Distributed Real-Time Systems
An Introduction to the Minitrack

Insup Lee
The University of Pennsylvania
Krishna Kavi
The University of Texas at Arlington
and
Nikola Serbedzija
GMD FIRST

A distributed real-time system is an integrated system comprising a set of dedicated hardware that monitors real-world phenomena, acts and reacts on events within specified time period. A real-time system is application driven: its requirements are dictated by the outside environment, not by the computer. The rapidly expanding application area poses a constant pressure to the computing community to improve skills and techniques used for the design and development of such systems.

Real-time technology has always been considered as one of the most complex areas of computer science. Time-constraint information processing is not only about efficiency, it involves timely response, correctness, concurrency, distribution, modeling, assessment and analysis. And the achievements and products of real-time technology are judged in real-life, not by toy examples.

The goal of the minitrack is to bring together the researchers dealing with different aspects of distributed real-time system design in order to answer the question how to achieve timeliness and predictability using high-level software techniques on distributed hardware and still maintain correctness and safety, at an acceptable cost/performance ratio. The nine contributions are divided into three sessions: Scheduling Algorithms, Real-Time Systems and Tools, and Formalisms.

Scheduling Algorithms.

Since the introduction of rate-monotonic scheduling, there has been an exponential growth in new scheduling algorithms, heuristics and analysis techniques. However, the very complex nature of real-time systems, with diverse requirements, and diverse environments where the real-time systems are used, there is still a need to investigate new scheduling approaches to meet specific objectives. The 3 papers selected for presentation in this session reflect this diversity. The first paper by K. J. Lin and A. Herkert addresses the need to produce results at regular intervals such that temporal failures can be detected sooner. The fault-tolerance requirements could lead to "jitter", where the results are produced at different intervals in different time periods. The authors present techniques to reduce such jitter. The second paper by S. Baruah and M. E. Hickey address the problem of finding on-line scheduling algorithms for imprecise computations. They define a measure to estimate the competitive ratio for resources as an indication of the goodness of on-line scheduling algorithms. The third paper by B. Hamidzadeh and Y. Atif presents a class of dynamic scheduling algorithms for real-time systems with aperiodic tasks.

Real-Time Systems and Tools.

This session deals with high-level techniques for programming, debugging, performance monitoring and maintenance of distributed real-time systems. The high-level support ranges from special-purpose programming languages through software environments, to a special-purpose operating system.

The first paper presents an on-line performance analyzer for a functional programming language Erlang. The authors describe both, the high-level programming language Erlang, dedicated for building large-scale industrial real-time applications, and a performance tool PEPA that monitors and measures performance of soft real-time
systems programmed in Erlang. The performance monitoring tool is scalable and has a sophisticated graphical user interface.

The next paper describes the debugging and dynamic modification facilities of the "Testbed" embedded system environment for distributed real-time systems. With the Testbed, a developer may observe and interact with the application while simultaneously monitoring and dynamically modifying the internal state of the application. The Testbed allows for possible corrections without a need to stop and restart the application.

The third paper deals with the design of a real-time co-operating system for simultaneous support of real-time and non-real-time activities. The authors developed virtual co-processors with an architecture analogous to hardware used in standard workstations and personal computers. The co-operating system RTCOS enables user to transform some of the general purpose processors into real-time co-processors and still use the programming environment available for general purpose operating system.

Formalisms.

Because of the complexity of real-time systems and the unpredictable stimuli that trigger reactions from the system, the validation of correctness cannot be based solely on exhaustive testing. Formal methods during the life-cycle of the system design and development must be relied on to achieve a high confidence in the correctness of the system. This session includes three papers that present different formalisms and their utilization in the design of real-time systems. The first paper by C. Priami discusses the need for integrating the performance analysis with the verification of behavior correctness. For this purpose, the author uses a stochastic process algebra called stochastic p-calculus. The author demonstrates the utility of this formalism for integrating the topology of interconnection networks in distributed real-time system.

The second paper by B. Belkhouche, R.R. Lang and C. W. Ng introduces a branching time semantics for the rendezvous mechanism of the Ada programming language. The authors utilize tree-based semantic domain to capture the denotational semantics of the rendezvous mechanism. The third paper by A. Choudhary, V. Gehlot and B. Narahari introduce a real-time specification and verification language known as PRESTEL. The authors believe that a two level specification is beneficial to the design of complex systems, where the lower level constructs can be used to specify real-time algorithms and intra-task level performance requirements, while the higher level constructs can be used to compose the tasks in meaningful ways.

The choice of nine papers was based on a thorough and rigorous reviewing process. A number of high quality papers had to be rejected. The minitrack coordinators wishes to acknowledge the reviewers for their valuable time and comments on the submissions. Their constructive comments together with authors' outstanding efforts make this minitrack a high quality event. Special thanks go to conference organizers for constant support during the preparation of the minitrack.