstations adopt the symmetric multiprocessor architecture (SMP) because of its simplicity and low cost. The architecture is very similar to that in Fig.

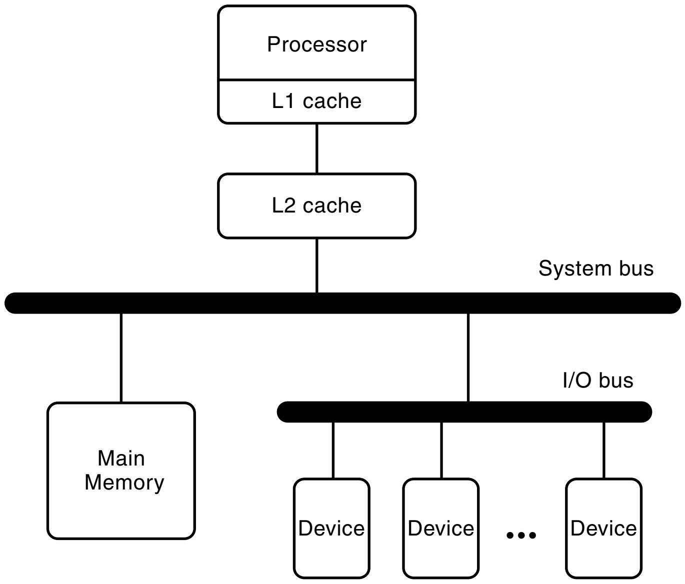

Figure 1. Architecture of a workstation. It consists of one or more processors associated with one level or multiple levels of caches. Memory is connected by a system bus. Other devices such as disks and input/output (I/O) devices are connected by I/O buses.

1 except for the fact that there is more than one pair of processor and cache attached to the system bus. Compared with single processor machines, multiprocessor ones require a higher bandwidth of the system bus. The locality property of most programs allows each processor to access data often in its own cache, thus reducing the bandwidth requirement. However, when the number of processors increases to more than 8, a hybrid architecture consisting of several buses or a crossbar-switched interconnect is usually used to balance the workload and to reduce the bus bottleneck effect.

SOFTWARE ENVIRONMENTS

Operating Systems

UNIX is the operating system used by most workstations. It is a time-sharing multitasking system. Different manufacturers may provide their own variants of UNIX in their machines. For example, in Sun SPARCStations, the operating system in Solaris, which is based on UNIX System V release 4; however, SunOS 4.x is based on BSD UNIX. SGI workstations use Irix. IBM has its own variant, AIX, and HP has HP-UX. There has also been an attempt to standardize UNIX initiated by the IEEE Standards Board. The standard UNIX is called POSIX.

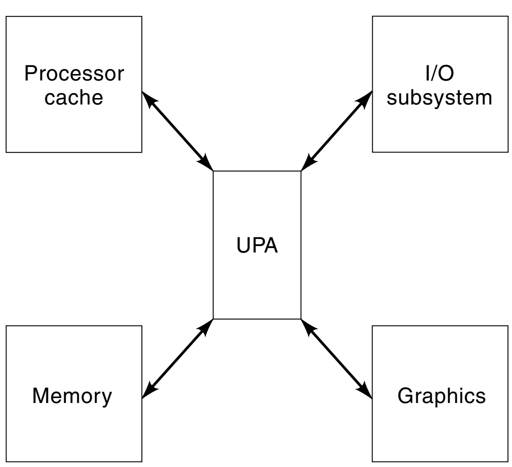

Figure 2. Sun Microsystems' Ultra Port architecture. Rather than using a bus to connect all components, a crossbar is used which allows multiple simultaneous data transfer among components.

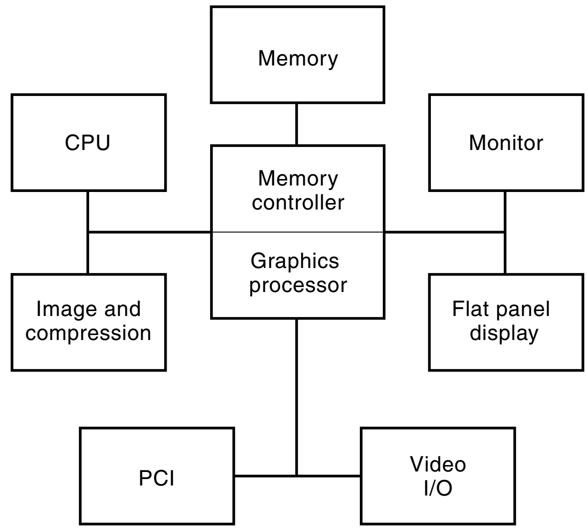

Figure 3. SGI's Unified Memory architecture. It uses a multiported main memory shared by all components of the system and allows a more efficient usage of the entire system memory.

Threads

The concept of threads is important in modern operating systems. It is used in workstations to improve system performance or exploit parallelism when designing and implementing complex applications. A key concept in traditional operating systems is the process. A process is basically a program in execution. It consists of the executable program running on the program's data and stack, which is called the address space of the process. Each process has its own address space and a single thread of control. However, in some cases it is desirable to have multiple threads of control that share a single address space. The major difference between threads and processes is related to the address space. Each process has its own address space, and the interaction between processes occurs through interprocess communication primitives. Threads have their own program counters, stacks, and states, but share a common address space, through which they communicate with each other. Threads are sometimes called lightweight processes because of the low overhead for their creation, deletion, and management in comparison with processes.

Threads were invented to allow parallelism to be combined with sequential execution and blocking system calls (system calls that may be blocked waiting for some events). In a single-processor machine, they may improve the system utilization and throughput. In a multiprocessor system, threads may run in parallel to support the exploitation of fine- or medium-grain parallelism.

Communication

The communication protocol often used to exchange data between workstations is TCP/IP, which is also the protocol used by the Internet. A new IP protocol called IPv6 is being designed and standardized to solve problems encountered such as the significant increase in the number of computers on networks. Layered protocols such as TCP/IP have high software overhead when used to exchange short messages. There are efforts to address this. Usually, workstations are connected with each other by Ethernet networks. Currently, there are two bandwidth Ethernets, 10 Mbps (megabit per second) and 100 Mbps. Asynchronous transfer mode (ATM) (4) networks are more expensive, but provide higher bandwidth (>100 Mbps).

Programming Languages

There are many programming languages available in workstations. The most commonly used ones are C, C++, and FORTRAN 77. Designed in 1972 at AT&T, C is a high-level programming language used at first to write the UNIX operating system, and for system programming. Over the years, the power and flexibility of C have made it a popular language for a variety of applications. C++ is an object-oriented programming language based on C (superset of C). There are two standards of C and C++: ANSI and GNU. Most compilers comply with either of the two standards. The compiler `cc` is an ANSI C compiler, and `gcc` is the GNU compiler, which integrates the C and C++ compilers. Both compilers process input files through one or more of four stages: preprocessing, compilation, assembly, and linking. FORTRAN 77 is a language used by many engineers to do engineering computation on workstations, and some scientific computation libraries are written in FORTRAN 77. The recent portable Java language is already proving popular on workstations as well (5).

User Interfaces

UNIX provides two interfaces for users. The basic one is a command line-based interface called the shell. The advanced one is a graphical user interface (GUI) based on the X Window system.

There are two common shells available in UNIX: the C shell and the Bourne shell. Each is an interactive command interpreter as well as a high-level programming language. As a command interpreter, the shell processes commands the user enters in response to its prompt. As a programming language, it processes groups of commands stored in files called shell scripts. System administrators and users often use them to do file management or other tasks that do not require the use of complicated programming languages such as C. The programming ability of a shell is limited. There is another language called `Perl` that bridges the gap between shell programming and C programming. It is an interpreted script language for easily manipulating text, files, and processes. It provides a more concise and readable way to do many tasks that were formerly accomplished by programming in C or in one of the shells. It is available in the public domain on the Internet, and widely used on various machines such as workstations.

The GUIs for workstations are based on the X Window system, which was first developed at the Massachusetts Institute of Technology (MIT) in 1984. X Windows is an industry-standard software system that allows programmers to develop portable graphical user interfaces. It is a collection of programs and libraries that allow the development of applications for graphics workstations in which the output can appear in multiple windows on the screen. It also provides a means for a program to be run on a remote computer, but to display the results on the local workstation. It is characterized by:

- *Device Independence.* X Windows allows programs to display windows containing text and graphics on any hardware that supports the X Protocol without modifying, recompiling, or relinking the application.
- *Client-Server Model.* In X Windows, application programs are called clients. The clients *talk* (in X Protocol) to a program called the (display) server, which drives the display and interprets keyboard commands and mouse clicks. The client and server may be on different machines, so sometimes X Windows is called a graphical windowing system for computer networks.
- *Mechanism / Style.* X Windows provides mechanisms to support many interface styles rather than enforcing any one policy.

MIT has placed X Windows source code in the public domain. Almost all UNIX graphical interfaces, including Open Look and Motif, are based on X Windows.

The Open Look user interface was developed by Sun Microsystems in partnership with AT&T. It is the specification of the user interface (i.e., how the buttons look, high-level descriptions of how the user interacts with the computer). Openwin is a specific implementation of the Open Look User interface for Sun computers using X Window. Motif is a look-and-feel interface that was developed by the Open System Foundation (OSF) as an alternative to Open Look. Motif was developed primarily because Open Look at the time did not support 3-D widgets. Motif was designed to be similar to Microsoft Windows and IBM's OS/2.

In principle, both Open Look and Motif can be regarded as toolkits of X Windows because they extend it with many widgets to ease the design and implementation of GUIs using X Windows. However, using these widgets, the interface is forced to be in one specific style.

There are even higher level environments to facilitate the development of GUIs. The Tcl and Tk tools are an example. The Tcl scripting language and the Tk toolkit are a programming environment for creating GUIs under X Window. They are easy to learn, very powerful, and contain many sophisticated features. They can dramatically reduce development time for X programmers.

Tcl and Tk were created by John Ousterhout (6). Tcl is a simple scripting language for controlling and extending applications. It supports generic programming constructs such as variables, loops, and procedures. Tcl programs can be easily incorporated into applications, and any application can extend Tcl with new features. One useful extension to Tcl is Tk, which is a toolkit for X Windows. It extends Tcl with commands for building Motif-like (look and feel) user interfaces by writing Tcl scripts instead of C code, Tk consists of a set of commands for creating user interface widgets and geometry

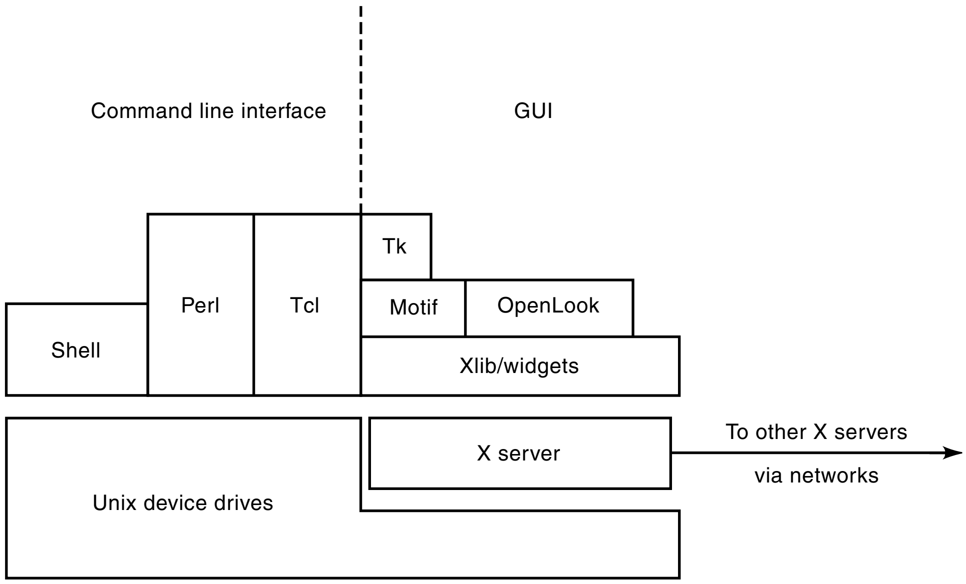

Figure 4. A software environment on a workstation. This diagram shows where each software discussed is in the whole software environment and the relationship between each.

managers to arrange these widgets on the screen. The Tcl and Tk source code is publically and freely available. There are many applications developed by academia and industry using Tcl/Tk. Figure 4 gives the relationship between all the software components discussed above.

APPLICATIONS

Workstations may be used in various fields such as:

1. CAD/CAM
2. Electronic publishing
3. Internet
4. Multimedia
5. Geographic information system
6. Graphics, imaging, animation, and visualization
7. EDA (electronic design automation)
8. Research
9. Software engineering/tools
10. Information management
11. Business-related software

There are thousands of software products available for workstations. Some are provided by the manufacturers, and others by third party venders.

CURRENT AND FUTURE RESEARCH AND DEVELOPMENT TOPICS

With rapid networking software and hardware technology advances, and with the downsizing of the supercomputing industries, networks of workstations are becoming the major computing infrastructure for science and engineering, from low-end interactive activities to large-scale sequential and parallel applications (7). However, networks of workstations were not initially designed for parallel computing. Besides relatively low network bandwidth, there are other important system and computation issues to be addressed for effectively utilizing networks of workstations for parallel computing. In practice, a network of workstations is a heterogeneous and nondedicated system. Heterogeneity represents computing power and architectural differences among workstations. Nondedication implies possible resource sharing and interactions of both parallel jobs and normal workstation user jobs. In addition, it would be too expensive to provide a centralized network scheduler for parallel jobs. Thus, effective coordination of parallel jobs and normal user jobs needs sufficient information about system heterogeneity and job interaction. A practical scheduler should also utilize the existing workstation operating system, such as a UNIX local scheduler (8). Another key enabling technology needed to make high-performance scientific computing on networks of workstations truly possible is to build affordable, scalable, and high bandwidth networks with low overhead communication protocols (7).

Workstation processors are getting much faster, but the performance of I/O is being improved at a much lower pace. Consequently, applications are becoming more I/O bound than computation bound. Multimedia devices require high I/O bandwidth, which also increases the burden for processors. New architectures such as Sun's UPA and SGI's UMA are proposed to improve the overall performance of workstations. Some techniques which were used previously by minicomputers or mainframes are now used in workstations. A specific I/O processor may be used to manage the I/O devices, thus reducing the workload on the main processor. Redundant arrays of inexpensive disks (9) has been proposed to increase disk performance and reliability. People are paying more attention to the development of fast I/O systems.

In addition to traditional applications, the advance in multimedia and network techniques promotes the use of networked workstations in computer-supported cooperative work (10) environments. Applications in this field include teleconferencing, remote education, multi-agent problem solving, and planning.

A major difference between PCs and workstations is related to processor speed and memory space. A workstation traditionally provides more computing power and larger memory space. With rapid development of the PC and with more engineering applications running on PCs, the distinction between PCs and workstations is becoming fuzzier. When multiprocessor PC systems become available, the PC-based systems will expand deeper into workstation territory. Applications and markets of workstations and PCs are quickly merging.

BIBLIOGRAPHY

1. D. E. Comer, *Internetworking with TCP/IP. Volume I: Principles, Protocols, and Architecture,* 3rd ed., Englewood Cliffs, NJ: Prentice-Hall, 1995.
2. D. A. Young, *The X Window System: Programming and Applications with Xt,* Englewood Cliffs, NJ: Prentice-Hall, 1990.
3. J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach,* 2nd ed., San Francisco, CA: Morgan Kaufmann, 1996.
4. D. E. McDysan and D. L. Spohn, *ATM Theory and Application,* New York: McGraw-Hill, 1995.
5. K. Arnold and J. Gosling, *The Java Programming Language,* Reading, MA: Addison-Wesley, 1996.
6. J. K. Ousterhout, *Tcl and the Tk Toolkit,* Reading, MA: Addison-Wesley, 1994.
7. T. E. Anderson, D. E. Culler, D. A. Patterson, and the NOW team, A case for networks of workstations: NOW, *IEEE Micro,* 54–64, Feb., 1995.
8. X. Du and X. Zhang, Coordinating parallel processes on networks of workstations, *J. Par. Distrib. Comp.,* 125–135, 1997.
9. P. M. Chen et al., RAID: high-performance, reliable secondary storage, *ACM Comp. Surv.,* 145–185, June 1994.
10. I. Greif, *Computer-Supported Cooperative Work: A Book of Readings,* Los Altos, CA: Morgan Kaufmann, 1988.

XING DU
University of Virginia
XIAODONG ZHANG
YONG YAN
College of William and Mary

ENHANCEMENT, IMAGE. See IMAGE ENHANCEMENT.

ENTERPRISE RESOURCE PLANNING. See MANUFACTURING RESOURCE PLANNING.