Session I: Real-Time Support

Chair: Ralf Guido Herrtwich, IBM European Networking Center

The first workshop session was dedicated to the issue of real-time support for multimedia applications. Audio and video are commonly referred to as time-dependent continuous media. Their timing dependencies need proper support by the computer system. This calls for the use of classical techniques from real-time computing. Yet, audio and video have slightly different requirements than traditional real-time applications.

During the first talk, Jim Hanko from SUN Microsystems presented a paper co-authored with Eugene Kuerner, Duane Northcutt, and Gerald Wall on "Workstation Support for Time-Critical Applications," introducing the topic. The initial thesis of the talk was that while today's workstations have a great deal of computational power, this power cannot be effectively delivered to support multimedia applications because system resources are not organized and managed in the necessary manner.

Existing workstations provide abundant CPU capacity, but poor I/O support. While in traditional computing the lack of I/O power can be masked by caching, continuous-media applications that do not reuse data cannot benefit from it. Workstations do not dedicate resources to I/O processing as mainframes do. The single CPU has to help out to perform I/O, constituting a performance bottleneck. That the CPU is scheduled based on "fairness" rather than urgency worsens the problem.

The authors point out that existing real-time systems cannot be used to overcome this problem because they rely on deterministic behavior and known load, assumptions not well suited for a workstation environment. Their statement that existing real-time scheduling techniques such as rate-monotonic scheduling are not appropriate for multimedia has, however, generated opposition from the audience.

A new resource management technique, Time-Driven Resource Management (TDRM), is proposed. The technique bases its decisions on the requester's deadline, importance, and expected resource requirements. The technique does not provide sharp guarantees for a certain quality of service, it rather encourages "graceful degradation."

The matter of how a resource management technique should affect system behavior was subject to major discussion. Many reservation schemes proposed (e.g., those from Anderson or Ferrari of the University of California at Berkeley) provide a fixed quality of service and reject new service requests when the provision of the service would endanger previously given guarantees. To a user, this may mean that his request for retrieving a video is turned down. Such a system can be compared to the telephone system: If a busy signal is received, the user will have to wait and dial again.

Adaptive policies provide more flexibility, but cannot guarantee fixed quality levels. It was mainly agreed that an ideal resource management technique should be a combination of both cases. Some fictitious "costs" can be chosen to decide which ap-

plication can use resources. If conflicts occur, the application (not the system) should decide on how to cope with the reduced bandwidth available.

In the second talk of session, Kevin Jeffay presented joint work with Donald Stone and Donelson Smith at the University of North Carolina at Chapel Hill on "Kernel Support for Live Digital Audio and Video." The main objective of this effort is to provide DVI-based video conferencing across Token Ring. The work is rooted in the design of a new real-time kernel for multimedia applications.

The kernel called YARTOS (yet another real-time operating system) is based on Kevin's real-time producer/consumer paradigm, which allows to reason about message-passing systems based on the rates of task execution. Tasks are deadline-scheduled and all requests for resources (shared software objects) are directed to the kernel to prevent priority inversions. For a given workload, YARTOS tries to guarantee that all tasks complete before their deadlines and that no shared resource is used simultaneously by more than one task.

A schedulability test is made prior to establishing a video conference. Hence, the system follows the traditional approach of fixed quality of service. However, YARTOS provides information about each execution sequence that would lead to a deadline violation. If a user can certify that the sequence found by YARTOS never occurs in practice, then the system will establish the call regardless of the schedulability test.

The system constructed runs on IBM PS/2 machines across Token Ring utilizing the Action Media 750 hardware. It was found that a delay of 6 to 7 frame times occurs for any end-to-end video and audio transmission. This translates into 200 – 230 ms delay which was found to be sufficient for the intended application. It is notable that the major portion of the delay occurs in the DVI pipeline of frame processing.

The Token Ring was found to be suitable to transport compressed multimedia data. However, it was found that DVI packets should be sent with a higher priority to minimize ring access time. During the discussion it was argued that such high priorities (or more general: real-time scheduling mechanisms) are more essential to support time-critical applications that a reservation process.

The third and final talk of the real-time session was given by Hideyuki Tokuda of Carnegie-Mellon University. His joint paper with Clifford Mercer entitled "Priority Consistency in Protocol Architectures" continued the discussion of network access – or more general: protocol processing – for continuous-media data. The goal is to prevent priority inversions in the communication subsystem, i.e., situations in which a high-priority activity is delayed by the execution of a lower-priority action. Several techniques for protocol processing are compared and analyzed. This analysis provides users with guidelines on the technique to choose, based on the ratio of protocol processing time to context switch time.

The authors have found that for a protocol processing time to context switch time ratio of 5-to-1 or 10-to-1 preempting protocol processing to avoid priority inversion makes sense. With a ratio of 2-to-1 and 1-to-1 their simulation showed that it is not worth preempting the protocol processing. This means in particular that for these scenarios it may not be worthwhile to generate individual threads for accomplishing protocol processing: The traditional UNIX method of protocol processing at the software interrupt level may suffice and lead to better overall response times.

Concluding the session on real-time support, several workshop participants brought up the issue of whether new system architectures would facilitate the handling of multimedia data in real time. The question was asked whether new switch-based rather than bus-based workstation architectures would not be a better solution to the resource contention problem. It was generally agreed that such new architectures would facilitate the introduction of audio and video in computers, but that there are strong user demands for providing multimedia functions in today's systems.

Workstation Support for Time-Critical Applications

James G. Hanko, Eugene M. Kuerner, J. Duane Northcutt, Gerard A. Wall
Sun Microsystems Laboratories, Inc.

Current workstations have a great deal of computational power. However, this power cannot be effectively delivered to support time-critical applications because the system resources are not organized nor managed in the necessary manner. What is needed is hardware and software platform technology that manages (i.e., acquires, processes, transfers, coordinates, and delivers) these time-critical data streams. This research into properly constructing, organizing, and managing system resources (according to principles of time-driven resource management) will enable workstations to solve many problems currently handled by dedicated-function embedded systems or point-solution add-on devices, while retaining the essential characteristics of the distributed workstation system environment.

Introduction

There are entire classes of problems to which today's workstation cannot be applied. This application area is characterized by the inclusion of information or activities whose value to the user is a function of time. Multimedia audio and video, visualization, virtual reality, transaction processing, and data acquisition are example application classes. Current workstation system architecture lacks support for these classes of applications. When problems such as these are encountered, they tend to be addressed in a specialized, *ad-hoc*, manner. One domain where this is particularly apparent is in the area of multimedia applications.

The workstations of today are not able to effectively deliver their power to time-critical applications. The answer to this problem is not give the users the "bare iron" (as is done with embedded systems and PC's), but rather to change the nature of the workstation to extend support for this new set of requirements. To achieve the desired goals, workstation system architecture must evolve in a number of areas: processor power and organization, memory structure and bandwidth, system interconnection schemes and topology, and I/O capabilities.

Problem Statement

To realize the promise of new application areas, workstations must be able to manipulate time-critical data streams. In the past, the time-critical nature of the data was often ignored or handled by the brute-force application of excess resources. However, new and demanding applications are emerging in which this approach cannot work. One of these application areas is videoconferencing, which deals with multiple, continuous, high-bandwidth data streams that must be mutually synchronized.

Time-critical information represents a fundamentally different form of data than is found in workstations today, and imposes a totally different set of requirements on system hardware and software. There are fundamental limitations in the structure of

current workstation hardware and software that prevent them from being effective in dealing with time-critical data. Furthermore, the embedded real-time techniques usually applied to these applications will fail in the workstation environment.

The support of time-critical data does not necessarily conflict with the workstation's traditional requirements, but does impose additional requirements on the system architecture.

Workstation Shortcomings

Current workstations do not provide the proper hardware and software support for applications which require the manipulation of time-critical data streams. The reasons for this are threefold: insufficient resources (both in kind and amount), inappropriate organization of the resources, and inadequate management of the resources.

Workstations have primarily been designed to maximize performance on compute-intensive applications. For example, although today's workstation CPU is faster than traditional mainframe computers, the memory and I/O bandwidth of a workstation is only a small fraction of the mainframe's. This is usually masked by providing the CPU with a large cache and main memory, so that the memory and I/O loading is reduced.

Applications that manipulate continuous, high-bandwidth, time-critical data streams do not have the data reuse property that allows caching schemes to work. Therefore, much greater demands are placed on system resources such as I/O and memory bandwidth, and the effective utilization of such resources becomes increasingly important in supporting time-critical applications.

In addition, some applications need more or different processing capabilities than current workstation processors provide. For example, workstation and mainframe architectures differ in that workstations do not dedicate processing resources to I/O operations as mainframes do (e.g., channel-processors). As a result, any processing associated with I/O must be done by the workstation's main, general-purpose, processor(s). Furthermore, general-purpose processors lack the capacity (in raw operations per second) to perform many functions (such as standard compression of full frame rate video) in real time, thus requiring special processing elements.

Finally, even if the necessary resources exist and are organized properly, the workstation system software is not able to manage the resources properly for this class of application. Resource scheduling decisions are typically based on a "fairness" criteria because of the workstation's time-sharing heritage. Individual resources, such as memory and CPU cycles, are managed in *ad-hoc* ways, with no way to coordinate them to ensure that time-critical activities are completed on time. Because system software also has no concept of time-criticality, all applications run, in effect, open-loop. Therefore, time-critical applications must currently accept the resources as provided to them as the result of seemingly random resource management decisions.

Shortcomings of Embedded Real-Time Systems

PC-class machine designers have addressed the problems of time-critical media applications by turning to an "embedded real-time" systems approach. Within a limited context, this approach appears to meet the applications' needs. However, it is an inappropriate approach for the workstation environment because it requires dedicated resources and deterministic behavior. In addition, a system using this approach must be re-architected when new applications or capabilities are introduced.

Traditional embedded real-time systems statically allocate excess resources for time-critical problems to "guarantee" that resource demands never exceed the supply. This can be seen in PC-class multimedia systems, where the entire CPU is dedicated to a single application.

Furthermore, traditional (strict priority and rate monotonic) real-time scheduling doctrine decrees that you can only use up to approximately 70% of the available processing time and still meet timeliness requirements [Liu 73]. Therefore, the system is artificially limited short of its full potential. This is unacceptable in a workstation environment where arbitrary amounts of application load may be impressed upon the system at the discretion of the user.

The "guarantees" that embedded real-time systems make are predicated on the assumption that all system activities are deterministic or that hard bounds can be placed on all variability. The workstation environment is inherently non-deterministic. For example, the user may activate another time-critical application, or a remote command might arrive via the network. A system architecture that relies on determinism fails in this environment.

Finally, there is no uniform way to handle unanticipated competing resource requirements. This means that the resulting systems are fragile; when separate resource demands are allowed to be made simultaneously, or when new features are added, the entire system must be re-architected. For example, two applications that handle a video and audio stream, respectively, may each have static resource allocations that allow them to run well in isolation. However, when they are run simultaneously by a user trying to have both sound and pictures, the assumptions of determinism and excess resources built into each application are violated. The results are unpredictable. The system must be reanalyzed and tuned before these applications can be used together.

Systems Implications

What is called for here is a general systems solution to the problem of managing time-critical activities. It is not sufficient for the system to simply "get out of the way". Rather, it is the system's responsibility to actively manage its resources in such a fashion as to, wherever possible, permit the applications' time-constraints to met.

The solution is not to cripple systems by reducing them to the level of PC's, nor simply to add mainframe style I/O channels, but rather to extend the architecture of the workstation to fully support these applications.

The Time-Critical Systems research group at Sun Microsystems Laboratories, Inc., is creating hardware and software, platform-level technology that manages (i.e., acquires, processes, transfers, coordinates, and delivers) collections of time-critical data streams.

The approach being followed involves ensuring that the proper resources exist in the system, that they are organized correctly, and that they are managed in a manner that supports the needs of time-critical applications.

Ensure Proper Resources Exist

The group's initial work is centered around a research testbed system that is being constructed in conjunction with David Sarnoff Research Center and Texas Instruments, as a part of a DARPA-sponsored research project — i.e., the High Resolution Video (HRV) Workstation project [HRV 90a, HRV 90b]. Through the development and use of this testbed hardware and software, several basic ideas, including: the implementation of core algorithms, the development of realistic workload generators, and the collection and analysis of system performance statistics are being explored. This involves examining the system architecture of the testbed in depth, and comparing its capabilities to that of today's workstations. The testbed system will also allow other groups (and their applications, within and outside of Sun) to provide us additional insights into the problem area through their use of this system.

The Time-Critical Systems research group is considering the difficulties which arise in the manipulation of time-critical data streams, and devising solutions which provide the necessary capacity to support time- critical applications. The group is also in the process of identifying the bandwidth and processing requirements for the sample applications, which will provide insight into the limitations of (and trade-offs between) processing elements, memory, I/O, and system interconnect topology.

Ensure Resources are Properly Organized

Support for time-critical data streams is a system-level problem which requires a system-level solution. As a part of this work, hardware architectures are being explored which provide support for time-critical resource management at the system-software-level. Also, the mere existence of interesting hardware that enables time-critical resource management is not enough. This research will investigate and provide insights into proper ways of organizing the hardware (i.e., hardware architectural features that lend support to the management of time-critical information).

This effort involves investigating architectures which make use of multiple, concurrent paths to high-bandwidth memories, in order to permit the execution of simultaneous, independent, time-critical activities in future workstations. It is also likely that this exploration into issues of system topology and interconnection may involve the study of different forms of switch-connected machine architectures (e.g., high-speed non-blocking switches, point-to-point links, hybrid bus/crossbar, etc.), as opposed to global-bus-based structures.

To date, this work has suggested that the requirements of multimedia-like, time-critical applications call for globally symmetrical resource accessibility (e.g., as provided by global bus-structured systems). However, the high-bandwidth, stylized information

flow patterns that accompany such applications suggests the need for concurrent operation (e.g., as provided by switch-connected architectures [Tennenhouse 89, Hayter 91]).

By introducing more concurrency into the workstation interconnect, highly asynchronous activities can be scheduled in parallel with less (or little) impact on each other. Alternative paths for data movement may greatly increase system scalability and flexibility. Additional data paths will also allow the exploration of insertion into the system of specialized processing elements such as data compression engines.

Ensure Resources are Properly Managed

Time-Driven Resource Management (TDRM) is our basic approach to solving the time-critical systems problems defined above. We are adding timeliness as a system-supported attribute of the data streams that exist within the workstation.

It is our contention that all resource management decisions should be made on the basis of timeliness and importance of the applications requesting the resources. Our work involves the creation of system resource management software that takes timeliness and importance into account when making its decisions concerning processor allocation (i.e., scheduling), memory allocation, synchronization operations, interprocess communication, I/O operations, etc. The classical arbitrary, priority-based resource allocation mechanisms will be compared to the new techniques and the results analyzed to understand the costs and benefits of such an approach.

According to this model, resource management involves making decisions based upon the requesters' deadline, importance, and expected resource requirements. That is, rather than strictly using deadlines, which fail in overload cases or priorities which are too static, a time and value evaluation will be made to choose among all of the requesters for system resources [Jensen 75].

Rather than trying to guarantee that overload or timing exceptions do not occur, it is assumed that overload is a common occurrence in a workstation. Resource overload should be resolved by application and user-defined policies. The system will detect overload and lateness exceptions, make the appropriate evaluations based on timeliness and value, and then follow user-defined policies (or signal the applications impacted by the resource overload) to resolve or react to the overload situation. It is application programs that should choose how to utilize less of an overloaded resource, and the manner of that reduction, rather the having the system choose for all applications [Northcutt 87].

The system will be architected to directly implement and encourage "graceful degradation" strategies. Reserving resources and handing them out blindly until a "brick wall" is hit is unacceptable since it directly detracts from the base nature of a workstation. Instead, best-effort use of the resources should be encouraged, short-term aberrations noted, and the system should gracefully adjust as overload is reached.

Conclusion

We are applying a methodology of effectively providing, organizing, and managing system resources to the problem of integrating multimedia into the distributed workstation programming environment. We believe that this work will lead to fundamental changes in the nature of workstation hardware and software. Furthermore, the techniques being developed here also have the promise being applicable to resource management in other areas (e.g., high-speed networks).

References

[Escobar 91] J. Escobar, D. Deutsch, and C. Partridge
A Multi-Service Flow Synchronization Protocol
Technical Report, Bolt Beranek and Newman, March 1991.

[Hayter 91] M. Hayter and D. McAuley
The Desk Area Network
Technical Report No. 228, University of Cambridge Computer Laboratory, May 1991.

[HRV 90a] High Resolution Video Workstation Project
High Resolution Video Workstation: Requirements and Architectural Summary
HRV Project Technical Report #90051, May 1990.

[HRV 90b] High Resolution Video (HRV) Workstation Project
System Support for Time-Critical Media Applications: Functional Requirements
HRV Project Technical Report #90101, November 1990.

[Jensen 75] E. D. Jensen
Time-Value Functions for BMD Radar Scheduling
Technical Report, Honeywell System and Research Center, June 1975.

[Lui 73] C. L. Liu and J. W. Layland
Scheduling Algorithms fo Multiprogramming in a Hard Real-Time Environment
J. ACM 20, 1, 1973.

[Northcutt 87] J. D. Northcutt
Mechanisms for Reliable Distributed Real-Time Operating Systems: The Alpha Kernel
Academic Press, Boston, 1987.

[Tennenhouse 89] D. Tennenhouse and I. Leslie
A Testbed for Wide Area ATM Research
In Proceedings ACM SIGCOMM, September 1989.