

Study of Real-time Scheduling Model and Real-time Scheduling Algorithms with Fault-Tolerance

Xiao Qin

(April 1999)

Department of Computer Science

Huazhong University of Science and Technology

ABSTRACT

New generation real-time systems must support real-time applications by maintaining an environment that satisfies real-time, predictability and reliability requirements. In the research of new generation real-time systems, architecture of real-time systems, real-time scheduling algorithms with fault-tolerance and developing tools for real-time systems are three of those key points.

In the first part of this thesis, models of architectures for distributed real-time systems are studied. These models, two for hardware design and another is for software, are quite useful in the distributed real-time applications. When design and implement distributed real-time system, first design hardware and then design software according to these models. In this way, the manufacturing costs of the distributed real-time system can be reduced.

Many real-time systems have the requirements of high reliability, the real-time tasks must finish with the specified deadlines in spite of hardware and software failure. In the aspect of scheduling, this thesis explores fault-tolerant real-time techniques. Before presenting real-time scheduling algorithm with fault-tolerance, the models of real-time scheduling are discussed. These models include static model and hybrid (static/dynamic) model of fault-tolerant real-time scheduling.

This thesis studies four real-time scheduling algorithms with fault-tolerance. First, “backup copies scheduled last” (BKCL) and “tasks without fault-tolerant requirements scheduled last” (NFRL) are presented. These two scheduling algorithms can schedule the tasks with the fault-tolerant requirements together with those without fault-tolerant requirements. For the tasks with fault-tolerant requirements, BKCL and NFRL can generate schedules that can tolerate one processor failure. Performance analysis of these two algorithms are presented, each algorithm has advantage under different workload. NFRL is

better than that of BKCL when the number of tasks with fault-tolerant requirements are far smaller than that of the tasks without fault-tolerant requirements. And on the other hand, when the number of tasks with fault-tolerant requirements are far greater than that of the tasks without fault-tolerant requirements, the performance of BKCL is better than that of NFRL.

Based on BKCL, this thesis proposes another efficient scheduling algorithm that is named as EBKCL for short. In BKCL, it is assumed that there are no overlaps between the backup copies. This assumption is enhanced in the algorithm EBKCL. If the primary copies of two tasks are allocated on two different processors (P_1 and P_2), and the backup copies of these two tasks are scheduled on the same processor P_3 , then the backup copies of the tasks are allowed to be overlapped. With this technique, the performance of EBKCL is better than that of algorithm BKCL as showed in the simulation experiments. In order to enhance the performance of the static real-time scheduling algorithm with fault-tolerance, a hybrid-scheduling algorithm is studied. The simulation shows that hybrid-scheduling algorithm has better performance than static scheduling algorithm.

Another point we focus in this thesis is real-time disk scheduling. Many disk algorithms of two-headed disk system and two-headed mirrored disk systems are illustrated. After presenting non real-time disk scheduling algorithms such as: First Come First Serve (FCFS) and Short Seek-Time First (SSTF), three real-time scheduling algorithms are proposed. They are: earliest deadline first (EDF), feasible earliest deadline first(F-EDF) and ignore missed deadlines(IGM-EDF). These three algorithms take deadlines of I/O requests into consideration, so they are useful in the real-time applications. After doing simulation experiments, we find that real-time algorithms perform better than non real-time algorithms when serving real-time I/O requests. Among real-time algorithms, IGM-EDF is the best one when employed in the hard real-time environment, F-EDF is the next best one, and it is useful in soft real-time applications.

Two-headed mirrored disk system contains several two-headed disks that store the same data, as a result, such disk system can be designed for real-time fault-tolerant application. A fault-tolerant disk scheduling is described and the probability of the failure of a real-time I/O task is derived in this thesis. The real-time fault-tolerant model can result in robustness for the real-time two-headed mirrored disk system.

In the last part of this thesis, a set of developing tools for real-time scheduling is discussed. The tools, used to study and develop real-time scheduling algorithms, include the automatic testing system for UNIX real-time scheduling algorithms and the simulation system for real-time scheduling algorithms. Two systems in this thesis present some general ideas of developing tool for real-time scheduling. According to these general ideas, some tools based on other operating systems can also be designed.

Key Words: Distributed real-time, system architecture, fault-tolerant , hierarchical model, real-time scheduling, heuristics, Hybrid, two-headed disk system, operating system, simulation experiment, performance analysis, automatic testing system, simulation system , UNIX system