Proposal to ARTES
Incremental Iterative Static Scheduling

Gerhard Fohler
Department of Computer Engineering
Mälardalens University
Box 883, 721 23 Västeras
gerhard.fohler@mdh.se

1 Summary

The goal of the proposed research is to develop methods and interactive tools for incremental static scheduling construction. Instead of providing algorithms for only the construction of schedules for a multitude of various task sets, we propose to develop methods that work by incrementally extending an existing, trusted schedule with interactive support. The result of the research can also be used for extending the schedule at run-time.

2 Problem Statement

The design of a real-time system requires the temporal planning of all system activities. Often, this is done as a single scheduling step, last in the design process, as construction of a schedule. A positive result of the scheduling phase can be used directly for the system's run-time management. A negative one, however, usually indicates a problem in the specification of the system, e.g., too high processing demands, requiring an overhaul of the specification and redoing of major part of the design process. Typically, the scheduling phases starts from scratch, i.e., a new schedule is constructed, even if the changes in the specification were only of minor size. Consequently, even the parts of the schedule that were satisfactory are redone, and possibly not constructed as well again. Similarly, an existing schedule for a certain application cannot be extended as the application changes.

Traditionally, static scheduling has been concerned with the efficient construction of feasible schedules for a multitude of various task sets. A change in application and the resulting task set requires the construction of a completely new schedule. The timing plans for unchanged tasks may change entirely in the process.

Industrial practice, however, has shown that task scheduling of applications is only rarely a one-off process. Rather, during the typical development a schedule is constructed and tested first. After it is trusted, following approval and possibly deployment, tasks are added to the original task set. Another round of testing and approval follows, and so on, until the full schedule is completed. New versions of products, or upgrades, are carried out by similar procedures.

As described above, classic scheduling methods abandon the trusted and approved schedules which have been constructed during the process, contrary to industrial practice and needs.

3 Main Ideas

We propose to develop methods for the incremental construction of static schedules to meet demands of industrial design practice. We envision the following steps for schedule construction: First, a static schedule for an initial task set is created. Then, additional tasks are scheduled by being incorporated into the existing task set without violating the feasibility of the schedule and not at all or only minimally modifying it. We plan to support both automatic and interactive methods.

We propose to adapt the slot shifting method [Fohler: PhD94, Fohler: RTSS95] to enable these steps. Originally, slot shifting incorporates tasks at runtime, focussing on efficiency. It extracts information about unused resources and leeway in a static schedule and uses this information to add tasks feasibly, i.e., without violating requirements on the already scheduled tasks. One of the issues will be to revise the notion of "slot", aiming at a more general granularity and thus improving leeway and also utilisation at run-time. Our research will work towards automatic incremental schedule construction and interactive design.

The technique can be seen as an incremental and iterative scheduling approach used early in the design phase. Instead of completing the entire specification before scheduling one could specify a subset of tasks and schedule that task set. After that another subset could be taken under consideration and the schedule for that task set could be incorporated into the first schedule, possibly with some modifications. This method results in an iterative design strategy where one can acquire early knowledge of scheduling decisions and problems as opposed to traditional yes or no answer from the static scheduler.

A graphical tool will display the current schedule and allow the incorporation of tasks. The tool will display restrictions from the original schedule as well as the effects of the new tasks' scheduling. Tasks can be deleted or shifted by hand to envision the consequences. In the tool new and by a fixed amount of time slots. The tool constantly monitors tasks, schedules, and timing requirements, providing information about leeway and free time. The tool will allow to fix the position of tasks in the schedule, i.e., they may not be moved in the next run. Possibly, the tool can be used for schedule construction from scratch as well.

4 Expected Results and Impact

The final direct results of our research will be the incremental scheduling technology and the tool application in an industrial environment.

Instead of providing ”yet another static scheduling algorithm”, our proposal will lead to entirely new grounds in of real-time scheduling, opening a new area of research. Apart from that, we also expect a number of ”spin-off” results: a generalisation of the slot-shifting method, understanding of industrial requirements for scheduling and design practice.

Volvo construction equipment has expressed demand for the proposed research for use in software design for big wheel loader trucks. It will deploy the tool as part of its software development environment and validate the appropriateness and correctness of our research.

5 Project Plan

This project will be conducted by a Ph.D. student with 25% of department duties and supervised by Gerhard Fohler. This means that the project will span over a 30-32 month period.

Month 1-8: Start of project. Literature studies and acquisition knowledge of industrial practice. Start of industrial requirement analysis. Recruitment of undergraduate students. Starting writing a state-of-the-art paper.

Month 9-11: Analysis of student prototypes. Finishing state-of-the-art paper and finalise industrial requirement specification.

Month 12-18: Analysis of requirement specification and existing techniques. Research for new and extending existing techniques based on requirement specification and state-of-the-art. Tool design based on requirements specification, prototype experience and research. Starting with Ph.D. thesis.

Month 18-22: Implementation, testing and field Installation of the tool together with writing Ph.D. thesis.

Month 22-27 Tool deployment and finalising Ph.D. thesis

This schedule will be extended with a 3 months research stay at a highly ranked international institution. During the 30 months the Ph.D. student will also gather about 20 points of graduate courses in relevant areas.

Jukka Mäki-Turja has started to study the problem area and has expressed interest to work on the project for his Ph.D. thesis. His CV is included in the appendix.

6 Preliminary Budget

We project the total project budget to amount to SEK 1.200.000. This sum is comprised of funding for

- one Ph.D. student salary for 2.5 years, considering departmental duties
- supervision cost (salary Gerhard Fohler)
- research stay for Ph.D. student at international institution (2-3 months)
- big screen PC for interactive tool
- conference travel for presentation of results
- a 20% department overhead is added to the above listed items

7 Related Research

The extension of existing schedules has mostly been studied in the context of new tasks arriving to a set of scheduled tasks dynamically, e.g., as aperiodic or sporadic tasks.

The issue of using resources unused by the periodic tasks on single processors has been studied in the context of earliest deadline first scheduling, [Chetto:TOSE89]. [Chetto:IPL89] gives online acceptance tests of O(n).

The use of unused resources in dynamic, single processor scheduling has been proposed by the slack stealing algorithm [Lehoczky:PRTS92] and extended to deal with hard aperiodic tasks, [Thuel:PRTS93, Thuel:PRTS94] and more general scheduling problems, [Davis:PRTS93, Audsley:PRTS94].

Another method to prepare for additional task is the use of server algorithms. Efficient algorithms for earliest deadline first have been presented in [Spuri:PRTS94]. Server algorithms for fixed priority scheduling have been presented in [Lehoczky:PRTS87].

8 Relation to Profile and Industry

Our proposed research aims at solving real-time problems stemming from industrial demand. It will be carried out in co-operation with the industrial partner. The process will provide us with insights in concrete, application specific real-time engineering practice and broaden the scientific foundation for industry.

We expect to learn how to specify and design real-world real-time systems by analysing the actual problems, their requirements and used industrial practice, and by providing solutions and research to meet the encountered demands.

The incremental scheduling method acknowledges the demand for reuse of trusted components. Thus we will enable designers to use building blocks and extend or assemble them to meet new requirements.

9 Context

9.1 The research group

The proposed research will be carried out at the Computer Engineering Department (IDt) at Mälardalens University, Västeras in co-operation with Volvo construction equipment. It is intended to fit with other successful research efforts at IDt for Volvo, in particular the already developed scheduling tool by Christer Eriksson and Kristian Sandström.

Real-time Systems research was established at MdH around 1990 by three persons as a glue between academia and industry. It has since then been the most active and fast growing research area, as well as a strong educational profile, at MdH. The real-time research at MdH is supported by a number of regional and national authorities, foundations and companies. As a result, real-time systems has been identified (by the university board) as the research area of highest priority at MdH. It has grown to approximately 30 persons providing 400 full time student equivalents, 17 involved in research.

CUS has been host to Euromicro Workshop on Real Time systems 1994 and will be organising Euromicro Conference 1998.

9.2 Complimentary activities

4 student exam projects are currently being defined to aid in developing technology and experimental prototypes.

9.3 Research and industrial co-operation

During the last year Christer Eriksson and Kristian Sandström at Mälardalen University have developed a scheduling tool that takes real-world constraints into consideration [Sandström97A, Sandström97B]. This tool is in use at Volvo Construction Equipment Components AB. The experiences gained from that project have been very positive. However, integrating this work with incremental scheduling would be very beneficial for the designer that uses a tool based on static scheduling especially when considering testability.

Contact person at Volvo Construnction Equipment Components AB:

Nils-Erik Bånkestad
Dept. TUE
Volvo Construnction Equipment Components AB
631 85 Eskilstuna
vcecmp.neb@memo.volvo.se

10 References

[Che89a] H. Chetto and M. Chetto. ***Scheduling Periodic and Sporadic Tasks in a Real-Time System***. Information Proceedings Letters, 30:177-184, February 1989.

[Che89b] H. Chetto and M. Chetto. ***Some Results on the Earliest Deadline Scheduling Algorithm***. IEEE Transactions on Software Engineering, 15(10):1261-1269, October 1989.

[Dav93] R.I. Davis, K.W. Tindell, and A. Burns. Scheduling Slack Time in Fixed Priority Preemptive Systems. In Proceedings 14th Real-Time Systems Symposium, pages 222-231, December 1993.

[Foh94] G. Fohler. ***Flexibility in Statically Scheduled Real-Time Systems***. Ph.D. Thesis, Technisch-Naturwissen-schaftliche Fakultät, Technische Universität Wien, Wien, Österreich, April 1994.

[Foh95] G. Fohler. ***Joint scheduling of distributed complex periodic and hard aperiodic tasks in statically scheduled systems***. In Proceedings 16th Real-Time Systems Symposium, Pisa, Italy, December 1995.

[Leh87] J.P. Lehoczky, L. Sha, and J.K. Strosnider. ***Enhanced Aperiodic Responsiveness in Hard Real-Time Environments***. In Proceedings 8th Real-Time Systems Symposium, pages 261-270, 1987.

[Leh92] J.P. Lehoczky and Sandra Ramos-Thuel. ***An Optimal Algorithm for Scheduling Soft-Aperiodic Tasks in Fixed-Priority Preemptive Systems***. In Proceedings of 13th Real-Time Systems Symposium}, pages 110--123, December 1992.

[Thu93] S. R. Thuel and J.P. Lehoczky. ***On-Line Scheduling of Hard Deadline Aperiodic Tasks in Fixed-Priority Systems***. In Proceedings 14th Real-Time Systems Symposium, pages 160-171, December 1993.

[Thu94] S. R. Thuel and J.P. Lehoczky. ***Algorithms for Scheduling Hard Aperiodic Tasks in Fixed-Priority Systems using Slack Stealing***. In Proceedings 15th Real-Time Systems Symposium, pages 22-33, San Juan, Puerto Rico, December 1994.

[Spu94] M. Spuri and G.C. Buttazzo. ***Efficient Aperiodic Service under Earliest Deadline Scheduling***. In Proceedings 15th Real-Time Systems Symposium, pages 2-11, San Juan, Puerto Rico, December 1994.

[Aud94] N.C. Audsley, R.I. Davis, and A. Burns. ***Mechanisms for Enhancing the Flexibility and Utility of Hard Real-Time Systems***. In Proceedings. 15th Real-Time Systems Symposium, pages 12-21, San Juan, Puerto Rico, December 1994.

11 Appendices

Appendix A: C.V. of Gerhard Fohler

Appendix B: C.V. of Jukka Mäki-Turja

Appendix C: Supporting letter from Volvo construction equipment

Appendix A

C.V. of Gerhard Fohler

Appendix B

C.V. of Jukka Mäki-Turja

Appendix C

Supporting letter from Volvo construction equipment