Wait-Free Synchronization in Quantum-Based Multiprogrammed Systems* (Extended Abstract)

James H. Anderson, Rohit Jain, and David Ott

Department of Computer Science
University of North Carolina at Chapel Hill

Abstract. We consider wait-free synchronization in multiprogrammed uniprocessor and multiprocessor systems in which the processes bound to each processor are scheduled for execution using a scheduling quantum. We show that, in such systems, any object with consensus number P in Herlihy's wait-free hierarchy is universal for any number of processes executing on P processors, provided the scheduling quantum is of a certain size. We give an asymptotically tight characterization of how large the scheduling quantum must be for this result to hold.

1 Introduction

This paper is concerned with wait-free synchronization in multiprogrammed systems. In such systems, several processes may be bound to the same processor. In related previous work, Ramamurthy, Moir, and Anderson considered wait-free synchronization in multiprogrammed systems in which processes on the same processor are scheduled by priority [4]. For such systems, Ramamurthy et al. showed that any object with consensus number P in Herlihy's wait-free hierarchy [2] is universal for any number of processes executing on P processors, i.e., universality is a function of the number of *processors* in a system, not the number of *processes*. An object has *consensus number* C iff it can be used to solve C-process consensus, but not $(C+1)$-process consensus, in an asynchronous system in a wait-free manner. An object is *universal* in a system if it can be used to implement any other object in that system in a wait-free manner.

In this paper, we establish similar results for multiprogrammed systems in which quantum-based scheduling is used. Under quantum-based scheduling, each processor is allocated to its assigned processes in discrete time units called *quanta*. When a processor is allocated to some process, that process is guaranteed to execute without preemption for Q time units, where Q is the length of the quantum, or until it terminates, whichever comes first. In this paper, we show that quantum-based systems are similar to priority-based systems with regard

* Work supported by NSF grants CCR 9510156 and CCR 9732916, and by a Young Investigator Award from the U.S. Army Research Office, grant number DAAH04-95-1-0323. The first author was also supported by an Alfred P. Sloan Research Fellowship.

to universality. In particular, we show that any object with consensus number P in Herlihy's wait-free hierarchy is universal in a quantum-based system for any number of processes executing on P processors, *provided* the scheduling quantum is of a certain size. We give an asymptotically tight characterization of how large the scheduling quantum must be for this result to hold.

Our results are summarized in Table 1. This table gives conditions under which an object with consensus number C is universal in a P-processor quantum-based system. In this table, T_{max} (T_{min}) denotes the maximum (minimum) time required to perform any atomic operation, Q is the length of the scheduling quantum, and c is a constant that follows from the algorithms we present. Obviously, if $C < P$, then universal algorithms are impossible [2]. If $P \leq C \leq 2P$, then the smallest value of Q that suffices is a value proportional to $(2P + 1 - C)T_{max}$. If $2P \leq C < \infty$, then the smallest value of Q that suffices is a value proportional to $2T_{max}$. If $C = \infty$, then Q (obviously) can be any value [2].

An important special case of our main result is that reads and writes are universal in quantum-based uniprocessor systems ($P = 1$). In this case, the scheduling quantum must be large enough to encompass the execution of eight high-level language instructions (see Theorem 1). In any practical system, the scheduling quantum would be much larger than this. Thus, in practice, Herlihy's wait-free hierarchy collapses in multithreaded uniprocessor applications in which quantum-based scheduling is used.

It is important to note that the results of this paper do *not* follow from the previous results of Ramamurthy et al. concerning priority-based systems, because priority-based and quantum-based execution models are fundamentally incomparable. In a priority-based system, if a process p is preempted during an object invocation by another process q that invokes the same object, then p "knows" that q's invocation must be completed by the time p resumes execution. This is because q has higher priority and will not relinquish the processor until it completes. Thus, operations of higher priority processes "automatically" appear to be atomic to lower priority processes executing on the same processor. This is the fundamental insight behind the results of Ramamurthy et al.

In contrast, in a quantum-based system, if a process is ever preempted while accessing some object, then there are no guarantees that the process preempting it will complete any pending object invocation before relinquishing the processor. On the other hand, if a process can ever detect that it has "crossed" a quantum boundary, then it can be sure that the next few instructions it executes will be performed without preemption. Several of the algorithms presented in this paper employ such a detection mechanism. This kind of detection mechanism would be ill-suited for use in a priority-based system, because a process in such a system can never be "sure" that it won't be preempted by a higher-priority process.

Our quantum-based execution model is based on two key assumptions:

(i) If a process is preempted during an object invocation, then the first such preemption may happen at any point in time after the invocation begins.

(ii) When a process resumes execution after having been preempted, it cannot be preempted again until after Q time units have elapsed.