DPARTS – A Dynamic Parallel Adaptive Multiprocessor Real-Time Scheduler

PROGRESS REPORT 1

Byron Jeff
Karsten Schwan

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

Accesion For

NTIS	CRA&I	☒
DTIC	TAB	☐
Unannounced		☐
Justification	*per ltr*	

By _____
Distribution /

Availability Codes

Dist	Avail and / or Special
A-1	

19950217 048

1

OFFICE OF THE UNDER SECRETARY OF DEFENSE (ACQUISITION)
DEFENSE TECHNICAL INFORMATION CENTER
CAMERON STATION
ALEXANDRIA, VIRGINIA 22304-6145

Jan 12, 1995

IN REPLY
REFER TO   DTIC-OCC

SUBJECT: Distribution Statements on Technical Documents

TO: Office of the Chief of Naval Research
800 north Quincy Street
Arlington, VA 22217-5000
Code 1133

1. Reference: DoD Directive 5230.24, Distribution Statements on Technical Documents, 18 Mar 87.

2. The Defense Technical Information Center received the enclosed report (referenced below) which is not marked in accordance with the above reference.

Progress Report
N00014-93-1-0783
1 July 93- 30 June 94

3. We request the appropriate distribution statement be assigned and the report returned to DTIC within 5 working days.

4. Approved distribution statements are listed on the reverse of this letter. If you have any questions regarding these statements, call DTIC's Cataloging Branch, (703) 274-6837.

FOR THE ADMINISTRATOR:

1 Encl

GOPALAKRISHNAN NAIR
Chief, Cataloging Branch

FL-171
Jul 93

DISTRIBUTION STATEMENT A:

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

DISTRIBUTION STATEMENT B:

DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY;
(Indicate Reason and Date Below). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED
TO (Indicate Controlling DoD Office Below).

DISTRIBUTION STATEMENT C:

DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTORS;
(Indicate Reason and Date Below). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED
TO (Indicate Controlling DoD Office Below).

DISTRIBUTION STATEMENT D:

DISTRIBUTION AUTHORIZED TO DOD AND U.S. DOD CONTRACTORS ONLY; (Indicate Reason
and Date Below). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office Below).

DISTRIBUTION STATEMENT E:

DISTRIBUTION AUTHORIZED TO DOD COMPONENTS ONLY; (Indicate Reason and Date Below).
OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office Below).   •

DISTRIBUTION STATEMENT F:

FURTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD Office and Date
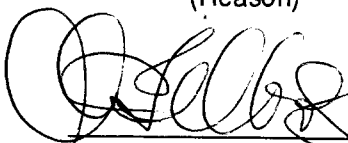Below) or HIGHER DOD AUTHORITY.

DISTRIBUTION STATEMENT X:

DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS
OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED TECHNICAL DATA IN ACCORDANCE
WITH DOD DIRECTIVE 5230.25, WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC
DISCLOSURE, 6 Nov 1984 (Indicate date of determination). CONTROLLING DOD OFFICE IS (Indicate
Controlling DoD Office).

The cited documents has been reviewed by competent authority and the following distribution statement is
hereby authorized.

_____
(Statement)

_____
(Reason)

_____
(Signature & Typed Name)

A.M. van Tilborg

ONR 311
(Assigning Office)

Office of Naval Research
(Controlling DoD Office Name)

800 N. Quincy St.
Arlington, VA 22217
(Controlling DoD Office Address,
City, State, Zip)

1/20/95
(Date Statement Assigned)

1 Research Agenda

Motivation. The area of real-time scheduling has received increased attention during the last few years, in part due to the creation many recent multiprocessor and distributed applications demanding real-time performance. Such applications include robotics [5], multi-media [3] and music [2]. These applications are characterized by their need for real time response, heavy computational load, and unpredictable events that must be processed in a timely manner.

The specific application domain addressed by this research concerns multi-media systems. Specifically, it concerns the real-time synthesis of voice and music, where partially ordered sequences of computations and communications must be processed under given time constraints. We are investigating the dynamic (on-line) scheduling of action sequences triggered by real-time events, where multiple action sequences and different independent segments of a single action may execute in parallel. A real-time parallel digital audio synthesis system serves as a concrete example of such an application. A sample real-time event in this application is a note played on a Musical Instrument Device Interface (MIDI) keyboard.

The proposed DPARTS multiprocessor scheduler is now partially implemented. It will perform scheduling in an event driven manner, in reaction to sequences of incoming events. Furthermore, DPARTS will have the ability to adapt its scheduling during execution. Such adaptations will be performed in accordance with user specifications that address the actions to be taken in response to exceptions like deadline failures and sudden overloads. The intent is to have DPARTS adapt its schedule to maximize some global, time-dependent measure of schedule quality, even under high system loads.

Multi-media and Music Applications. Some recent research has addressed the scheduling and real-time processing of multi-media applications [3]. However, such research has addressed relatively static applications like the transfer of successive image frames across a network [1] or the scheduling of a music application with some fixed number of tasks [2]. In contrast, we are addressing multi-media applications that are highly dynamic, which means that applications must react to unexpected or unanticipated external events. Such reactions may result in the on-line creation of additional tasks that must be dynamically scheduled in reference to existing task sets. Furthermore, for dynamic music programs that react to unanticipated human inputs, timing constraints have to be stated differently from other application domains. Namely, it is not natural to state timing constraints in terms of start times, execution times, and deadlines. Instead, novel semantics have to be defined for 'deadlines', such as semantics that capture notions of lateness based on the perception of different musical instruments by the listener. For example, a deadline for an instrument like a drum approximates the well-known notion of hard deadlines [4] because drum beats must be very precise. On the other hand, for musical tones that are long and drawn out, such as those generated by string instruments, 'softer' deadline semantics are appropriate. We are investigating and specifying such novel timing semantics and using them in the development of efficient on-line scheduling algorithms and schedulers supporting this class of real-time applications.

Distributed Multiprocessor Scheduling. The scheduler being developed in this research is a specific example of an operating system service to be offered in future parallel and distributed systems. Part of this research will address the issues of distributed service design and implementation currently being addressed by other OS researchers. Specifically, we are building on the research described in [6] to construct the DPARTS scheduler to be internally concurrent so it can be easily scaled to different size parallel machines and to varying application demands. This implies that scheduling is performed by multiple concurrent and cooperating tasks.

DPARTS Scheduling Details. Some details about the DPARTS scheduler further demonstrate its novelty. As stated above, each external event may trigger a sequence of processes that jointly handle the event. These processes must execute in an application-specific order. DPARTS must have information about (1) such orderings, (2) the computation time of the involved processes, and (3) about possible alternate or optional processes. We are developing a directed graph representation to contain such information, where the computation times and any other information DPARTS needs to know about processes is stored in the graph's nodes, and the directed edges represent the processes' order of computation. Processes must be scheduled concurrently and to meet real-time constraints, such that the graph's topological order is maintained. This implies that multiple parts of the schedule graph may be active at any point during the applications execution.

The research questions addressed in this work focus on the development of fast heuristic algorithms, on experimentation with those algorithms, and on the use of such algorithms within the music application described above. We are not developing optimal algorithms since all of the scheduling and assignment problems we are addressing have been shown NP-hard. Some specific questions we are addressing include:

- When should DPARTS be invoked? What is the required frequency of DPARTS invocation with respect to the latency of scheduling decisions and the overheads incurred by scheduling?

- Should DPARTS have the option of rescheduling existing tasks if such rescheduling can result in the successful scheduling of otherwise unschedulable process sequences?

- Deadline semantics. In our sample music application, precise deadlines are usually not necessary in order to produce acceptable results. As such we are evaluating what type of sematics are appropriate for addressing the application we are scheduling.

2 DPARTS Status

The design of the scheduler is nearly completed, with implementation currently in progress. Through initial testing we have determined that the single largest degrader of performance is the Unix signal call. The elapsed time of catching and responding to a signal takes nearly half

of our proposed scheduling cycle. Based on these results our scheduler design is based on the non-preemptive execution of threads. The scheduler indicates to each application thread how many cycles it should execute before giving up the processor. Through the use of cooperative task scheduling we believe that much of the performace gained will be retained.

Preliminary responses to the above stated research questions are as follows:

- When should DPARTS be invoked? The DPARTS scheduler will be invoked once each execution frame. The execution frame time will be based on the Just Noticable Difference Perception Time for audio. We plan to start with a 5 millisecond cycle time. Also a DPARTS scheduler will be executed on each processor in a non-overlapping fashion throughout the system. Therefore, initially only one scheduler will be active in the system at any instant.

- What is the required frequency of DPARTS invocation with respect to the latency of scheduling decisions and the overheads incurred by scheduling? And, should DPARTS have the option of rescheduling existing tasks if such rescheduling can result in the successful scheduling of otherwise unschedulable process sequences? Initially, tasks will not be reschedulable. When all of the available time for a frame is filled, then the task invocation request will be passed to the next scheduler with an increasingly higher priority. In this manner, even low priority events such as string invocations will be scheduled in a reasonable time frame.

- Deadline semantics. In our sample music application, precise deadlines are usually not necessary in order to produce acceptable results. As such we will evaluate what type of sematics are appropriate for addressing the application we are scheduling. Furthermore, our initial model for incoming events will assign a relative priority to each different instrument. Within each time frame, available tasks will be scheduled in priority order and any remaining tasks will be passed to the next scheduler with a temporarily higher priority than the initial priority so that increasingly earlier events will be scheduled.

Digital Audio Synthesizer. Our sample application, a parallel digital audio synthesizer, consists of a number of interlinked modules that communicate by passing packets of audio samples. The application embodies modules to perform the following functions:

- Digital Audio Generation (sine wave, FM, wavetable lookup).

- Filtering (a generic digital filter that produces high pass, low pass, bandpass, and notch filter outputs).

- Amplification.

- Mixing.

- Effects (chorusing and delays).

3

- Output.

The user will specify the linkages between these modules, and the application will accept input events that triggers a set of modules which produce the synthesized digital audio output.

Digital Audio Synthesizer Status. The initial Digital Audio Generation, Mixing, and Output modules are complete. In addition, the Input Event Generator that converts MIDIFILES into a stream of incoming events has been completed.

Equipment Status. PCs are becoming increasingly important as actual business and research computing engines. For example, much of the research in Japan addressing real-time applications (the TRON project) is based on PCs. Therefore, our current equipment base is a PC purchased with equipment funds allocated to this work:

- 90 Mhz Pentium, 32 Megs RAM, 17 in SVGA Color Monitor.

This machine is attached via Ethernet to high performance computing and visualization engines, as well as to MIDI input devices and to multimedia output engines. The cost of the items are summarized in the attached expense report.

References

[1] D.P. Anderson and R.G. Herrtwich. Resource management for digital audio and video. In *Seventh IEEE Workshop on Real-Time Operating Systems and Software*, pages 99–103. IEEE, May 1990.

[2] D.P. Anderson and R. Kuivila. A system for computer music performance. *ACM Transactions on Computer Systems*, 8(1):56–82, Feb 1990.

[3] D.P. Anderson, S. Tzou, R. Wahbe, R. Govindan, , and M. Andrews. Support for continous media in the dash system. In *Proceedings of the 10th International Conference on Distributed Computing Systems, Paris France*. IEEE,ACM, 1990.

[4] Aloysius Ka-Lau Mok. *Fundamental Problems of Distributed Systems for the Hard Real-Time Environment*. PhD thesis, Laboratory for Computer Science, Massachussetts Institute of Technology, May 1983.

[5] Karsten Schwan and Rajiv Ramnath. Adaptable operating software for manufacturing systems and robots: A computer science research agenda. Technical report, Computer and Information Science, The Ohio State University, OSU-CISRC-TR-84-4, May 1984.

[6] Hongyi Zhou. *Task Scheduling and Synchronization for Multiprocessor Real-Time Systems*. PhD thesis, College of Computing, Georgia Institute of Technology, May 1992.