Computer Configurations

JERRY S. HORTON, SENIOR MEMBER, IEEE, AND DAVID P. GROSS

Invited Paper

The current technology of Energy Management Systems computer configurations is reviewed in this paper. Computer system architectures have changed significantly over the years because of changing requirements from utilities and changes in the available hardware and software technology. The paper shows how configurations have evolved and in today's systems there is much commonality in approach. Finally, the current state of design suggests that new and more improved configurations will be achieved in the future that are greater in power but are more simplistic in design and more easily maintained.

I. Introduction

Modern Control Centers require significantly greater computer power than they have had in the past. This growth has been caused by larger network sizes, more applications, and the introduction of full graphic man–machine interfaces. Control systems are required to maintain data acquisition scan rates, man–machine response, and certain applications processing, even under peak load.

Utility operations are becoming more dependent upon maintaining their control systems. The cost of replacement, in dollars, manpower, and disruption of operations, is becoming prohibitive. This makes ease of expansion a prime requirement.

Systems designers have met these needs by designing computer configurations using distributed processing configurations containing compatible computers of different capabilities with each computer matched to its particular task. Different philosophies of design have led to varying approaches based on mainframes, super-minicomputers, and microprocessors.

This is in contrast to the designs of ten years ago, which were often based on dual real-time process control computers. While such machines are still available (SEL, Modcomp, Harris) and are used by some vendors for smaller systems, most larger systems today are based on mainframe computers such as VAX, IBM, CYBER, and UNIVAC.

EMS systems of today are characterized by having a large number of software functions. They have been consistently measured as having 500 000 to 750 000 lines of high-level code over and above the operating system. The majority of the code is in Fortran, and some systems contain PLM, "C," or assembly language.

The larger systems all can be labeled as "large real-time distributed systems," and as such their design and implementation poses many challenging problems.

Manuscript received January 15, 1987; revised September 23, 1987.

The authors are with Energy Management Consulting Associates, Wayzata, MN 55391, USA.

IEEE Log Number 8717951.

II. Configuration Requirements

The starting point in configurations design are the system design requirements. System requirements have evolved over the years and are subject to the perceived needs either by vendors, consultants, or the utility engineers themselves. These requirements are typically characterized by the following:

- custom designed man–machine interfaces,
- extensive computational requirements for solving complex mathematical algorithms,
- high availability requirements,
- heavy communication loads.

The man–machine response times in today's configuration must match the need for human comprehension. Maximum configuration performance must be available under all conditions including emergency operations where the power system itself is stressed.

These peak load configuration requirements are defined as scenarios expected during major disturbances in the power system. This scenario typically includes hundreds of alarms in the first 30 to 60 s (e.g., a "burst"), continued influx of alarms thereafter, and frequent display call-up and supervisory control activity by operators who react to the disturbance. These performance requirements rule out the usefulness of time-sharing techniques, where configurations are designed to maximize the use of the computer equipment rather than minimize the response time to any individual user. EMS systems must be designed to achieve required response time of any individual task independent of the computer system loading requirement.

In addition to needing quick response time to operator actions, the configurations must process a heavy computational burden by solving a large number of sophisticated mathematical algorithms. These algorithms were developed in the 1970s and are standard requirements in most EMS systems today. This computational burden has driven

0018-9219/87/1200-1659$01.00 © 1987 IEEE

most of the vendors today to large multiple CPU mainframe computers.

A high degree of system availability is necessary in EMS configurations. Availability requirements were 99.8 percent for many years, and as computers and electronics have become more reliable, availability specifications have increased to 99.9 percent, or no more than 8 h downtime a year.

Most EMS systems today have a heavy communications burden from a large number of RTUs and communication data links. In general, the solutions have been to distribute communications loads and to maintain high-speed data access to transmitted data.

The major components of the typical configuration are shown in Fig. 1. As shown, a system consists of seven functional components. The requirements and performance implications of each must be carefully analyzed and considered as a whole in configuration design. Each component will be discussed in the following sections.

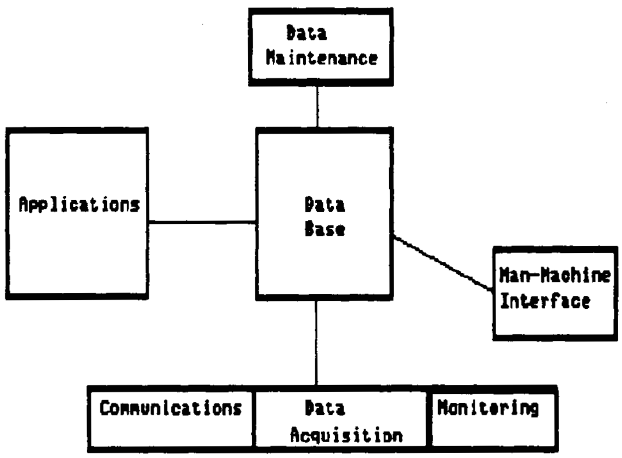


Fig. 1. Configuration components.

III. Applications

The modern EMS system must accommodate a full range of network analysis and scheduling functions. The classic network applications, State Estimator, Load Flow, Contingency Analysis, etc., are all based upon manipulation of matrices of floating-point numbers. As these matrices are too large to manipulate directly by even the largest computers, numerical methods have been evolved to take advantage of the sparsity of the matrices to reduce the dimensionality of the problem. These numeric methods have made the problem solvable in moderate sized computers, but system functional requirements have more than grown to keep pace with available CPU horsepower.

For instance, where an adventurous utility in 1975 required 10 or 20 ac security analysis cases run every hour, today systems may have to run 50 or 100 every 15 or 30 min—and on network models which are on average twice the size. Thus the modern advanced applications have created a need for high-precision storage and fast manipulation of high-resolution floating-point numbers. The processing requirements vary from system to system depending on the network size and the application periods required but are typically in the range of four to fifteen million instructions per second (MIPS) with requirements for long word lengths or double precision arithmetic in places.

A typical set of applications that are incorporated into a modern EMS system are described below and characterized in terms of their demands upon the system. (These are summarized in Table 1).

A. AGC—Automatic Generation Control

AGC is a true process control function, controlling generating units to meet system demand and maintaining system frequency and interchange schedules. As such, its needs are modest. (It was possible to execute AGC every 2 s on early process control computers). Today, the same algorithms are in use as were employed in 1970 (indeed they are digital implementations of analog computer installations in use in 1960), and require, typically, 20–50 ms to execute. AGC, however, must execute as scheduled every 2 or 4 s—"jitter" in its timing can cause control pulses at the units to overlap, which in turn can cause units to "trip" off control. Newer algorithms for generation control based on modern state variable control theory are being implemented commercially for the first time during 1987–1988 and will undoubtedly increase the CPU demands of AGC.

AGC usually also incorporates Economic Dispatch. The traditional La Grange multiplier search of EDC is so efficient that its CPU demands are negligible; newer approaches based on Linear Programming are only slightly more burdensome.

B. Interchange Scheduling

Utilities buy and sell power from each other in large blocks in mechanisms not unlike a stock market without brokers. (In some areas brokerage systems are being implemented.) Utility operators require the ability to review their positions over as many as 100 or more transaction contracts by hour for a week or a month ahead. This requires semi-sophisticated transaction oriented software, and displays which can organize and totalize information in various ways. Display/application response is a critical parameter, as operators are generally reviewing their situation while making transactions over the phone.

C. Interchange Evaluation Economy A

Economy A allows the system operator to negotiate interchange transactions that can be executed within the next hour. It usually executes 10–20 economic dispatches in this process. Typically a 15–30-s response is demanded.

D. Interchange Evaluation Economy B

Economy B determines the feasibility of long-term energy transactions with other utilities. The time frame for the study is typically 48 to 168 h. Economy B utilizes the dynamic programming algorithm of unit commitment, and may require 2–10 min of CPU time, and is also I/O intensive.

E. Unit Commitment

The Unit Commitment program computes the hourly generation schedule over a period of up to 168 h. The generation schedule consists of unit hourly start-up and shutdown times and hourly fuel usage. Dynamic programming,

Table 1 Configuration Loads

Function	Approx. Total (MIPS)	Arithmetic Processing	Entered Interrupts	I/O Bandwidth (bytes)	Database (Trans./s)
SCADA	1–2	0.01–0.1	500–1000	500–1000	5000
AGC	0.1	0.05		25	10
Scheduling functions	0.5–2	0.5–2		50–100	1000
Network analysis	5	4–5		150–300	2000
Training simulation	1–5	1–3	100	100	1000–2000
External communications	1		100	5–20 000	1–2000

LP, or MILP are used, and run times are in minutes. UC is also typically I/O intensive.

F. Short-Term Load Forecasting

The Short-Term System Load Forecasting program produces a probabilistic forecast of the hourly system load for the next seven days. While the mathematics are complex, CPU demands are slight.

G. Network Analysis

The explosion in the use of network analysis codes for monitoring, studying, predicting, and optimizing the operation of the transmission network has alone driven much of the need for compute capability in these systems. All of the various functions, while they may have different operational and mathematical objectives have at their heart the solution of steady-state 60-Hz complex-variable representations of the power system. They all have, therefore, similar steps in their execution:

- Computing an objective function from state variables (such as bus mismatch in a load flow).
- Computing a gradient, derivative vector, or solving LP to adjust state variables, using sparse matrices.
- Solving for new state variables (voltage and angle) via repeated solutions of sparse matrix equations.
- Storing and retrieving intermediate results.
- Computing and storing detailed final results for tabular and graphic display.

The programs vary primarily in the mix of CPU and I/O they require, how often they are used, and what constraints are placed on their usage. Table 2 describes these parameters for each. The programs all have CPU and I/O loads

Table 2 Network Analysis Computer Load (1 MIPS CPU, 500 Busses)

Program	CPU Time (s)	I/O Time (s)	Frequency (min)
State estimation	5–10	10–20	5–30
External model	10–20	10–20	15–30
AC security analysis	21–26	3–5	15–30
DC security analysis	3–16	3–5	5–30
Voltage scheduling	70–80	5–10	15–60
Security dispatch	22–30	5–8	15–60
Load flow	20–30	5–10	60–90
Optimal power flow	70–100	5–10	60–90

which vary directly with the size of the network being modeled. For large networks, the use of sparsity makes their behavior nearly linear with model size, if the coding is efficient. As with the use of more powerful CPUs, the availability of more efficient algorithms is quickly used not to reduce the computer cost and power required but to increase the analytical functionality demanded.

It has long been surmised that specialized equipment such as array processors could be used to reduce execution time required by these programs. A number of factors have precluded this approach from being universally accepted by EMS vendors. Array processors, however, can be made useful in network applications by off-loading main CPUs for special calculations if applications in the main CPU are closely coordinated with those in the array processor. Algorithms can be made resident in the array processor or selected parts of algorithms may be distributed in the array processor. The computational improvement is dependent upon the algorithm selected, the input/output burden to and from the array processor, and the capabilities of the software algorithms available in the array processor. The decision to use or not to use an array processor is mostly a commercial one. Using an array processor, a smaller main CPU can be used with the lower cost/performance ratio as a larger CPU. Thus depending on the cost and performance of the main CPU, the array processor may or may not be justified.

H. System Operator Training

This function is used to train new system operators and to provide refresher courses for existing personnel.

The training is performed using a system operator training simulator that simulates the action of the power system as accurately as possible. Simulators include economic, as well as security training capability. As these must execute all EMS functions as well as simulate system behavior in real time, they typically require a system configuration "parallel" to the EMS itself—either the "back-up" portion of the system or a third computer subsystem.

A major decision in configuration design is whether to require a "stand-alone" training simulator or one that executes in the "back-up" computer. Many utilities require a stand-alone simulator because of the high computer utilization needed by the training simulation. This high utilization would be prohibitive for software development activities that normally occur in the back-up machine. "Stand-alone" simulators do require a complete set of hardware front ends for data acquisition simulators and consoles, so this approach does impose significant additional cost to utilities.

I. Data Collection and Storage

It is increasingly desired that the EMS be capable of recreating, in sufficient detail for examination and analysis, anything of significance on the system. This has led to two widely accepted functions—Historical Data Storage and Retrieval and Post-Disturbance Review. The former captures selected data (such as all telemetered data and all generation control data) at "reasonable" rates such as once every 1-5 min and archives them to disk. The principle design problems have been to minimize CPU and I/O activity associated with this function; to avoid the need for tape storage more than daily; and to provide for retrieval and replay of historical data even after the structure of the basic system database (say—a new RTU is added) has changed. The advent of cheap optical write-only memory will ease some of the design constraint with the result that history requirements will increase.

Disturbance Review, on the other hand, is intended to capture system data at a high rate during a disturbance on the power system. It is aimed at assisting in after the fact "what happened" analysis. It is as though the History function is kicked into high gear during a disturbance. Because the period just prior to a disturbance can be critical in understanding it, this function actually continuously captures a pre-disturbance moving window of data.

One architectural innovation in use by at least one vendor is to have the remote terminal units perform the disturbance data collection function. This off-loads the host CPUs but creates a synchronism problem of activating all RTU disturbance collections when one is triggered.

J. Sequence of Events

An important system function is to capture high-speed power system events such as relay operation and circuit breaker tripping to a resolution fine enough for after the fact analysis. Modern RTUs are capable of capturing and time tagging such events with a resolution of 1 ms, and the system as a whole has to guarantee 8-ms accuracy between RTUs. The goal is to provide 1/2-cycle (or 8.3-ms) resolution between events, as most operations will "sequence" themselves on a step basis of power system cycles. SOE places more of an electronics design constraint on the communications system than a CPU system architecture constraint on the central EMS.

IV. Communications

The EMS systems must obtain data from the outside world. These data come from remote terminals at substations or from other computer systems. The data rates or data acquisition have traditionally been rather slow (1200 Bd), but the trend in the industry is to faster communications (1200 to 2400 Bd for RTUs and up to 9600 Bd for inter-computer communications). What has been lacking in speed has been more than made up for in volume. Typical systems are requiring simultaneous communications for over 100 circuits, while some of the bigger systems require simultaneous communications of up to 500 circuits. The data received over the circuits are very different in structure to the high-resolution floating-point data required for applications. The field data are typically 1 or 2 bits in length for status data and 12 bits in length for measurements. The communication protocols are byte-oriented with interrupts required for each byte. With 100 to 300 circuits operating concurrently at 1200 Bd, capability to handle from 15 000 to 45 000 interrupts per second as well as capability to apply 0.5 to 1.5 MIPS of processing of low-resolution fixed-point data is required.

All modern EMS architectures, as might be expected, use some form of microprocessor-based communications front-end for remote terminal communications. Larger systems typically have multiple front-ends. Key architectural issues and constraints include:

- Ability to handle multiple different (and only semi-standard) RTU protocols.
- Ability to synchronize all of RTU's internal clocks to within 10 ms for SOE. (This typically precludes the use of a computer vendor's standard communications controller without modification, as the host CPU usually cannot be relied on to provide a sufficiently accurate signal. Instead, all front-ends are usually synchronized directly to an external satellite clock.)
- How additional functionality performed in the front-end affects 12- to 32-bit engineering conversion, limit checking, dead-band change detection, etc. The more it does, the more complex its database becomes and the more complex the interface between it and the host CPU becomes—but the more it off-loads the CPU.

The inter-computer communications links are rapidly moving towards industry standards. Except for older pool systems (which historically are based on IBM binary synchronous protocols), ISO X.25 implementations are universally used, and groups within the power industry are moving towards standards for the higher layers of the ISO model. Since all mainstream computer industry suppliers today support X.25, it is typical that in an EMS the servicing of inter-computer links is performed by computer manufacturer's standard gateway products—and, of course, this is the most desirable approach for the utility and the EMS vendor.

While these gateway products off-load the lower layer communications from the host CPUs, the loading of the higher layers—collect or store data, run applications, etc.—must still be carried out in the host. As the more complex links are only now being implemented, the actual CPU and I/O demands of them are poorly understood and are a potential architectural concern. It is fair to say that the functional capability of the links (in terms of information transferred and programs to be run to deliver or receive this information) is today far greater then the initial uses to which the links are put. Today, these links are more often a complex appendage to the system than an integral and major part.

A major issue with data communications, especially from RTUs, is whether data are transmitted continuously or on change (by exception). This subject is addressed in the paper on data acquisition elsewhere in this issue as well as in a section of this paper.

V. Man-Machine Interface

The operator man-machine interface for EMS systems typically consists of a number of consoles, each with several

CRTs and a keyboard, the CRTs being process control or industrial colorgraphic monitors. The display generators for these units are capable of presenting sets of special graphic characters for portraying station diagrams and the like. With these units, the man-machine interface burden on the host could be large, both in terms of display presentation on call-up and in terms of data update on displays in real time. Typical performance constraints are presented in 1 s or less and updated at 2-4 s, although the former is more typically honored in the breach, as systems in the field often are "slow" at display response.

A typical man-machine interface configuration is shown in Fig. 2. The man-machine interface function is distributed

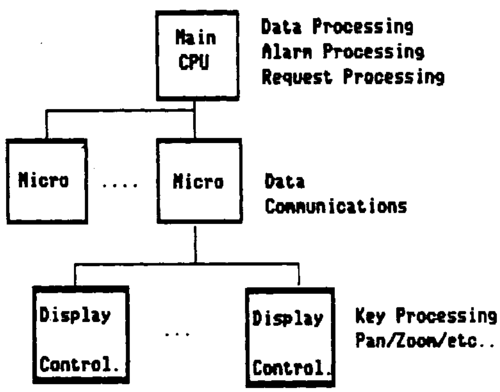


Fig. 2. Man-machine configurations.

over several processors. Generally, the function is distributed as much as possible to the lowest level of processors in the system to achieve a parallel processing environment. However, because of the need for central storage of data for applications and access of that data by man-machine interface, distribution of function to higher level CPUs is unavoidable. It is clear that high inter-computer data transfer rates with fast availability of data from disk and memory are critical.

While efficient software design and implementation is crucial, that man-machine interface can impose heavy demands on disk accesses, CPU utilization, and I/O to the display generators. For this reason many vendors have gone to distributed approaches with some (such as encoding dynamic data) or all of the man-machine interface workload being performed in outboard micro- or minicomputers. As with almost any product today, the human interface largely determines our perception of the quality of the product. For this reason, display response time is a vital measure of system performance and is the focus of concerted optimization by vendors.

Full graphics man-machine interfaces are the state of the art. These subsystems use workstations from the CAD-CAM industry to form the basis of the man-machine interface. These workstations offer the potential for higher resolution, better graphic presentation of data, and the use of capabilities of image transformation such as pan, zoom, clutter/declutter, windowing, etc.

Vendors are committed to these systems today and are investing sizable resources in their development. However, it is fair to say that today the new hardware is primarily being used to provide pan, zoom, and windowing on existing man-machine interface philosophies, and that until this development is complete radical improvements in man-machine interface through full graphics will have to wait. Full graphics systems require large computational capability and vast amounts of data. While all system processing requirements are not yet understood, some statements can be made. A full graphic representation of a typical limited graphic one-line diagram can consist of 3000 to 5000 vectors each of which requires 10 to 20 bytes of storage. Presentation can take half to one million instructions to translate, clip, declutter, and otherwise process these vectors. Similar resources are required when panning and zooming graphic diagrams. When one considers that a single graphic diagram may show the equivalent to 100 to 300 limited graphic one-lines, it becomes evident that millions of bytes of storage and many millions of instruction are required for each active CRT on an EMS system. The trend in EMS systems is towards larger man-machine interfaces and towards larger numbers of consoles and CRTs. Typical numbers run from 5 to 50 consoles, each having two to four CRTs. With a typical system of 20 consoles with one CRT active per console, there can be requirement for 100 or more megabytes of storage and several MIPS of CPU capability.

Depending on where the "world map," or complete system diagram is stored, there can be major tradeoffs in storage and response speed. For instance, if the entire "map" can be stored in the graphics controller, then the need to dynamically transfer or down-load it from the host is obviated. However, if adequate storage is not available, then relatively long down-load times will cause some display call-ups to exceed today's standards. A critical design factor is then the speed of the data pathways to and from the CRT controller and other CPUs or databases. System performance is still uncertain with regard to this issue, as it is with regard to both host and workstation CPU loadings and workloads.

VI. Redundancy, Fail-Over, and Backup

EMS systems are specified as requiring availabilities of above 99.8 or 99.9 percent. To achieve this, a redundant configuration has been required. Traditionally, completely redundant systems have been provided. However, newer configurations are achieving the required reliability using several computers of the same type with one as backup. This achieves no loss of function with a single component failure.

Systems are typically redundant at all levels from the communications interface to the host, with the only exceptions being CRTs, display generators, and RTU communications. In the last case, perhaps one fourth of EMS installations also provide redundant communications lines.

Database backup or checkpointing is a major architectural issue. Various approaches exist to maintain a current and complete copy of all data in backup memory. The most efficient in terms of CPU loading is a transparent dual write to dual access disk, provided the controller and not the primary CPU accomplishes the dual write. Other approaches involve the primary performing two writes, one to each drive, or a transfer of data to the secondary which then writes to its disk. Backup must capture all data including entries, alarms, applications program results, and so on.

Fail-over is the process of transferring operations from a primary component (such as a CPU) to a secondary case. Process computers of the last decade where able to perform rapid and near-bumpless transfers as the "boot" time was modest for small machines. With the utilization of larger, more general-purpose machines and their operating systems, fail-over times have grown due to the complexity of the systems and the fact that it is generally not acceptable to preclude the use of the secondary computer(s) for off-line work in order to accelerate fail-over times. Achieving fail-overs in 30 s to 1 min without compromising standard operating systems becomes a challenge.

VII. Expansion

In the past, utilities have purchased EMS with an (optimistically) expected life to 10 to 15 years. This meant that each 8 to 10 years, a replacement plan had to be formulated and a new system designed and purchased. This has typically caused not only significant expense, but also severe and unacceptable disruption of daily operations. Today, EMS systems require significant spare capacity in the initial configuration as well as the capability to expand the configuration without affecting daily operations. The expansion into spare capacity within the configuration and the expansion of the configuration itself is a major design consideration in EMS configurations. Expansion is addressed by adding front-end processors, upgrading CPUs, or adding CPUs (in distributed systems). Architectures which allow for easy expansion of all facets of the configuration carry a premium over less open approaches.

There are a number of expansion philosophies that can be argued for configuration expansion. Experience has shown that the hardware cost/performance of both specialized front-end processors and main computers has improved dramatically. One philosophy is to purchase an EMS initially with specialized front-end processors for the life of the EMS system to ensure availability of these parts. On the other hand, the utility can plan on replacing the main CPUs during the system life and can select vendors who can provide future expansion of the mainframe. Since the main CPUs are normally the most expensive hardware elements, the expansion philosophy allows the utility to minimize the capital costs on initial procurement.

VIII. Processing Philosophy

Two approaches to processing data in EMS systems are worth discussing. The first is to process all data all the time; the other, to process only those data which have changed. Either of these approaches requires the following data processing:

- transmission from RTU
- transmission from front-end to CPU
- update on CRTs
- triggering of various applications.

Almost all systems perform some exception processing, such as CRT update of infrequently executed applications results. There is, however, a wide range in the basic SCADA area, with some vendors bringing all telemetered data to the screens every scan and update cycle all the time, and others doing those tasks only on change.

Processing all data all the time obviously wastes resource in quiescent conditions but guarantees a relatively level loading under peak load conditions (except for alarm processing). It also simplifies software designs. Exception processing makes for very low loading under normal conditions but may be less efficient under peak load, and also makes for a more complex software design. The most efficient configuration would appear to be a design which was heavily exception-based under normal conditions but which moved to more efficient processing as system activity increased.

IX. Configuration Design

When one considers the requirements of the typical EMS system, it becomes evident that large amounts of resources are required. An earlier section of this paper summarized the advanced applications which require 4 to 10 MIPS of processing capability using high-resolution floating-point numbers, while data acquisitions and communications require 15 000 to 45 000 interrupts per second and 0.5 to 1.5 MIPS of processing power using 16-bit fixed-point numbers. Full graphics requires further MIPS of processing as well as megabytes of memory for each console.

In the past, such requirements would have presented a formidable problem to the system designer, but today these resources are well within the capabilities of off-the-shelf hardware and configurations. The system designer has only to decide upon the configuration, estimate the loads on the configuration elements based upon measurement from existing system, and then select components to match the performance requirements at minimum cost.

In order to minimize cost, the amount of new software must be minimized. This is the most restrictive single constraint on the designers. High-performance hardware may be available to greatly ease a possible bottleneck, but often software is not available to efficiently use that hardware. The choice of configuration design is thus dictated, not by technical considerations, but by the commercial necessity to recover the capital cost of existing software product.

New configuration designs tend to be augmentations and additions to existing configurations. Use of new hardware tends to be limited to replacement of existing hardware with cheaper, higher performance hardware that is software-compatible with prior hardware.

Designers have faced these problems in many ways and three configurations have been used in past systems. These configurations are centralized, hierarchical, and network configurations. The centralized configuration was used almost exclusively in the 1960s to mid-1970s. In the late 1970s and early 1980s, when high-speed data buses between computers became readily available, the centralized configuration was augmented to become hierarchical. This was achieved by moving data acquisition, communications, and in some cases, man-machine interface to outboard processors, often microcomputer-based. The early 1980s saw the advent of network systems. These were again made possible by the development of high-speed inter-computer buses and network operating systems that allowed software compatibility with software developed for centralized systems.

Today's systems are, for the most part, a combination of the three configurations defined above. Because of changes in computer technology that impact CPU processing power and data transfer rates between multiple CPUs, there is no clear single solution agreed upon in configuration design.

A. Centralized Configurations

A centralized computer system consists of one computer performing applications, data acquisition, and man–machine interface. Until a few years ago, this was the most common configuration in the industry. The type of computer selected for this configuration had to be very versatile, with the ability to perform complex floating-point calculations and, at the same time, handle large interrupt loadings. Mainframe computers can handle the floating-point load but lack the operating system and I/O capability required for the interrupt handling. Real-time oriented minicomputers could handle the I/O loading but fell short in computational ability. The solution was the so-called mid-range computers. These computers are 16 to 32 bits and often have off-board I/O processors and a total capability of 1 to 2 MIPS of processing capability. Redundancy in such configurations was achieved by duplicating the entire configuration and providing two symmetrical computer systems.

The main advantage of this type of configuration is the inherent simplicity and ease of maintenance. All functions operate in the same environment and can obtain data from the same database. Inherent disadvantages are the poor expansion ability. Once the performance of the CPU has been fully used, the only option is to replace the CPU with a more powerful version. Unfortunately, the demands of the EMS systems for processing power soon outstrip the capabilities of even the highest range machines in the mid-range computer systems. This leaves the system designer no option but to replace the entire system.

B. Hierarchical Systems

In a hierarchical system, the system designer faces the design problems by matching the EMS tasks to different types of computers each suitable for its intended task. The hierarchical system makes use of distributing processing to bring the CPU power and memory requirements within the range of that required by the EMS system using cost-effective hardware. The hierarchical configuration consists of the host computer front-ended by processors for data acquisition and man–machine interface. The host computer provides the high-speed high-resolution arithmetic for the application programs while the front-end processors provide fixed-point and character handling, as well as interrupt handling for communications and man–machine interface. The host computers are typically mainframes in the range of 4 to 16 MIPS.

Internal memories in excess of 2 million bytes and external memories in excess of 2 thousand million bytes are not uncommon. The front-end processors are typically based on microprocessors or 16-bit minicomputers as in past systems. These front-end processors are generally configurations of 16- or 32-bit microprocessors. Typically, a 16-bit microprocessor is used to process interrupts and communications protocol and pass the data to a higher level processor. This higher level microprocessor controls the scan, detects changes in status, and converts analogs to engineering units, and ultimately passes a processed database to the host computers.

The man–machine interface uses graphics terminals and front-end microprocessors. A graphics terminal typically drives 2 to 4 CRTs at a console. Vector transformations, clipping, pan, zoom, and declutter functions are processed within the graphics controller. Large displays are held resident within the graphics controller; however, display call-up and display update functions are processed within the man–machine microprocessor.

The hierarchical configuration has several advantages. The first is the inherent power of the host computer. With machines available of up to 50 MIPS or more, computers are no longer a limiting resource. It should also be possible to change the host to a more powerful computer with little effect on the continuity of operations at the Control Center. The front-end systems can also be expanded in place as a function is distributed among several front-end systems. Additional front-end computers can increase the amount of communications and man–machine activity up to the limits of the host computer to accept and handle the data. Redundancy in a hierarchical configuration can be achieved by providing spare front-end processors of each type and a redundant host computer.

There are, however, problems of requiring inter-CPU communications and the distribution of both the function and the database. Also, if the various computers are not of one family, the user must support two system environments.

C. Network Configurations

Network configurations consist of computers in which communications paths exist between all computers in the network. This configuration is similar to the hierarchical configuration with the difference that there can be many host computers and that processing of data does not have to be sequential between computers in the hierarchical path. The network can allow the system designer to select various CPUs of one manufacturer's line and to mix and match them within the configuration depending upon the CPU and I/O power required. Individual EMS tasks may then be allocated to the computers as appropriate. Care must be taken to use machines of sufficient size so that the largest task or linked sequence of tasks that must operate sequentially can be accommodated within a single computer. Data paths can be provided to broadcast rapidly changing data to all machines' resident databases, and mass resident data can be stored in one location and accessed by all machines. In this way, allocating any task to any machine can easily be achieved. Redundancy in such a configuration can be achieved by providing spare machines on the network bus. These machines can take over the tasks of any failed machine. The mass storage and network bus must, of course, be duplicated for redundancy. Up to the present, network configurations have been provided by connecting CPUs and mass storage controllers to a common bus. Each CPU has an address and responds only to messages transmitted to it. A broadcast command is available to communicate with all network nodes simultaneously. Typically, more powerful computers have been selected for applications and less powerful computers for the data acquisition and man–machine tasks. As with the hierarchical configurations, the network computers are front-ended by graphics controllers and channel communications microcomputers as appropriate. Expandability in such configuratins can be achieved without disruption of system operations by adding to or replacing CPUs in the network configuration. The DEC VAX cluster is the most obvious example of this approach.

D. Commonality Among Today's Configurations

As shown above, each type of configuration has strong and weak points that characterize its performance. Configuration designers have considered these strong and weak points in today's design approaches. It is clear in today's systems that designs for EMS systems have a number of key elements in common. These include the following:

Mainframe computers with multiple CPUs available in each computer. The capability to distribute CPU processing load among several CPUs in a mainframe is cost-effective in terms of peripherals available and is logical because of the need for common data by parallel functions. This also offers a good expansion alternative when multiple CPUs can be added to the same mainframe.

Most systems have data acquisition functions distributed over multiple 16/32-bit front-end processors. This serves two purposes; first, it increases the throughput capacity of the EMS; and second, it isolates the high interrupt load required for data acquisition functions from the main CPU load.

Most of the man–machine functions are distributed to front-end controllers. Most vendors have distributed the processing as much as possible and, to the extent possible, have isolated the interrupt processing of man–machine requests to individual controllers. The amount of function distribution and how the functions interact are distinguishing factors among system vendors.

Most systems have a centralized database concept, although some are more centralized than others. All systems maintain a memory resident telemetered database and some maintain all relevant database items in one location in the configuration. It is not uncommon for some vendors to have a centralized database concept where multiple copies of data exist in several CPUs that are broadcast from a central location.

All EMS systems employ relatively high-speed intercomputer links and disk drives and/or networks to transfer data between CPUs. This need is apparent in two areas. First, high-speed transfer of data-acquisition data from front-end processors to main CPUs; and, second, in procesing man-machine requests between CPUs. The need for speed of throughput is particularly important in interfacing with powerful graphic controllers. The design requirement of full graphics has generated the need to transfer large amounts of data to and from the controller and other CPUs.

All EMS systems have adopted powerful full graphic controllers. Many systems employ controllers that are more powerful than minicomputers used in configurations in the early years. A common design approach is to use large CPU memories and maintain much of the dynamic data memory resident. The memory available today is 32 Mbytes and above. This allows for fast access to the data and allows many programs to be memory-resident, thus improving the overall throughput available without adding complexity to applications software design.

There is a tendency in the industry today to reduce the complexity of the software required to develop and maintain systems by applying more powerful hardware solutions. The result has been a progressive approach to system design that will no doubt continue in the future as more advanced hardware technology is available.

X. Database Considerations

The differences between the three types of configurations have a major impact on the database structure. The centralized designs allow the database to be both stored and accessed in a single computer. Each element of the database needs to be stored just once and is independent of the number and type of applications using it. The database is always consistent.

With the advent of hierarchical systems, the problems of database management became severe. It was necessary to store items of database required for applications executing in a particular machine within that machine. If data were required by more than one application and those applications were running in different parts of the configuration, then the data had to be stored twice. The distribution of functions within the configuration led to a distribution of the database within the configuration. Comprehensive database generation of maintenance facilities to allow the master database to be created and edited on one machine, and then individual databases for each machine within the configuration created and down-loaded into the various machines in the configuration, are required. Facilities are also required to make sure that when a database element is changed in one machine, it is kept consistent as on-line changes are made. Such database maintenance programs are essential to maintainability of the hierarchical system, and from a practical viewpoint, without them a hierarchical system may not be viable.

Network systems can be designed such that each computer has its own database, and therefore may contain the same problems as described for hierarchical systems. The network structure with a broadcast capability will also allow centralized database with all computers obtaining data from the same source to be used. The selection of which method is to be used in a network configuration is a tradeoff between simplicity and loading. The distributed database is more complex but will require less activity on the network bus. The centralized database is significantly less complex but tends to maximize use on the network bus.

XI. Configuration Analysis

If the EMS logical model of Fig. 1 is augmented by data rates and CPU loads which were developed in earlier sections, Fig. 3 results. Hypothetical examples of each of three configuration types result by partitioning the model in ways that the different configurations partition EMS tasks. These are shown in Fig. 4.

Examining Fig. 4, we can quickly draw a few conclusions. First is that the fully centralized configuration appears to be swamped with interrupt processing, I/O, and CPU workload. This is the case in reality and is the reason why such a design is no longer used for large systems. Second is that there are reasonable data rates from front-end or outboard processors to the host(s) in both the hierarchical and network configurations. In both cases, designers have a choice

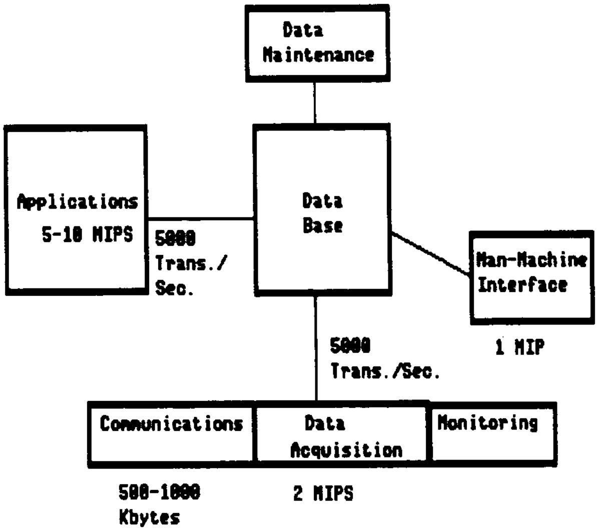


Fig. 3. EMS configuration model.

of methods to interface the microprocessor subsystems to the hosts, as shown in Fig. 5. There are advantages and disadvantages to each method, depending upon the specific hardware characteristics of the host.

The radial-parallel channel approach has the advantage of very high speed, and if the host has I/O processors (IOPs) with direct memory access, it also totally unloads the host CPU. This configuration is typical of CDC and Gould equipment, among others. It requires parallel channels for each front-end processor so that for large configurations the host must have a number of IOPs. It does have a disadvantage in that it is a relatively closed architecture dependent upon manufacturer-specific hardware/software interfaces.

The radial-serial channel approach must also have a number of channels, but probably does not fully unload the host, may be less expensive, and is a more "open" approach. This approach is typical of early systems using DEC VAX machines.

The third approach, using LANs (usually Ethernet), is the most open of the three, permitting easy modular expansion. It also, today, provides relatively high host unloading as most computers have Ethernet servers which unload the host of I/O and which may not conflict with the host CPU for memory access. Most importantly, it is possible to connect many different Ethernet devices to what is now a standard LAN, especially if the software protocols are widespread industry semi-standards. This approach is the one favored today for both large hierarchical and distributed systems, and is finding its way into small SCADA systems. It also is a popular way to interface to the new graphics workstations. The potential drawback is that LAN throughput can be quite a bit less than the hardware bandwidth due to contention or collision.

The third obvious conclusion is that disk access is potentially a bottleneck in all of the configurations. The only configuration alternative is one which distributes the database across multiple, independent, disk I/O servers. As any CRT must have access to any data item, this approach, as noted above, is replete with design problems unless expensive solutions (such as multiple complete database copies) are taken. Again, this matches both intuition and practice. Many

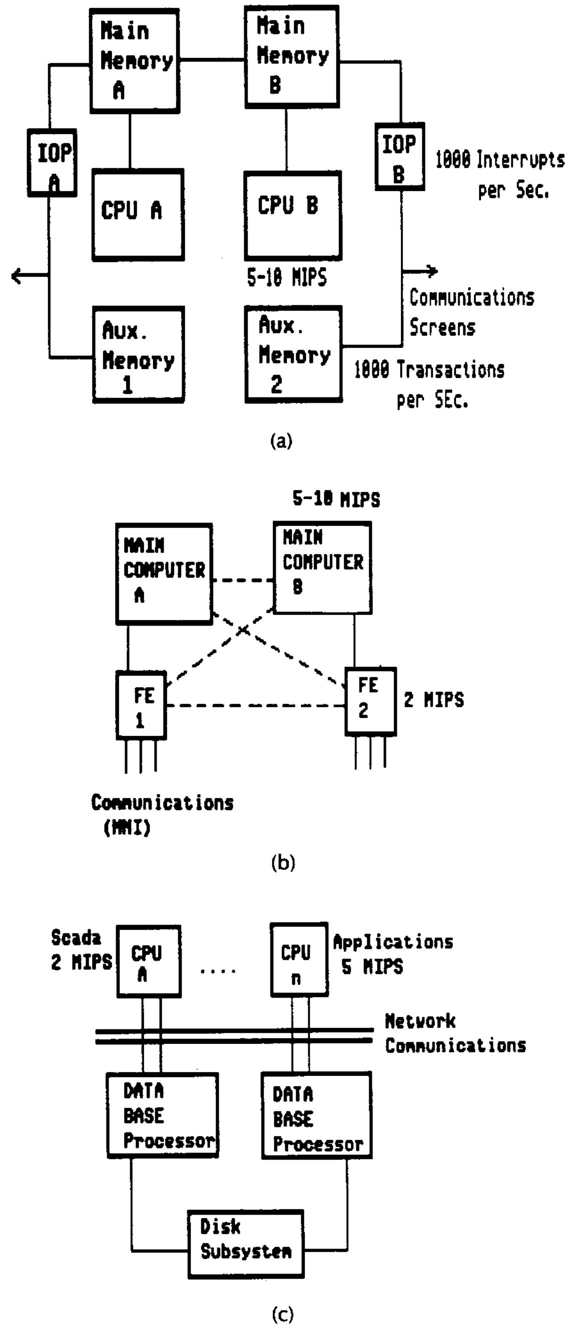


Fig. 4. (a) Centralized configuration. (b) Hierarchical configuration. (c) Network configuration.

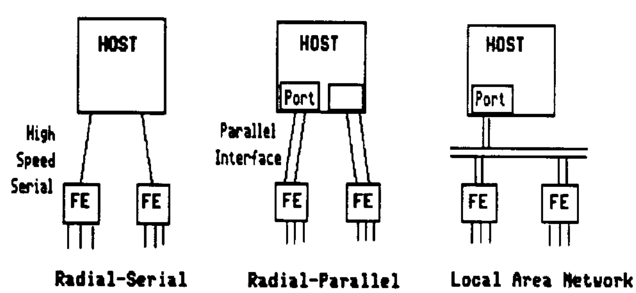


Fig. 5. Front-end interfaces.

of the historical problems of these systems, especially CRT response or alarm processing speed, are related to disk I/O saturation. Both the hierarchical and the distributed systems have this problem, whether the disk server is the host CPU or a network of computers. One partial solution is to give each processor in the system its own dedicated disk which is used for strictly local functions such as task swapping for the operating system.

A fourth, less obvious, conclusion, is that large applications with response time constraints will continue to require single, large, fast CPUs to execute them. That is, if an optimal power flow which takes 8 min on a 1-MIPS machine must complete in 1 min to be useful, then a roughly 8-MIPS CPU is required. While research is going on in applying parallel processing to these problems, it is a theoretical approach today and unlikely to be implemented soon, barring a breakthrough. An important note is that while the jump from 1 to 8 MIPS is practical today, eightfold increases in disk I/O are harder to come by—most manufacturers have only 1 or 2 disks and controllers at their disposal, with much narrower performance ranges. (Access time, not I/O bandwidth, is the limiting factor.) Consequently, disk bottlenecks are more likely to be a serious problem in the larger systems configured with 2 or 4 CPUs, so that software design to minimize disk I/O becomes a paramount concern. A major evaluation criteria in assessment of configuration performance are the capabilities in interrupt processing and CPU processing for floating-point calculations. It is important to note that interrupt processing in itself is not of major concern, but the disk I/Os and context switching of operating system tasks are. The configuration must be carefully evaluated to insure that heavy CPU processing to support applications can coexist with heavy interrupt processing loads.

The alarm processing function and CRT data requests functions are particularly important in evaluation of the configuration. Generally, most of the interrupt processing for such functions as data acquisition and man-machine keyboard requests can be distributed over a number of processors. Alarm processing and CRT data requests generally must occur on CPUs where the total database exists and heavy processing floating-point calculation loads may exist. For those configurations that have applications completely isolated in separate computers this is not a major concern, of course, but at the expense of a more complex configuration. For other configurations, the computer CPUs must in essence turn into a real-time high-interrupt processing loads during peak loading conditions in the power system, possibly at the expense of slower response of the applications. Several capabilities are important under these conditions. First, fast context switching between memory-resident tasks and databases must be available. Second, a larger CPU must be employed to account for heavy floating-point computational load being completed on time. These capabilities will allow the tasks to be rescheduled to quickly serve the interrupt load, and yet still meet the heavy application loads. Note also that the operating system must be efficient enough so that the overhead in processing of the great number of tasks and context switching is minimized. The importance of the operating system and how the heavy floating-point CPU loads and interrupts are processed are critical to the successful operation of configurations where these types of loads are centralized in one CPU.

The analysis of the interrupt and floating-point loading of configurations is further complicated by the availability of multiple CPUs in a particular computer. One must determine how tasks are allocated to each CPU and which CPU has operating overheads associated with it. Further, whether certain functions can or cannot be scheduled in a selected CPU must be considered.

XII. Future Developments

The capabilities and performance of computer hardware is continually improving. It is hard to predict exactly how fast the new technology is moving. In the past, designers have consistently underestimated the rate of change and most have not accounted for cost effectively changing to the new technology. Indeed, it appears that some lessons are to be learned from past experiences.

EMS system will tend to have open communications architectures that will allow new hardware to be connected and may allow for most distribution of functions and the use of LANs.

Since memory is becoming increasingly more inexpensive and the price/performance ratio of CPU processing will continually decrease, the configurations design will emphasize maintainability of the system in terms of best software and hardware maintenance at the expense of performance considerations. Configurations can be designed to be simpler for software considerations and yet meet the performance loads necessary for power system controls.

The above also complements the philosophy of using standard hardware units available from major computer manufacturers rather than the use of special-purpose devices. EMS systems will focus on new power applications that serve the dispatcher directly complemented by hardware and software from the mainstream of the information technology industry. This argues well for a future where modular enhancement of systems is a standard practice, rather than a once-a-decade replacement.

XIII. Summary

Configuration requirements, design, and analysis of EMS were reviewed. It was seen that configurations design has changed greatly and it has been adapted to the current computer technology available. In the past, the conventional configurations used were the centralized, hierarchical, and network systems as shown. The current designs are a combination of these three, but they are especially designed to take advantage of the computer tools available today. It is clear, that requirements such as full graphics and the more common need for power applications, such as network analysis, have also served to greatly influence the approach designers have taken in design. Finally, it should be noted that today's new configurations are still yet to be tested with years of field experience and so the coming few years will hold much excitement for the utility industry.

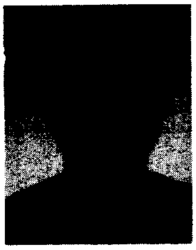
References

[1] J. W. Evans, "Survey of EMS architectures," presented at the PICA Conf., 1987.

[2] R. D. Burn, P. Kilbuurne, and R. Masiello, "EMS cluster configuration," presented at the PICA Conf., 1987.

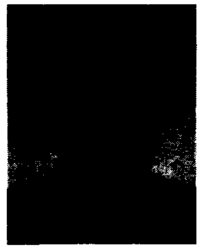
[3] H. Daniels and N. Mayur, "More than mainframes," *IEEE Spectrum*, vol. 22, pp. 54–61, Aug. 1985.

[4] J. C. Russell, R. D. Masiello, and A. Bose, "Power system control center concepts," presented at the PICA Conf., 1978.



Jerry S. Horton (Senior Member, IEEE) was born in Charleston, WV, on September 6, 1947. He received the B.S.E.E. degree from Louisiana State University, Baton Rouge, in 1972, the M.B.A. degree in industrial management, and the M.S.E.E. degree from University of Tennessee, Knoxville, in 1974 and 1975, respectfully, and the Ph.D. degree in electrical engineering from Virginia Polytechnic Institute University, Blacksburg, in 1983.

From 1977 to 1982, he worked for Control Data Corporation as a Senior Consultant responsible for development and implementation of Network Analysis programs. From 1982 to 1984, he was a Lead Consultant and General Manager for ECC, Inc., a consulting firm for electric utilities. In 1984, he became the Chief Operating Officer of Energy Management Consulting Associates, Inc., a consulting firm specializing in energy management systems, and President of Engineering Software Associates, and educational software company. His research interests include configuration systems design, advanced computer applications in power systems, and microcomputer technology.



David P. Gross was born in England in 1945. He received the B.Sc. degree in mathematics from the University of Adelaide, Australia, in 1965.

He was employed at the Weapons Research Establishment and co-authored several papers on the digital simulation of tactical weapons systems. In 1971, he joined the Energy Management System Division of the Control Data Corporation. As Manager of Applications Engineering, he was responsible for the design and implementation of some of the most advanced control systems of the period. He has co-authored IEEE papers and been invited to participate in numerous seminars and panel discussions on the subject of power systems control. He is currently co-founder and Chief Executive of EMCA, Inc., a consulting firm to the Electric Utility Industry.