Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems

Paul Pop
Dept. of Computer and Information Science,
Linköping University, Sweden
http://www.ida.liu.se/~paupo

1. Motivation

Communication-intensive heterogeneous real-time systems are systems where communication between the functions implemented on different nodes has an important impact on the overall system properties such as performance, cost, maintainability, etc.

The application software running on such distributed architectures is composed of several functions. The way the functions have been distributed on the architecture has evolved over time. Initially, in automotive applications, for example, each function was running on a dedicated hardware node, allowing the system integrators to purchase nodes implementing required functions from different vendors, and to integrate them together into their system. The number of such nodes in the architecture has exploded, reaching more than 100 in a high-end car. This has created a huge pressure to reduce the number of nodes, use the resources available more efficiently, and thus reduce costs.

However, in order to use the resources more efficiently and reduce costs, several functions have been integrated in one node and, at the same time, certain functionality has to be distributed over several nodes. Although an application is typically distributed over one single cluster, we also begin to see applications that are distributed across several clusters. This trend is driven by the need to further reduce costs, improve resource usage, but also by application constraints like having to be physically close to particular sensors and actuators. Moreover, not only are these applications distributed across several nodes or even network clusters, but their functions can exchange critical information through the gateway nodes.

Such safety-critical hard real-time distributed applications running on heterogeneous distributed architectures are inherently difficult to analyze and implement. Due to their distributed nature, the communication has to be carefully considered during the analysis and design in order to guarantee that the timing constraints are satisfied under the competing objective of reducing the cost of the implementation.

A characteristic of the research efforts concerning the design of embedded systems is that authors concentrate on the design, from scratch, of a new system optimized for a particular application. For many application areas, however, such a situation is extremely uncommon and only rarely appears in design practice. It is much more likely that one has to start from an already existing system running a certain application and the design problem is to implement new functionality (including also upgrades to the existing one) on this system. In such a context it is very important to operate no, or as few as possible, modifications to the already running application. The main reason for this is to avoid unnecessarily large design and testing times. Performing modifications on the (potentially large) existing application increases design time and, even more, testing time (instead of only testing the newly implemented functionality, the old application, or at least a part of it, has also to be retested).

However, this is not the only aspect to be considered. Such an incremental design process, in which a design is periodically upgraded with new features, is going through several iterations. Therefore, after new functionality has been introduced, the resulting system has to be implemented such that additional functionality, later to be mapped, can easily be accommodated.

2. Contributions

In this thesis, a safety critical application is viewed as a set of interacting processes mapped on heterogeneous networks consisting of several interconnected programmable processors. Process interaction is not only in terms of dataflow but also captures the flow of control.

We have considered both the non-preemptive static cyclic scheduling and the static priority preemptive scheduling approaches for the scheduling of processes and messages.

The scheduling and mapping strategies are based on a realistic communication model and execution environment. We take into consideration the overheads due to communication and the execution environment, and consider the requirements of the communication protocol during the scheduling and mapping tasks.

In addition, the mapping and scheduling techniques are considered inside an incremental design process, where the modification of existing applications has to be minimized, and the resulted system has to be structured in such a way that future applications can also be accommodated.

The main contributions of this thesis are:

- a less pessimistic schedulability analysis technique that bounds the worst-case response time of a hard real-time application with both control and data dependencies;
- a schedulability analysis in the context of a communication protocol employing a time-division multiple access scheme, considering four different approaches to message scheduling;
- a schedulability analysis for hard real-time applications mapped across multi-cluster distributed real-time systems consisting of time-triggered and event-triggered clusters, interconnected via gateways, including communication buffer size and worst case queuing delay analysis for the gateways responsible for routing inter-cluster traffic;
- static scheduling algorithms for systems with both data and control dependencies, that take into consideration the overheads due to the communication and the execution environment, and consider the requirements of the communication protocol during the scheduling process;
- several optimization strategies for the synthesis of the bus access scheme in order to fit the communication particularities of a given application;
- approaches to mapping and scheduling for hard real-time applications within an incremental design process, such that the already running applications are disturbed as little as possible and there is a good chance that, later, new functionality can easily be added to the resulted system.