Progress report: ARTES project **WARPING** Applications of wait/lock-free protocols to real-time systems

Hans Hansson
Department of Computer Engineering
Mälardalens Högsk., S-721 23 Västerås
E-Mail: han@idt.mdh.se

Marina Papatriantafilou   Philippas Tsigas
Chalmers University of Technology
Department of Computing Sciences
S-41296 Göteborg
Email: (ptrianta, tsigas)@cs.chalmers.se

August 1999

1 Progress Summary

The project started in April 1999 and its main goal is to study and apply the lock/wait-free methods in order to derive new protocols and algorithms for real-time systems and show how they can improve the system behaviour, compared with lock-based methods. An important area that we plan to explore is how to extend current *real-time operating system kernels* with these methods.

During the first 4 month of the project we have:

- introduced the students to the area of lock-free and wait-free interprocess communication.
- had the first contacts with the industrial partners and planned our collaboration
- continued our work on the subject with the new students involved.

As a result of this effort:

- a significant part of the literature in the subject has been covered and a monthly project-seminar series has been introduced.

- we have received the TPK-sources from the NorthernReal-time Group (Ken Tindell) and will receive sources from ENEA OSE systems (Lars Österberg) soon.

- we continued our work on wait-free, lock-free snapshot implementations on a real-time system consisting of CAN-bus-connected nodes. The new results continue the first experimental results that we started some time ago and show clearer the significant improvements that can be achieved by using lock-free and wait-free techniques and the trade offs between wait-free and lock-free implementations. This work is going to be presented at SNART '99.

2 Scientific Merits

In most real-time systems, concurrency and access to shared resources are controlled by *locking*. A serious disadvantage is that locking may give rise to *priority inversion*, i.e. situations in which a high-priority task has to wait for a low-priority task to release a lock. That is a serious problem, because it can make a task miss its deadline, which, in turn can cause various types of disaster in the system.

The conventional methods to cope with this problem are based on having the kernel of the system dynamically adjust priorities to ensure bounds on the waiting times; however, the problem of waiting due to blocking is still there, enforcing serialisation on the processes and causing concurrency bottlenecks and possibly also deadlocks. The drawbacks of locks are expected to increase and make the use of locks more problematic, as synchronisation becomes finer grained, and as the cost of memory delays and de-scheduling become even higher related to processor speed. It is possible to share data and objects, without using locks. Wait/lock-free interprocess communication/coordination permits access to concurrent objects without the use of locking; thus, it offers guarantees not only regarding priority inversion (it eliminates this problem altogether), but also regarding *efficiency* (by allowing maximum concurrency and thus, low completion times) and *fault-tolerance* (no task is blocked due to a task that crashed while holding a lock). The wait-free condition guarantees progress and completion for every job (of every task) regardless of the execution speeds (and priorities) of the other tasks in the system. *Lock-free* implementations have a more relaxed requirement: locking is ruled out, as in wait-free implementations, but repeated interferences (e.g. under extreme load conditions of the system) may cause a given operation to take longer time to complete. Comparing the two, lock-free methods imply less-overhead (i.e. in memory requirements), while wait-free methods guarantee fully predictable behaviour for every task.

The so-far experience with non-locking synchronisation has shown that it can offer significant benefits in practice and —more significantly— it is efficient alternative to lock-based synchronisation methods for managing concurrency and access to shared resources. However, there has been only limited (although very successful and promising) effort to study and apply the lock/wait-free synchronisation ideas on on real-time systems and in practice in general.

3 Project Plan

Since our project started only 4 months ago we do not have major revisions on our project plan.

Basically -as we explained in the initial plan- we will receive the source code from ENEA and Northern Real-Time Group and until the end of the 1999 study their systems and the architectures for which they are applicable to. The purpose of this is twofold:

1. to identify the data structures whose implementation is a significant factor for the performance of the respective real-time OS

2. to identify the synchronisation capabilities (via the wait-free agreement protocols that they support) of the architectures on top of which these OS are running.

This will lead to the identification of the most efficient feasible ways for lock-free implementation of these basic kernel data structures for the respective architectures.

After the end of the above first phase, we plan a gradual incorporation of the new data structure implementations in the existing kernels; including testing and evaluation of these. We believe that it is very important not to build a new OS from scratch, but only modify existing ones, with an ambition to keep the application programmers interface (API) intact. This will allow us to concentrate on the wait-free/lock-free aspects in real time operating systems and to obtain more exact measurements of the improvements; moreover, the expected improvements of the OS may thus be immediately available to the community that is already using it. At the beginning of this second phase, or better at the end of the first phase, we will refine our project plan and identify the concrete steps that the different sites have to take.

4 Industrial Relevance

The project has clear industrial relevance since the goal is to show that (and how) commercial RT-kernels can be extended with wait-free/lock-free mechanisms, as will be evaluated by real case-studies. Industrial relevance is also indicated by the interest from our two industrial partners: ENEA OSE Systems and Northern Real-Time Group. On the exploitation side we expect that (if we are successful) wait free techniques will be introduced in commercial RT-kernels and OSs.

5 Staff

The following two graduate students have been recruited to the project:

- Per Håkan Sundell, Computer Science, Chalmers
- Björn Allvin, Computer Engineering, Mälardalen Univ.

In addition to these and their supervisors Philippas Tsigas (Chalmers) and Hans Hansson (Mälardalen), Marina Papatriantafilou (PhD, Chalmers) and Andreas Ermedahl (PhD student, Uppsala Univ.) participate in the project (at an approx. 10% activity level with funding from other sources).

6 Mobility

In addition to support for two graduate students the project has received a separate mobility grant (154 KSEK) from ARTES to cover additional costs related to the distributed nature of the project and industrial co-operations. Specific mobility related activities this far, include:

- Björn Allvin, Andreas Ermdahl, Hans Hansson, and Per Håkan Sundell, attended the 11th Euromicro Workshop on Real-Time Systems in York, UK, where they also had a meeting with Ken Tindell (and others) from the Northern Real-Time Grop. Both the Workshop and the meeting were very fruitful for the project.
- There has been three project meetings until now. The first meeting took place in Gothenburg March 30th, the second one took place in Uppsala May 5th and the third one took place in Västerås June 7th. Our plan is to have a meeting every month. At these meetings we regularly schedule a seminar talk, in addition to the discussions, project planning, etc.

Up until now we have already had two types of mobility that we expect to continue having on even larger scale:

- mobility between the participating academic nodes
- mobility between academia and industry: we expect that the graduate students will spend some time (a few weeks) at our industrial partners to learn more about their products and how the wait free techniques can be integrated in their products and development process.

7 Publication

B Allvin, H Hansson, A Ermedahl, M Papatriantafilou, H Sundell, P Tsigas. Evaluating the Performance of Wait-Free Snapshots in Real-Time Systems . In Proc of the SNART'99 Conference on Real-Time Systems, Linköping, 24-25 Augusti, 1999.

8 WWW page of the project

The WWW page of our project can be found in the following URL:
`http://www.mrtc.mdh.se/projects/warp/`